Design and Optimisation of Oleochemical Processes

Jones, Mark Nicholas

Publication date:
2019

Document Version
Publisher's PDF, also known as Version of record

Ph.D. Thesis
Doctor of Philosophy

DTU Chemical Engineering
Department of Chemical and Biochemical Engineering

Design and Optimisation of Oleochemical Processes

Mark Nicholas Jones

Kongens Lyngby 2020

Process and Systems Engineering Centre (PROSYS)
Department of Chemical and Biochemical Engineering
Technical University of Denmark (DTU)

Søltofts Plads
Building 227
2800 Kongens Lyngby, Denmark
www.kt.dtu.dk

Abstract

Process Systems Engineering (PSE) is a discipline which connects a wide range of chemical engineering topics in a systems view approach. The reason for this systematic view of this scientific field is the need of computational concepts, numerical methods and computer-aided tools which can be applied to different use cases in industry. The multi-scale framework developed in this work encompasses four levels of application:

(I) A server based property prediction software prototype which quantifies the uncertainty of group contribution methods and quantitative structure property relationship models and provides the confidence bounds of the estimates.

(II) A modelling development level which allows the user to develop models in a flexible way by using common programming languages for fast prototyping (Python) and high performance computing (Fortran).

(III) An interface to process simulators to analyse and optimise entire flowsheets with advanced routines.

(IV) A superstructure optimisation layer where surrogate models generated from unit operations or process models can be embedded in a superstructure formulation and solved for the optimal process structure and operating point.

The contributions presented in this work show how the developed framework allows to tackle research in machine learning, optimisation and Monte Carlo driven methods such as sensitivity analysis. The developed tools were applied to the oleochemical domain with selected processes. In conclusion, this work demonstrates that a modular approach to process systems engineering, combined with tools integration from various vendors, allows to gain new knowledge in a time-efficient and augmentable manner.

Resumé

Process systems engineering (PSE) er en disciplin som sammenknytter en række kemitekniske emner med fokus på en systemisk tilgang. Årsagen for denne systematiske tilgang er efterspørgslen af beregningsmæssige koncepter, numeriske metoder og computerstøttede værktøjer som kan anvendes til forskellige studier i process systems engineering. Den multi-skala framework som er udviklet og beskrevet i denne afhandling omfatter fire niveauer:

(I) En server baseret software prototype for beregning af kemiske egenskaber kvantificerer usikkerhed af gruppe-bidrags metoder og kvantitative struktur-egenskabs modeller.

(II) Et modellerings niveau hvor brugeren kan udvikle modeller på en fleksibel måde med forskellige programmeringssprog til fast-prototyping (Python) og high performance computing (Fortran).

(III) Et interface til process simulerings værktøjer som gør det muligt at analysere og optimere processskemaer med avancerede rutiner.

(IV) Et superstruktur optimerings-niveau hvor enhedsoperationer kan blive indlejret i superstrukturelle formuleringer og blive løst for den optimale proces-struktur og drift.

Bidragene i dette værk viser hvordan frameworket kan benyttes til at udføre forskning ved hjælp af machine learning, optimering og Monte Carlo drevet sensitivitetsanalyse. De udviklede værktøjer er blev anvendt på det oleokemiske domæne med udvalgte processer. Overordnet viser dette arbejde, at en modulær tilgang til processteknologi kombineret med værktøjsintegration fra forskellige leverandører giver mulighed for at opnå ny viden på en tidseffektiv måde og med muligheden for at udvide.

Preface

The work presented in this thesis was carried out as part of my Ph.D. project from April 15th, 2016 to April 14th, 2019. The work took place at the PROSYS Research Centre, Technical University of Denmark in Lyngby with an external stay of three months at the Department of Computing of Imperial College in London.

This Ph.D. project was funded by the European Unions Horizon 2020 Research and Innovation Programme under the Marie Sklodowska-Curie Grant Agreement No. 675251, and was supervised by:

- Gürkan Sin (main supervisor), Associate Professor, PROSYS Research Centre, Technical University of Denmark, 2800 Kgs. Lyngby, Denmark
- Bent Sarup (co-supervisor), Alfa Laval Copenhagen A/S, 2860 Søborg, Denmark

This work wouldn't have been possible without the guidance and help by many people. First of all I would like to thank my supervisor Gürkan for his support over these three years and the many constructive and fruitful discussions we had. Also for setting up the ModLife project, from which I harvested very valuable knowledge with all the educational training we received from it. Many thanks also to my co-supervisor Bent Sarup for the constructive feedback on my work.

Deep gratitude also to Sasha, Hector and Rasmus for sharing the office with me for so long. I had very much fun with you and you made it a joy to sit for so long in front of the computer. Thank you again Sasha for managing the ModLife project and helping me out with your beloved Fortran language.

Jerome, Robert, Lukasz, Merve, Frederico, Resul, thank you for the countless lunch breaks we had together, and the fun trips during the project meetings and conferences.

Thank you also Jerome for all the support for developing SAFEPROPS and the great time we had at ESCAPE27 and PSE2019.

I would also like to express great thanks to Camille Goulon for helping out with the SAFEPROPS development and to Yasemin Aktas who contributed greatly to the solvent crystallisation model.

Thank you also to Ruth at Imperial to having me as a guest for my external research stay and giving me excellent input on my work. Also thank you to Simon, Johannes and Miten making the hot summer of 2018 bearable at Imperial with fish and chips at lunch.

Great thanks also to the whole PROSYS team for the best work environment one could wish for and for all the country seminars, canoeing and julefrokost events we had together.

I'm deeply grateful to have such great friends, thank you making this world such a nice place and supporting me during my PhD.

And most importantly, thank you mum and dad for your love, support and always being there for me. Thank you Rasmus for being such a great brother, I enjoy every minute hanging out with you.

Am Wasserfall.
Beim Anblick eines Wasserfalles meinen wir in den zahllosen Biegungen, Schlängelungen, Brechungen der Wellen Freiheit des Willens und Belieben zu sehen; aber Alles ist notwendig, jede Bewegung mathematisch auszurechnen. So ist es auch bei den menschlichen Handlungen; man müsste jede einzelne Handlung vorher ausrechnen können, wenn man allwissend wäre, ebenso jeden Fortschritt der Erkenntnis, jeden Irrtum, jede Bosheit. Der Handelnde selbst steckt freilich in der Illusion der Willkür; wenn in einem Augenblick das Rad der Welt still stände und ein allwissender, rechnender Verstand da wäre, um diese Pausen zu benützen, so könnte er bis in die fernsten Zeiten die Zukunft jedes Wesens weitererzählen und jede Spur bezeichnen, auf der jenes Rad noch rollen wird. Die Täuschung des Handelnden über sich, die Annahme des freien Willens, gehört mit hinein in diesen auszurechnenden Mechanismus.

Friedrich Nietzsche - Menschliches, Allzumenschliches

Ph.D. Publications

The following publications have resulted from this Ph.D. project.

Articles in peer-reviewed journals

[1] H. Forero-Hernandez, M. N. Jones, B. Sarup, A. D. Jensen, J. Abildskov, and G. Sin, "Comprehensive development, uncertainty and sensitivity analysis of a model for the hydrolysis of rapeseed oil," *Computers and Chemical Engineering*, vol. 133, pp. 106–631, 2020. DOI: 10.1016/j.compchemeng.2019.106631.

[2] M. N. Jones, H. Forero-Hernandez, A. Zubov, B. Sarup, and G. Sin, "Splitting triglycerides with a counter-current liquid-liquid spray column: Modelling, global sensitivity analysis, parameter estimation and optimisation," *Processes*, 2019.

[3] M. N. Jones, J. Frutiger, N. G. Ince, and G. Sin, "The Monte Carlo driven and machine learning enhanced process simulator," *Computers & Chemical Engineering*, vol. 125, pp. 324–338, 2019. DOI: 10.1016/j.compchemeng.2019.03.016.

Contributions to peer-reviewed proceedings

[4] J. Frutiger, M. Jones, N. G. Ince, and G. Sin, "From property uncertainties to process simulation uncertainties – Monte Carlo methods in SimSci PRO/II process simulator," in *Proceedings of the 13th International Symposium on Process Systems Engineering (PSE 2018)*, ser. Computer Aided Chemical Engineering, vol. 44, 2018, pp. 1489–1494. DOI: 10.1016/B978-0-444-64241-7.50243-3.

[5] M. Jones, H. Forero-Hernandez, A. Zubov, B. Sarup, and G. Sin, "Superstructure optimization of oleochemical processes with surrogate models," in *Proceedings of the 13th International Symposium on Process Systems Engineering (PSE 2018)*, ser. Computer Aided Chemical Engineering, vol. 44, 2018, pp. 277–282. DOI: 10.1016/B978-0-444-64241-7.50041-0.

[6] H. Forero-Hernandez, M. Jones, B. Sarup, J. Abildskov, A. Jensen, and G. Sin, "A simplified kinetic and mass transfer modelling of the thermal hydrolysis of vegetable oils," in *Proceedings of the 27th European Symposium on Computer Aided Process Engineering (ESCAPE 27)*, ser. Computer Aided Chemical Engineering, vol. 40, 2017, pp. 1177–1182. DOI: 10.1016/B978-0-444-63965-3.50198-7.

[7] M. Jones, H. Forero-Hernandez, B. Sarup, and G. Sin, "Multi-scale modeling approach for design and optimization of oleochemical processes," in *Proceedings of the 27th European Symposium on Computer Aided Process Engineering (ESCAPE 27)*, ser. Computer Aided Chemical Engineering, vol. 40, 2017, pp. 1885–1890. DOI: 10.1016/B978-0-444-63965-3.50316-0.

Contributions to international peer-reviewed conferences

[8] M. Jones, C. Hansen, H. Forero-Hernandez, B. Sarup, and G. Sin, "Monte carlo based sensitivity analysis and derivative-free optimisation," 1st International Young Professionals Conference on Process Engineering (YCOPE 2019), 2019, p. 49.

[9] H. Forero-Hernandez, M. Jones, B. Sarup, J. Abildskov, A. Jensen, and G. Sin, "Modelling, uncertainty and sensitivity analysis of the batch thermal hydrolysis of vegetable oils," 16th Euro Fed Lipid Congress and Expo, 2018.

[10] M. Jones, H. Forero-Hernandez, A. Zubov, B. Sarup, and G. Sin, "Design, global sensitivity analysis and optimisation of a counter-current spray column for splitting triglyceride mixtures," 16th Euro Fed Lipid Congress and Expo, 2018.

[11] M. Jones, H. Forero-Hernandez, B. Sarup, and G. Sin, "Superstructure optimisation with general disjunctive programming and surrogate models," PSE@ResearchDayUK, 2018.

[12] M. Jones, J. Frutiger, J. Abildskov, and G. Sin, "Safeprops: A software for fast and reliable estimation of safety and environmental properties for organic compounds," 2016 AIChE Annual Meeting, 2016.

Contents

Acronyms

AAD	average absolute deviation
API	application programming interface
ARE	average relative error
ARD	automatic relevance determination
CFD	computational fluid dynamics
DFT	density functional theory
EOS	equation of state
FAME	fatty acid methyl ester
FVM	finite volume model
GDP	general disjunctive program
GC	group contribution
GPR	Gaussian process regression
LML	log marginal likelihood
MG	Marrero-Gani
MSE	mean squared error
PCE	polynomial chaos expansion
PSE	process systems engineering
QSPR	quantitative structure-property relationship
RMSE	root mean squared error

SD standard deviation

SMARTS SMILES arbitrary target specification

SMILES Simplified molecular-input line-entry system

SRK Soave-Redlich-Kwong

SSE sum of squared errors

CHAPTER 1

Introduction

Energy efficient design and the use of renewable feedstock are two of the twelve principles of green chemistry [1] to achieve more sustainable processes and to obtain products which have formerly been produced from fossil based sources. Oils and fats from vegetables fall into this category of feedstock. Chemicals derived from this feedstock and processed through mechanical, chemical and enzymatic conversion routes are called oleochemicals.

Oleochemical processes have been studied and applied by the chemical industry for several decades and have undergone profound changes [2]. The global oleochemical industry has grown to a worldwide market and is expected to reach 28.6 billion US Dollars in the year of 2025 which the Southeast Asian (Asian-Pacific) region holds the biggest market share. With the highest proportion, fatty acids made up over 55 per cent of the total demand in the year of 2016 [3].

In the recent time companies which have their production plants based in North America or Europe shift their focus on oleochemical end consumer applications while Asian Pacific based companies have overtaken the market of basic oleochemical products and intermediates [4]. The low margin of glycerol is caused by the globally increased fatty acid methyl ester (biodiesel) production which affects also the profitability of oleochemicals negatively [5, 6]. Therefore, an efficient removal and purification of glycerol is important and further aims are to convert basic oleochemicals to high-value products with applications in food and non-food use cases [7].

This thesis with the title 'Design and Optimisation of Oleochemical Processes' identifies relevant research topics in the oleochemical process domain. The focus was put on extending the methodology of property prediction with uncertainty analysis performed in previous work by Hukkerikar [8] and Frutiger [9], perform multi-platform process modelling, design and analyse oleochemical processes and to perform advanced optimisation techniques.

1.1 Motivation and goals

The aim of this thesis is to provide a framework in process systems engineering (PSE) which integrates current research topics in combination with industrial tools, modelling environments and free-licensed scientific software with the source code being publicly shared. The developed methodologies and tools should be applicable on the different levels of the modular framework and vice versa, the different framework levels should be easily connectable to new routines developed in the scientific community. Further, the framework should be applied to the oleochemical process domain and advance the field with how these developed tools can be utilised for industrial application and what new knowledge can be gained when utilising the framework in respect to property prediction, process, design and optimisation. Especially important is also the identification of the reasons between the differences in performance of routine implementations by various vendors.

Thus, the following challenges motivated this thesis project:

- Establishing a modelling framework which makes use of general purpose programming languages and still keeps the different levels (property prediction, process modelling, flowsheeting and optimisation) highly modular for providing independence from the computing platform and modelling environment.

- New developments in machine-learning and optimisation are shared more and more as public repositories in the scientific community. A framework which can easily make use of these scientific packages is highly desirable.

- Oleochemical processes are in need of properties of compounds where experimental data is rare (especially for fine and high value added chemicals) and needed properties are usually predicted by models obtained through regression. The predictive models have to therefore be further developed and improved in regards to data selection, data scarcity and the predictions' confidence/credible intervals.

- Finding the optimal process structure and operating point of processes is a highly researched topic and superstructure optimisation has still to be made more generic and user-friendly for industry. Surrogate modelling methods can be used to provide the necessary data from lower-level models to the superstructure formulation and by this help to make

superstructure optimisation applicable for a wide range of chemical domains.

A major goal of this thesis is to provide models of oleochemical processes for which no model exist in the process simulator at hand or no shared implementation is provided in the research literature. Given the presented framework and the challenges presented before, the goals of this research project are the following:

- Valuable process models need to be identified with a preliminary technology assessment.
- The developed models need to be integrated in this framework and should be able to connect or be transferable to different modelling environments.
- The property prediction level of the framework can either be accessed as a look-up tool to retrieve needed property data or through an application programming interface (API) to retrieve the property values directly in the routines. The implementation of the property prediction routines should take distributed and scalable systems architectures into account and therefore should be developed as a server-based application. This will also make it possible to connect the unit operation models to the property prediction level.
- Uncertainty and sensitivity analysis needs to be applied to analyse the propagation of the uncertainties from the property estimates to the process output. Novel procedures should be investigated to apply the methods with a process simulator and reduce the evaluation time.
- Optimisation methods should be identified and applied in respect to economic and sustainability benchmarks.

1.2 Outline of the thesis

The structure of this thesis is as follows.

First the multi-scale framework is presented and the reader is given an overview of the developments in process systems engineering in respect to tools integration. Chapter 3 presents the oleochemical process domain and

a technology assessment of selected processes. Important property prediction models are then discussed and how the development of previous and the current work has advanced in respect to property prediction with uncertainty analysis. Further, the property prediction tool SAFEPROPS will be presented in this chapter. Moving from the property prediction to the process design level, an overview of the identified processes is given and how these were modelled, analysed and optimised. The next level uses the rigorous spray column model developed on the process design level to perform superstructure optimisation with surrogate functions. The thesis concludes with an outlook of potential future research and a final conclusion of what has been achieved with this work.

1.3 Summary of main contributions

The following research objectives are addressed in this work and form the main contributions of this thesis.

1.3.1 Methodologies applicable for sustainable process design

A multi-level modelling framework which encompasses property prediction, process modelling and superstructure optimisation has been developed. The framework is applied to the selected processes in the oleochemical process domain. Three processes were modelled: (1) a spray column as a finite volume model (FVM), (2) a molecular distillation process as a connection of unit operations in the process simulator PRO/II and (3) a solid flash algorithm for simulating solvent crystallisation. The modular model library allows to optimise and analyse the processes with external algorithms and analysis tools such as differential evolution for optimisation and variance-based sensitivity analysis. Techniques applied in this work are:

- Data-driven and stochastic property prediction with group contribution methods and quantitative structure-property relationship (QSPR) models
- Uncertainty and global sensitivity analysis via a full Monte Carlo approach and via polynomial chaos expansion (PCE) to reduce the needed number of evaluations

- Derivative-free optimisation for parameter estimation
- Multi-criteria optimisation with respect to economic cost and sustainability
- Surrogate modelling for superstructure optimisation via general disjunctive programming

The model library was implemented in two ways: Either models were implemented in Fortran and Python, or processes were established in the commercial process simulator PRO/II. A Python interface has been developed to access the commercial process simulator.

1.3.2 Improved property models

Property models important for describing fixed physical, thermo-physical and thermodynamic properties of lipids have been studied. The Marrero-Gani (MG) group contribution and the Soave-Redlich-Kwong (SRK) equation of state were applied and a property prediction tool named SAFEPROPS has been developed to support the Marrero-Gani group contribution method and QSPR models. Gaussian process regression (GPR) was applied to develop a methodology for machine-learning driven property prediction.

1.3.3 Process analysis and improvements

Improvements in respect to the selected key processes were e.g. the steam consumption of the spray column and the more efficient separation of saturated and unsaturated fatty acids via solvent crystallisation. Due to the modular structure of the framework the developed routines were applied with different modelling environments.

Bibliography

[1] P. T. Anastas and J. C. Warner, *Green Chemistry: Theory and Practice*. Oxford University Press: New York.

[2] D. E. Haupt, G. Drinkard, and H. F. Pierce, "Future of petrochemical raw materials in oleochemical markets," *Journal of the American Oil Chemists' Society*, vol. 61, no. 2, pp. 276–281, February 1984. DOI: 10.1007/BF02678781.

[3] "Oleochemicals market size, share & trends analysis report by product (fatty acid, glycerol, fatty alcohol), by region (APAC, MEA, Europe, North America, CSA), and segment forecasts, 2018–2025," Grand View Research, Tech. Rep., 2018.

[4] A. McWilliams, "Global markets for oleochemical fatty acids," BCC Research, Tech. Rep., 2017.

[5] I. Atadashi, M. Aroua, and A. A. Aziz, "High quality biodiesel and its diesel engine application: A review," *Renewable and Sustainable Energy Reviews*, vol. 14, no. 7, pp. 1999–2008, 2010. DOI: 10.1016/j.rser.2010.03.020.

[6] R. Ciriminna, C. D. Pina, M. Rossi, and M. Pagliaro, "Understanding the glycerol market," *European Journal of Lipid Science and Technology*, vol. 116, no. 10, pp. 1432–1439, 2014. DOI: 10.1002/ejlt.201400229.

[7] J. Salimon, N. Salih, and E. Yousif, "Industrial development and applications of plant oils and their biobased oleochemicals," *Arabian Journal of Chemistry*, vol. 5, no. 2, pp. 135–145, 2012. DOI: 10.1016/j.arabjc.2010.08.007.

[8] A. S. Hukkerikar, "Development of pure component property models for chemical product-process design and analysis," English, PhD thesis, 2013.

[9] J. Frutiger, "Property uncertainty analysis and methods for optimal working fluids of thermodynamic cycles," English, PhD thesis, 2017.

CHAPTER 2

Multi-scale Framework for Design and Optimisation

2.1 Multi-scale framework for property prediction, unit operation modelling, flowsheeting and superstructure optimisation

Modelling, simulation, control and optimisation of chemical processes demand extensive knowledge of several subjects where the scope includes chemistry, physics, mathematics, computer science, engineering disciplines, economics and sustainability. With the emergence of computer systems and information technology, Roger W. H. Sargent recognised that a systematic and structured view on the chemical and process engineering discipline was in need [1]. This lead to the process systems engineering (PSE) discipline and assisted the chemical engineering community to develop the concepts and algorithms which are nowadays commonly taught in the chemical engineering curriculum and applied in industry.

This brief look into the past gives reason to formulate the question of how computer-aided tools have to be adapted to the future needs of process engineering where machine-learning methods and the simulation of computational expensive models need to be integrated with existing frameworks. Therefore the goal in this work is to develop a flexible and modular tool set which allows to use established process simulators along with novel research methods.

Braunschweig et al. [2] made the distinction between process modelling components (PMCs) and process modelling environments (PMEs) in the year of 2000. Their article describe how computer-aided tools need to be able to be used together. During the last 19 years great progress has been made in information technology and the process systems engineering community has to keep up with the pace of developments. For that reason the CAPE-OPEN standard has been developed with the objective to standardise physical and thermodynamic property packages, unit operation modules, numerical solvers and flowsheet analysis tools. Important developments by the foundation are COM, COBIA and COMBIA which are standardised object models allowing PMCs and PMEs to be integrated or combined. In respect to PMEs one can separate them in PMEs with their own modelling languages (e.g. gPROMS ModelBuilder v5.0, Modelica or GAMS) or PMEs based on a general-purpose language such as Python or Julia (e.g. Pyomo, DAE Tools or Jump) [3]. Further, PMEs can be divided into equation-orientated and modular environments, e.g. process simulators such as PRO/II and Aspen are modular while Pyomo and gPROMS are equation-orientated environments. The latter solve the whole system of equations for all unit operations by the solver instance whereas modular approaches have the unit operation models being solved locally and then passing the solutions to the connected units in the flowsheet [2].

In this work a framework based on general-purpose languages (Python and Fortran) is proposed and different PMEs and PMCs are adapted to the individual use case of the task to be accomplished. An example would be the modelling of a unit operation as a PMC in Fortran applying a functional programming paradigm [4] and defining CAPE-OPEN compliant data objects. Whereas Pyomo is used as the PME for the ease of formulating superstructure optimisation problems and solving them. The framework supports wrapping the low-level Fortran code with a high-level language such as Python. Unit operations can then be analysed with the newest developments in research in form of scientific packages. Further, functional programming can be combined with object-oriented programming when for example a class structure is needed and several instances of a distillation unit operation have to be created.

The framework is divided into four levels (Figure 2.1) with the property prediction level at the lowest position. The property prediction models are called from the unit operation model layer above to retrieve the necessary property values of the chemical system and to simulate the unit operation. During the simulation the unit operation model layer will access the property prediction layer multiple times during the iteration to obtain a converged so-

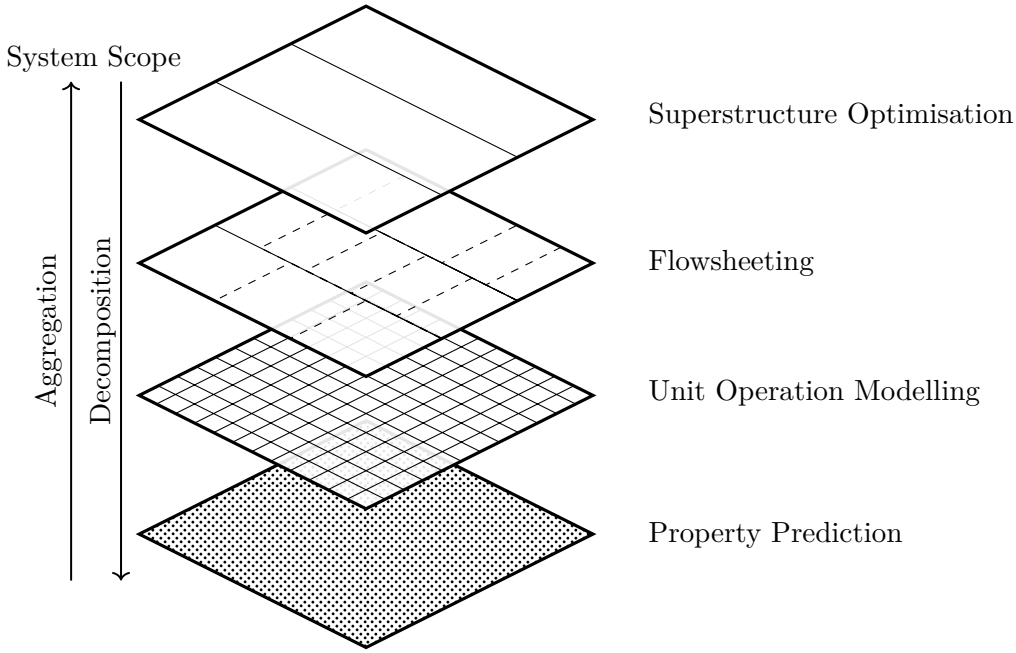


Figure 2.1: Multi-scale framework encompassing property prediction, unit operation modelling, flowsheeting and superstructure optimisation..

lution. The process flowsheet layer connects different unit operations through mass, energy and information streams. Likewise, the flowsheet layer will iterate and call the unit operations and property prediction models multiple times to reach a converged solution. Usually the iteration algorithm applied by modular process simulators is the Newton-Raphson iteration method [5] to solve the system of nonlinear equations. The top level performs superstructure optimisation. On this level the information from all lower levels is aggregated in surrogate functions to describe all possible process configurations (superstructure) in terms of conversion, mixing, splitting, separation, economic cost and sustainability indicators. The optimisation will then identify the optimal process structure and operating point subject to the defined global constraints. The framework allows to access each level individually to perform computational methods such as sensitivity analysis, Monte Carlo techniques, machine-learning and optimisation routines (Figure 2.2).

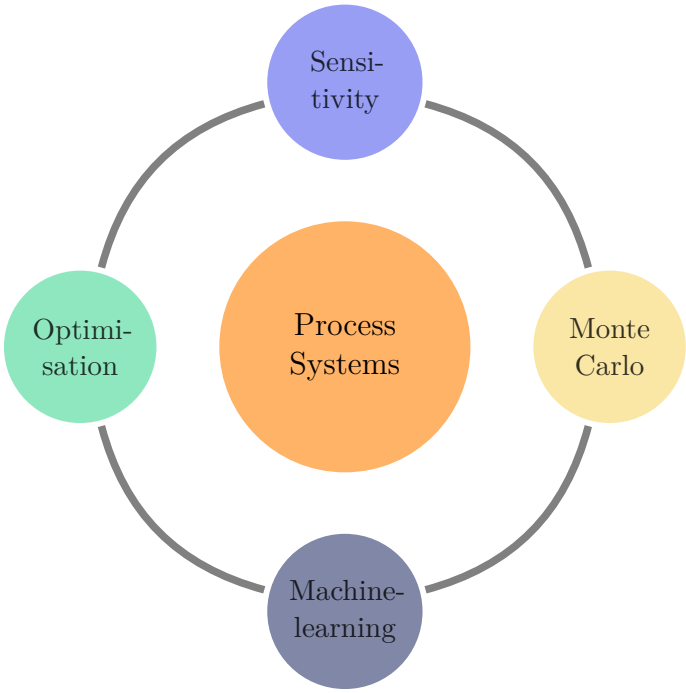


Figure 2.2: Process systems on the different levels of the multi-scale framework can be connected to different methods.

It is important to share the knowledge of how such frameworks and tools can be implemented. Therefore a brief instruction is provided in the appendix on how to develop user-added unit operations or subroutines for PRO/II. In the following the different framework levels are presented in regards to their characteristic functionality.

2.1.1 Property prediction level

Fixed-physical, temperature-dependent and thermodynamic properties are calculated with different models or correlations. These can be quantum-mechanic based models where the electron system of the atoms in a molecule are calculated with density functional theory (DFT). Also simplified, semi-empirical models of describing a molecule exist (e.g. UNIFAC) which dissect a molecule in specified groups of connected atoms and assign different parameter values to these groups and the interaction between each other. These parameters were

fitted to experimental phase equilibrium data and allow to predict the activity coefficient in non-ideal mixtures in case of the UNIFAC model. The activity coefficient belongs to the thermophysical property class which is dependent on temperature, pressure and the mixture composition. Temperature-dependent or fixed physical properties can be retrieved from correlations which are fitted to the experimental data. There also exist so called group contribution methods which, similar to the UNIFAC model, dissect a molecule into predefined groups. The group contribution values to the property are then obtained through regression of the specific group contribution model formulation. Equations of state models describe the relation between pressure, volume and temperature. To calculate the properties of a chemical component or mixtures in the gas phase the Soave-Redlich-Kwong (SRK) equation of state (EOS) could for example be applied. For the SRK model the critical pressure, the critical temperature and the acentric factor is needed as input parameters.

2.1.2 Unit operation modelling level

The modelling of unit operations belongs to the classical engineering task and takes considerable time and effort to generate a so called 'digital twin' of the real, physical system. The phenomena such as the thermodynamic behaviour, the reaction kinetics, the mass- and heat transfer of a chemical system have to be described to model a process. Usually assumptions are made and if the model can be verified with the experimental data or industrial scale process data, the assumptions are valid. If not, the assumptions have to be critically evaluated and consequently modified or dismissed. A model will also show if it has advanced predictive performance if analysis and optimisation methods can be applied to the model and give correct results. The choice of the PME in which the model is implemented can differ due to either the application case of the model, who has to utilise the model (domain expert, optimisation consultant or software developer) or if the model has to be implemented on special devices such as microcontrollers. An important aspect to not underestimate is the extensive documentation and testing of the implementation so that developers will always understand what action the code is meant to initiate.

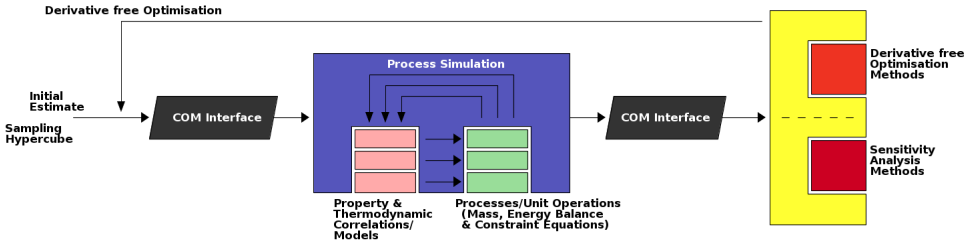


Figure 2.3: Interfacing of process simulator (e.g. Pro/II or Aspen) for black box evaluation of unit operations or flowsheets.

2.1.3 Flowsheeting level

At the third layer the unit operations are connected to form a process flowsheet. Here the inlet and outlets of each unit operation are connected with each other. Tools such as sequential or equation oriented process simulators can help to perform this task with a graphical user interface. To be able to use the convenience of a process simulator an application programming interface (API) can be utilised to connect to it. This gives the user the possibility to populate the flowsheet embedded in the simulator with values, run the simulation and store the output values in a Monte Carlo approach. In this way the process simulator can be used to perform variance-based sensitivity analysis or derivative-free optimisation as seen in Figure 2.3. Further, the API (COM interface) can also be used to perform surrogate modelling or other techniques which make use of multiple evaluations of a process.

2.1.4 Superstructure optimisation level

The top level defines the superstructure of a specific process domain with all possible processing routes. For this task surrogate models are obtained from the rigorous unit operations or flowsheets and embedded in a general disjunctive program (GDP). This keeps the superstructure formulation simple and the advantage of applying surrogate functions is that the rigorous process calculations are transferred to the superstructure optimisation layer whereas other frameworks [6, 7] describe the possible process steps for example with simplified constants for the conversion rates, separation or cost factors. No

optimal operating points of the individual determined unit operations can be provided with these frameworks.

2.2 Framework implementation

A server based property prediction application has been developed and is presented in this work. The software prototype allows to be deployed on a server by a hosting service or on the server of the companies' or research institutions' intranet for internal use. It is also possible to run the application just on the local machine if the user has to perform property prediction for single user modelling and simulation purposes. The web-interface allows easy data retrieval through a graphical interface. The API allows to directly define the needed property estimates in a script and access the values for the subsequent calculation routines. The SQL database allows to easily extend the prediction model library with new or modified versions of group contribution or other models. The stack of tools which make up the whole application is solely based on free-licensed and open-source packages in Python, these include RDKit to dissect the molecules presented as Simplified molecular-input line-entry system (SMILES) strings with SMILES arbitrary target specification (SMARTS) definitions into the defined groups, Psycopg as the PostgreSQL adapter and Flask for the web-interface. The diagram in Figure 2.4 shows how the SAFEPROPS database (SP-DB) is accessed with the connect() function. To disconnect via Psycopg the disconnect() function is called. The database itself will be discussed further in Section 4.4. SAFEPROPS applies various functions to calculate the properties for a given chemical with the listed functions (get_groups(), filter_groups(), etc.). The API allows to let for example a model in the model library retrieve values from SAFEPROPS. Various models have been implemented such as the three flash calculation routines in reference to Biegler et al. [5] and the two oleochemical processes (spray column and crystalliser) presented in this work. The models are either written in Fortran and/or Python whereas Fortran code was wrapped with f90wrap [8] to make it accessible to Python. The models in the process simulator can be regarded as part of the model library in Figure 2.4. The Python code for setting up a connection with the PRO/II process simulator through the COM standard is found in the appendix. Different scientific packages were used to perform analysis, optimisation or machine-learning methods such as SALib [9] (sensitivity analysis), scipy [10] (optimisation) and scikit-learn [11] (machine learning). Polynomial chaos expansion (PCE) as a surrogate modelling tech-

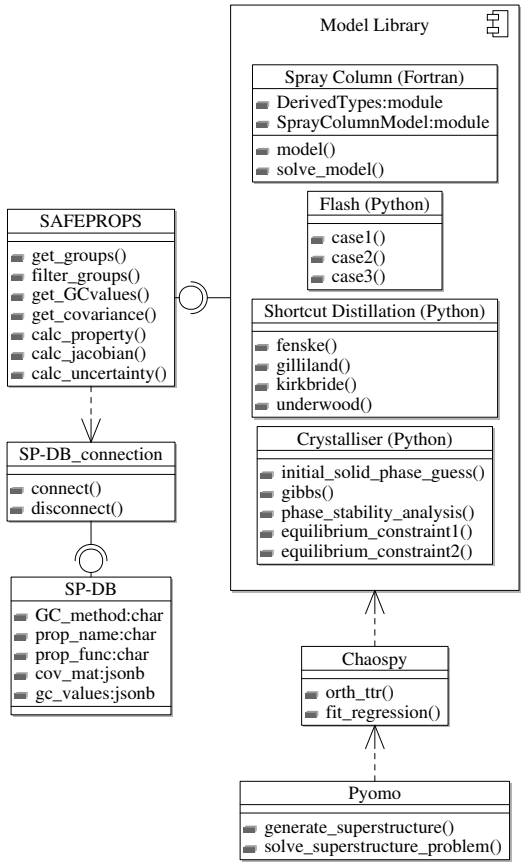


Figure 2.4: Diagram of the modular and functional elements of the multi-scale framework.

nique is supported by the packages Chaospy [12] or UQlab [13]. UQlab is a Matlab package and the scipy.io.loadmat and scipy.io.savemat functions were used to pipeline data between Matlab and Python. The surrogate model retrieved was embedded in the superstructure optimisation problem formulated with the Pyomo package.

Bibliography

[1] R. Sargent, "Process systems engineering: A retrospective view with questions for the future," *Computers & Chemical Engineering*, vol. 29, no. 6, pp. 1237–1241, 2005. DOI: https://doi.org/10.1016/j.compchemeng.2005.02.008.

[2] B. Braunschweig, C. Pantelides, H. Britt, and S. Sama, "Process modeling: The promise of open software architectures," *Chemical Engineering Progress*, vol. 96, pp. 65–76, Sep. 2000.

[3] D. D. Nikolić, "Dae tools: Equation-based object-oriented modelling, simulation and optimisation software," *PeerJ Computer Science*, vol. 2, e54, April 2016. DOI: 10.7717/peerj-cs.54.

[4] J. Hughes, "Why functional programming matters," *The Computer Journal*, vol. 32, no. 2, pp. 98–107, January 1989. DOI: 10.1093/comjnl/32.2.98.

[5] L. Biegler, I. Grossmann, and A. Westerberg, *Systematic methods of chemical process design*, ser. Prentice-Hall international series in the physical and chemical engineering sciences. Prentice Hall PTR, 1997.

[6] A. Quaglia, "An integrated business and engineering framework for synthesis and design of processing networks," English, PhD thesis, 2013.

[7] M.-O. Bertran, "Modelling, synthesis and analysis of biorefinery networks," English, PhD thesis, 2017.

[8] J. Kermode, *F90wrap*, https://github.com/jameskermode/f90wrap.

[9] J. Herman and W. Usher, "Salib: An open-source python library for sensitivity analysis," *Journal of Open Source Software*, vol. 2, no. 9, 2017.

[10] E. Jones, T. Oliphant, P. Peterson, *et al.*, *Scipy: Open source scientific tools for Python*, 2001–.

[11] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[12] J. Feinberg and H. P. Langtangen, "Chaospy: An open source tool for designing methods of uncertainty quantification," *Journal of Computational Science*, vol. 11, pp. 46–57, 2015. DOI: 10.1016/j.jocs.2015.08.008.

[13] S. Marelli and B. Sudret, "Uqlab: A framework for uncertainty quantification in matlab," in *Vulnerability, Uncertainty, and Risk*, pp. 2554–2563. DOI: 10.1061/9780784413609.257.

CHAPTER 3

The Oleochemical Process Domain

3.1 Overview

The oleochemical process domain can be divided into two branches: (1) the refining of vegetable oils and (2) the chemical conversion to intermediate and final products. This thesis focuses on the second branch where possible chemical conversion routes of vegetable oils can be transesterification, hydrolysis, aminolysis and saponification. The derived intermediates are glycerol, alkyl esters such as methyl esters, fatty acids, fatty acid amides, soaps and fatty alcohols. The platform chemicals under the oleochemicals are glycerol, fatty acid methyl ester (FAME), alcohols and amines [1, 2]. High-value substances would be for example tocols (tocopherols and tocotrienols), phospholipids, phenols, phenolic acids, ascorbic acids, chlorogenic acids and β-carotene [3–5].

These substances are usually neither available in pure form nor highly concentrated in mixtures. Further, the available mixtures may contain a high number of compounds with scarce thermodynamic and kinetic data available. A range of compounds have similar physical and chemical properties which make the desired product design challenging. Also, these compounds and products are degradable through high temperatures or oxidation processes which is a major concern of the oleochemical industry [6, 7]. Hence, the reactors and separation technology designed to recover the products from these mixtures include processes that are not commonly used in traditional chemical industries, and the process synthesis of these systems may require development of new approaches. Given the lack of highly available data, a strong integration of experimental studies with computer-aided methods is needed to address these challenges.

The chemical conversion routes of triglycerides are presented in Figure 3.1 to 3.4. We refer to the different side-chains R', R'', R''' as R^* for the products

in the schemes. Transesterification allows to obtain alkyl esters from vegetable oils by letting them react with alcohols.

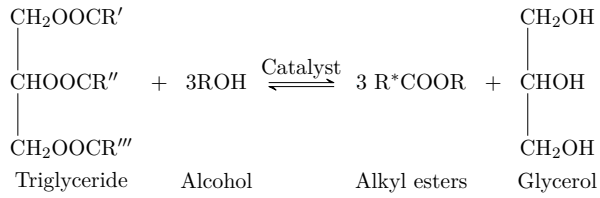


Figure 3.1: Transesterification of triglyceride with alcohol.

The hydrolysis reaction involves water reacting with the triglycerides to give fatty acids and glycerol. Diethanolamine is the reactant for the aminolysis reaction which gives fatty acid amides. The saponification of triglycerides is achieved by using a salt such as sodium hydroxide.

It can be seen that all reaction routes generate glycerol as a product (Figure 3.6). This causes a problem for the profitability of biorefineries since the increased global production of biofuels through transesterification has lead to an all time low in the price of glycerol and thereby the cost-efficient purification of glycerol has become an important task for biorefineries [8]. Also, saponification and aminolysis have become commercially less applied due to the less efficient purification of glycerol with these reaction routes [9]. Moreover, soaps and fatty acid amides are generated directly from fatty acids (Figure 3.7) or

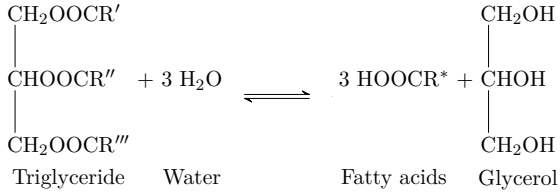


Figure 3.2: Hydrolysis of triglyceride with water.

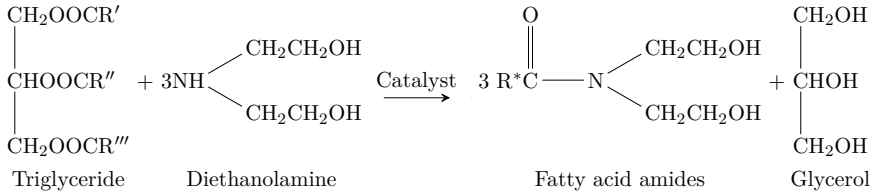


Figure 3.3: Aminolysis of triglyceride with diethanolamine.

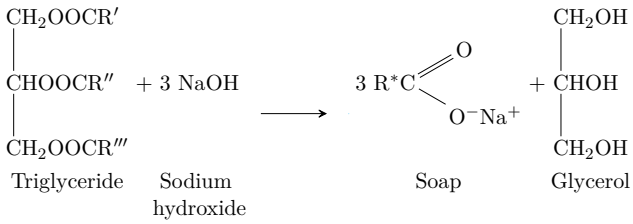


Figure 3.4: Saponification of triglyceride with sodium hydroxide.

fatty acid methyl esters via saponification or hydrogenation nowadays and not from triglycerides [9].

Palm oil is the globally most produced and has the highest yield from a palm oil field in respect to mass per area (Figure 3.5). This makes it the most processed raw material in the oleochemical domain for food application use and FAME production.

3.2 Selected oleochemical processes

The composition of vegetable oils can vary depending on the type of oil and the region where the oil plants were harvested. This makes the needed processes difficult to design because property calculations are based on the oils compositions and e.g. distillation schemes change subject to the raw materials composition and product specification. A detailed unit operation design or process design of a complete process flowsheet can vary significantly and we therefore will concentrate on research aspects for the most common processes in the oleochemical domain. Thus, in this work the conversion step via hydrolysis and the separation and purification needed for obtaining fatty acid cuts and micro-nutrients will be covered.

The spray column is an industrial scale unit operation which allows to retrieve high conversion rates for the hydrolysis step. A spray column is operated at high temperature and high pressure where the energy is provided by the high pressure steam fed at the top of the column. Although spray columns have been studied by several researchers, the hydrolysis of vegetable oil in a spray column is still not fully understood. Especially the kinetics and the hydrodynamics are aspects which should be studied to be able to provide a model for a spray column which can simulate the system with different vegetable oils and give reliable results. In Chapter 5 the foundation is laid to describe a spray column as a finite volume model (FVM) and allow to im-

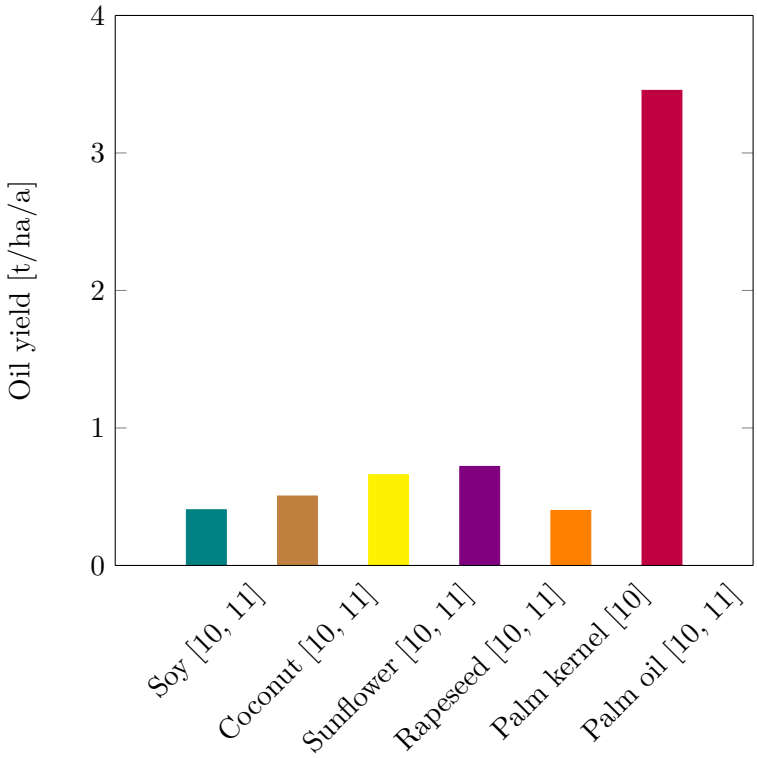


Figure 3.5: Global average of soy, coconut, sunflower, rapeseed, palm kernel and palm oil yields from one hectar field per year.

plement different kinetic models and surrogate functions from computational fluid dynamics (CFD) simulations to describe the hydrodynamics. The operating cost due to high pressurised steam makes a large contribution to the operating cost of the column.

Fatty acid separation can be achieved by a broad range of technologies and especially distillation columns are industrially applied to separate different fatty acids in respect to their chain lengths. Other technoloies are the panning and pressing method [12, 13], hydrophilisation and solvent crystallisation [12]. When it comes to separating saturated from unsaturated fatty acids, distillation columns can't achieve the desired separation. Therefore a solvent crystalliser model is presented and validated with the results by Wale [13]. Further, crystallisation units can also be combined with membranes [14].

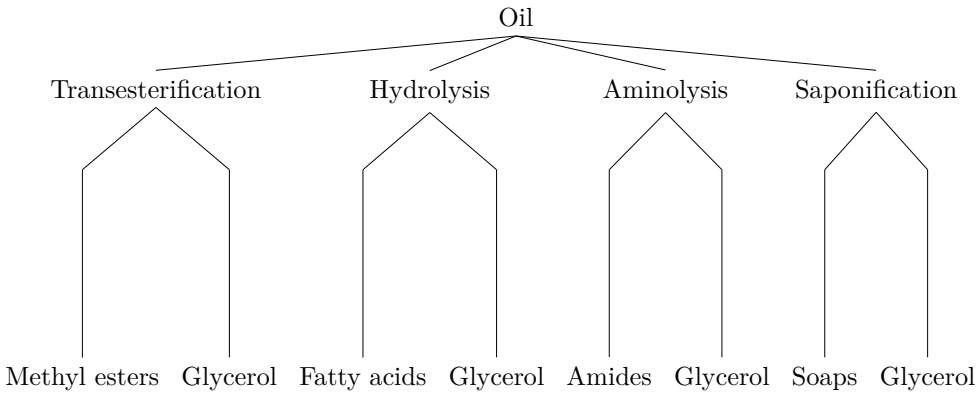


Figure 3.6: Chemical conversion routes of vegetable oils.

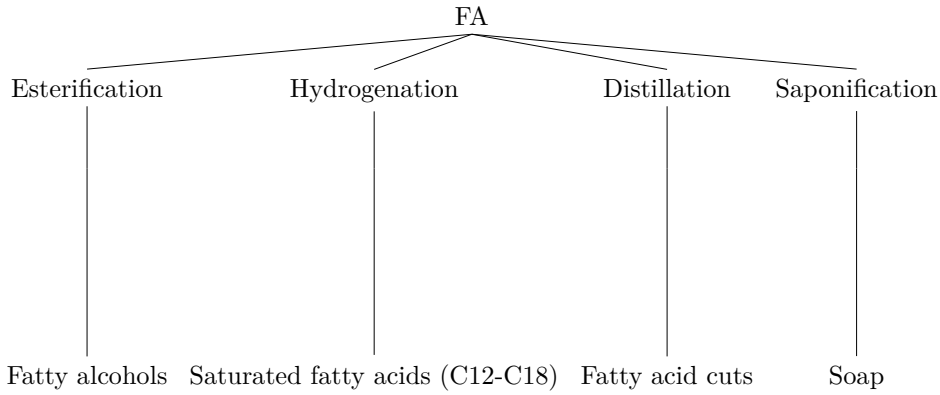


Figure 3.7: Processing routes for fatty acids (FA).

Molecular distillation allows the separation of minor compounds found in vegetable oil processing. It allows to separate valuable products e.g. micronutrients from each other. In Chapter 5 a molecular distillation unit is studied by means of sensitivity analysis to gain insights for separating α-tocopherol and β-carotene from vegetable oils where purity is an important aspect and covering uncertainties in property estimates are important in regard to achieving the desired separation and providing a robust design against uncertainty propagation. Table 3.1 gives an overview of the technology assessment for the process technologies studied in this work.

Table 3.1: Technology assessment of selected oleochemical processes.

Process technology	**Conditions**	**Performance**	**References**
Vegetable oil hydrolysis		**Conversion**	
Spray column	225-280 °C; 30-70 bar	87 - 95 %	[15–19]
PFR	240 °C; high P	83.7 %	[18]
Semi batch	230-240 °C; high P	86 %	[18]
CSTR	230-240 °C; 50 bar	65 %	[18, 20]
Batch	225-260 °C; 55 bar	84 %	[17, 18]
Enzymatic reactor	30-50 °C; ambient P; pH 5-9	36-80 %	[9, 21–26]
Fatty acid separation		**Purity**	
Thin & falling film evaporation	225-260 °C; 0.047-0.066 bar	99%	[27]
Molecular distillation	∼180 °C; ∼0.008 mbar	40-99 %	[28–35]
Solvent crystallisation	-50-0 °C; ambient P	> 90 %	[12–14, 36, 37]
Glycerol purification		**Purity**	
Vacuum distillation	120-125 °C; < pH 5	>96%	[8, 38]
Ion exchange	∼300 °C	95-99%	[8, 39]
Membrane separation technology	dependent on membrane type	90-99%	[8, 14]

Enzymatic hydrolysis is being researched [23–26] to reduce the economic cost and ecological footprint of the hydrolysis of triglycerides. Decentralised production could be the answer to make enzymatic processing of vegetable oils feasible. The enzymatic reactors are located geographically with the goal to improve the transportation of the pre-hydrolysed oil to the main plant equipped with a spray column and optimise the economic and ecological performance of such a production setup. This process network has to then be compared to a single production plant setup. Studies have already been made in respect to biofuel production [40, 41]. A single, centralised configuration (Figure 3.8) would be the installation of an enzymatic reactor before a spray column. This would reduce the operating cost of the spray column since the vegetable oil has already been hydrolysed by about 30 %.

The third possible process configuration would be the installation of a single enzymatic reactor (Figure 3.9) or reactors in series with or without a

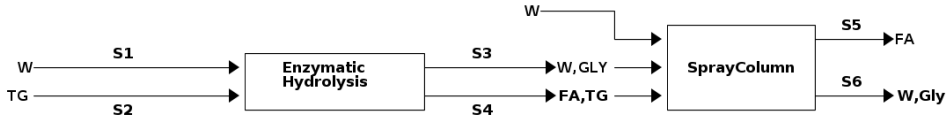


Figure 3.8: Enzymatic pre-hydrolysis and full hydrolysis with a spray column.

recycle of the separated triglyceride stream back to the first reactor. Here the high cost of the enzymes need to be evaluated and reduced.

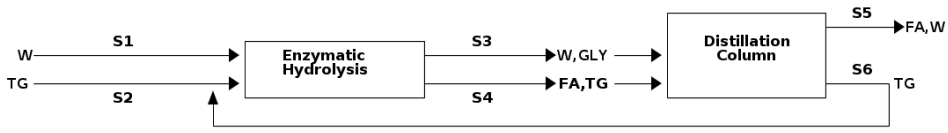


Figure 3.9: Enzymatic hydrolysis with recycling the triglycerides to obtain full conversion.

In Chapter 6 the first step is made to evaluate these process structures against each other by applying superstructure optimisation with surrogate models. A subsequent research topic would be the optimisation of such decentralised biorefinery supply networks [42].

Bibliography

[1] R. Kumar, I. Sundari, S. Sen, N. Dasgupta, and R. Chidambaram, "Chapter 19 – animal fat– and vegetable oil–based platform chemical biorefinery," in *Platform Chemical Biorefinery*, S. K. Brar, S. J. Sarma, and K. Pakshirajan, Eds., Amsterdam: Elsevier, 2016, pp. 361–377. DOI: 10.1016/B978-0-12-802980-0.00019-5.

[2] Q. Yan and B. F. Pfleger, "Revisiting metabolic engineering strategies for microbial synthesis of oleochemicals," *Metabolic Engineering*, 2019. DOI: https://doi.org/10.1016/j.ymben.2019.04.009.

[3] H. Schwartz, V. Ollilainen, V. Piironen, and A. Lampi, "Tocopherol, tocotrienol and plant sterol contents of vegetable oils and industrial fats," *Journal of Food Composition and Analysis*, vol. 21, no. 2, pp. 152–161, 2008. DOI: 10.1016/j.jfca.2007.07.012.

[4] C. Janu, D. S. Kumar, M. Reshma, P. Jayamurthy, A. Sundaresan, and P. Nisha, "Comparative study on the total phenolic content and radical scavenging activity of common edible vegetable oils," *Journal of Food Biochemistry*, vol. 38, no. 1, pp. 38–49, 2014. DOI: 10.1111/jfbc.12023.

[5] E. M. Hernandez, "4 – Specialty Oils: Functional and Nutraceutical Properties," in *Functional Dietary Lipids*, ser. Woodhead Publishing Series in Food Science, Technology and Nutrition, T. A. Sanders, Ed., Woodhead Publishing, 2016, pp. 69–101. DOI: 10.1016/B978-1-78242-247-1.00004-1.

[6] S. Cremaschi, "A perspective on process synthesis – challenges and prospects," *Computers & Chemical Engineering*, vol. 81, pp. 130–137, 2015. DOI: 10.1016/j.compchemeng.2015.05.007.

[7] C. Malwade, H. Qu, B.-G. Rong, and L. P. Christensen, "Conceptual process synthesis for isolation and purification of natural products from plants - a case study of artemisinin from artemisia annua," in *11th International Symposium on Process Systems Engineering*, ser. Com-

puter Aided Chemical Engineering, vol. 31, 2012, pp. 1707–1711. DOI: 10.1016/B978-0-444-59506-5.50172-3.

[8] M. Ardi, M. Aroua, and N. A. Hashim, "Progress, prospect and challenges in glycerol purification process: A review," *Renewable and Sustainable Energy Reviews*, vol. 42, pp. 1164–1173, 2015. DOI: 10.1016/j.rser.2014.10.091.

[9] D. J. Anneken, S. Both, R. Christoph, G. Fieg, U. Steinberner, and A. Westfechtel, "Fatty acids," in *Ullmann's Encyclopedia of Industrial Chemistry*. American Cancer Society, 2006. DOI: 10.1002/14356007.a10_245.pub2.

[10] T. T. Sue and T. P. Pantzaris, *Pocketbook of Oil Palm Uses*. Malaysian Palm Oil Board, 2017.

[11] S. Noleppa and M. Cartsburg, *Auf der Ölspur - Berechnungen zu einer palmölfreien Welt*. WWF Deutschland, 2016.

[12] G. Haraldsson, "Separation of saturated/unsaturated fatty acids," *Journal of the American Oil Chemists' Society*, vol. 61, no. 2, pp. 219–222, February 1984. DOI: 10.1007/BF02678772.

[13] S. N. Wale, "Separation of fatty acids by extractive crystallization," English, PhD thesis, 1995.

[14] H.-Y. Wang, K.-L. Tung, and J. D. Ward, "Design and economic analysis of membrane-assisted crystallization processes," *Journal of the Taiwan Institute of Chemical Engineers*, vol. 81, pp. 159–169, 2017. DOI: https://doi.org/10.1016/j.jtice.2017.09.023.

[15] G. V. Jeffreys, V. G. Jenson, and F. R. Miles, "The analysis of a continiuous fat-hydrolysing column," *Transactions of the Institution of Chemical Engineers*, vol. 39, no. nil, pp. 389–396, 1961.

[16] Rifai, Elnashaie, and Kafafi, "Analysis of a countercurrent tallow splitting column," *Trans. Instn. Chem. Eng*, vol. 55, pp. 59–63, 1977.

[17] T. A. Patil, D. N. Butala, T. S. Raghunathan, and H. S. Shankar, "Thermal hydrolysis of vegetable oils and fats. 1. reaction kinetics," *Industrial Engineering Chemistry Research*, vol. 27, no. 5, pp. 727–735, 1988.

[18] P. D. Namdev, T. A. Patil, T. S. Raghunathan, and H. S. Shankar, "Thermal hydrolysis of vegetable oils and fats. 3. an analysis of design alternatives," *Industrial Engineering Chemistry Research*, vol. 27, pp. 739–743, 1988.

[19] M. Attarakih, T. Albaraghthi, M. Abu-Khader, Z. Al-Hamamre, and H. Bart, "Mathematical modeling of high-pressure oil-splitting reactor using a reduced population balance model," *Chemical Engineering Science*, vol. 84, pp. 276–291, 2012.

[20] T. A. Patil, T. S. Raghunathan, and H. S. Shankar, "Thermal hydrolysis of vegetable oils and fats. 2. hydrolysis in continuous stirred tank reactor," *Industrial Engineering Chemistry Research*, vol. 27, no. 5, pp. 735–739, 1988. DOI: 10.1021/ie00077a002.

[21] V. R. Murty, J. Bhat, and P. K. A. Muniswaran, "Hydrolysis of oils by using immobilized lipase enzyme: A review," *Biotechnology and Bioprocess Engineering*, vol. 7, no. 2, pp. 57–66, April 2002. DOI: 10.1007/BF02935881.

[22] M. Vacek, M. Zarevúcka, Z. Wimmerb, K. Stránský, B. Koutek, M. Macková, and K. Demnerová, "Lipase-mediated hydrolysis of blackcurrant oil," *Enzyme and Microbial Technology*, vol. 27, no. 7, pp. 531–536, 2000. DOI: 10.1016/S0141-0229(00)00239-8.

[23] Z.-M. He, J.-C. Wu, C.-Y. Yao, and K.-T. Yu, "Lipase-catalyzed hydrolysis of olive oil in chemically-modified aot/isooctane reverse micelles in a hollow fiber membrane reactor," *Biotechnology Letters*, vol. 23, no. 15, pp. 1257–1262, August 2001. DOI: 10.1023/A:1010589412271.

[24] V. R. Murty, J. Bhat, and P. Muniswaran, "Hydrolysis of rice bran oil using an immobilized lipase from candida rugosa in isooctane," *Biotechnology Letters*, vol. 26, no. 7, pp. 563–567, April 2004. DOI: 10.1023/B:BILE.0000021956.33855.11.

[25] L. Freitas, T. Bueno, V. H. Perez, J. C. Santos, and H. F. de Castro, "Enzymatic hydrolysis of soybean oil using lipase from different sources to yield concentrated of polyunsaturated fatty acids," *World Journal of Microbiology and Biotechnology*, vol. 23, no. 12, pp. 1725–1731, December 2007. DOI: 10.1007/s11274-007-9421-8.

[26] D. Goswami, J. K. Basu, and S. De, "Lipase applications in oil hydrolysis with a case study on castor oil: A review," *Critical Reviews in Biotechnology*, vol. 33, no. 1, pp. 81–96, 2013. DOI: 10.3109/07388551.2012.672319.

[27] H. Stage, "Fatty acid fractionation by column distillation: Purity, energy consumption and operating conditions," *Journal of the American Oil Chemists' Society*, vol. 61, no. 2, pp. 204–214, February 1984. DOI: 10.1007/BF02678770.

[28] J. Lutišan and J. Cvengroš, "Mean free path of molecules on molecular distillation," *The Chemical Engineering Journal and the Biochemical Engineering Journal*, vol. 56, no. 2, pp. 39–50, 1995. DOI: https://doi.org/10.1016/0923-0467(94)02857-7.

[29] U. R. Unnithan, *Refining of edible oil rich in natural carotenes and vitamin E*, Patent, US, August 1999.

[30] S. C. Cermak, R. L. Evangelista, and J. A. Kenar, "Distillation of natural fatty acids and their chemical derivatives," in *Distillation*, S. Zereshki, Ed., 2012, ch. 5. DOI: 10.5772/38601.

[31] C. Batistella, E. Moraes, R. Maciel, and M. Maciel, "Molecular distillation - rigorous modeling and simulation for recovering vitamin e from vegetable oils," *Applied Biochemistry and Biotechnology*, vol. 98, pp. 1187–1206, 2002.

[32] E. B. de Moraes, P. F. Martins, C. B. Batistella, M. E. T. Alvarez, R. M. Filho, and M. R. W. Maciel, "Molecular distillation," in *Twenty-Seventh Symposium on Biotechnology for Fuels and Chemicals*. Humana Press, 2006, pp. 1066–1076. DOI: 10.1007/978-1-59745-268-7_90.

[33] P. Martins, V. Ito, C. Batistella, and M. Maciel, "Free fatty acid separation from vegetable oil deodorizer distillate using molecular distillation process," *Separation and Purification Technology*, vol. 48, no. 1, pp. 78–84, 2006. DOI: https://doi.org/10.1016/j.seppur.2005.07.028.

[34] S. Wang, Y. Gu, Q. Liu, Y. Yao, Z. Guo, Z. Luo, and K. Cen, "Separation of bio-oil by molecular distillation," *Fuel Processing Technology*, vol. 90, no. 5, pp. 738–745, 2009. DOI: 10.1016/j.fuproc.2009.02.005.

[35] N. Tehlah, P. Kaewpradit, and I. M. Mujtaba, "Development of molecular distillation based simulation and optimization of refined palm oil process based on response surface methodology," *Processes*, vol. 5, no. 3, 2017. DOI: 10.3390/pr5030040.

[36] R. L. Demmerle, "Emersol process: A staff report," *Industrial & Engineering Chemistry*, vol. 39, no. 2, pp. 126–131, 1947. DOI: 10.1021/ie50446a011.

[37] K. T. Zilch, "Separation of fatty acids," *Journal of the American Oil Chemists' Society*, vol. 56, no. 11Part1, 739A–742A, 1979. DOI: 10.1007/BF02667432.

[38] K. Yong, T. Ooi, K. Dzulkefly, W. Yunus, and H. Hassan, "Refining of crude glycerine recovered from glycerol residue by simple vacuum distillation," *Journal of Oil Palm Research*, vol. 13, pp. 39–44, January 2001.

[39] M. Carmona, J. L. Valverde, A. Pérez, J. Warchol, and J. F. Rodriguez, "Purification of glycerol/water solutions from biodiesel synthesis by ion exchange: Sodium removal part i," *Journal of Chemical Technology & Biotechnology*, vol. 84, no. 5, pp. 738–744, 2009. DOI: 10.1002/jctb.2106.

[40] G. Perkins, T. Bhaskar, and M. Konarova, "Process development status of fast pyrolysis technologies for the manufacture of renewable transport fuels from biomass," *Renewable and Sustainable Energy Reviews*, vol. 90, pp. 292–315, 2018. DOI: 10.1016/j.rser.2018.03.048.

[41] N. Dahmen, J. Abeln, M. Eberhard, T. Kolb, H. Leibold, J. Sauer, D. Stapf, and B. Zimmerlin, "The bioliq process for producing synthetic transportation fuels," *Wiley Interdisciplinary Reviews: Energy and Environment*, vol. 6, no. 3, e236, 2017. DOI: 10.1002/wene.236.

[42] L. Čuček, M. Martín, I. E. Grossmann, and Z. Kravanja, "Multi-period synthesis of optimally integrated biomass and bioenergy supply network," *Computers & Chemical Engineering*, vol. 66, pp. 57–70, 2014. DOI: 10.1016/j.compchemeng.2014.02.020.

CHAPTER 4

Data-driven and Stochastic Property Prediction

4.1 Vegetable oils, oleochemicals and important properties

Vegetable oils are mostly mixtures of triglycerides with even length fatty acid side-chains. Traces of components such as tocopherol, carotenes and impurities can be found in vegetable oils. But usually vegetable oils are characterised by their triglycerides content as shown in Table 4.1 [1]. Simple triglycerides can be described with the three equivalent fatty acid side chains which are composed of the same number of C atoms and the occurrence of double bonds (DB) in the chain. Thus the notation of simple triglycerides and fatty acids can be given with (C:DB) [2]. Natural triglycerides can differ in the individual side-chain lengths and Zong et al. present a more detailed description method [3] for these components. Chang and Liu [4] present three different ways how to define the triglyceride mixtures to characterise vegetable oils and calculate their properties. Namely, the 'mixed triglycerides', 'simple triglycerides' and 'pseudo-triglyceride' approach. The first approach defines the triglyceride side-chains as fragments and decomposes a triglyceride mixture in these specific fragments. Properties can then be calculated with correlations and the regressed values for the fragments. The 'mixed triglyceride approach' calculates the properties for the mixture from the individual properties of each triglyceride, in this case the triglycerides are not dissected further in any fragments. The third method defines a pseudo-triglyceride for the whole mixture by using the percentage of the side-chains of each present triglyceride. Two schemes are presented by Chang and Liu to then calculate the properties with the

Table 4.1: Composition of vegetable oils.

Triglyceride (C:DB)	Sunflower oil	Soybean oil	Palm oil	Coconut oil
Caprylic (8:0)	-	-	-	6.21 ± 0.34
Capric (10:0)	-	-	-	6.15 ± 0.21
Lauric (12:0)	-	-	-	51.02 ± 0.71
Myristic (14:0)	-	-	1.23 ± 0.28	18.94 ± 0.63
Palmitic (16:0)	6.52 ± 1.75	14.04 ± 0.62	41.78 ± 1.27	8.62 ± 0.50
Stearic (18:0)	1.98 ± 1.44	4.07 ± 0.29	3.39 ± 0.65	1.94 ± 0.17
Oleic (18:1)	45.39 ± 18.77	23.27 ± 2.43	41.90 ± 1.20	5.84 ± 0.50
Linoleic (18:2)	46.02 ± 16.75	52.18 ± 2.64	11.03 ± 0.02	1.28 ± 0.18
Linolenic (18:3)	0.12 ± 0.09	5.63 ± 3.48	-	-

pseudo-triglyceride. Scheme A adapts a group-contribution method and constructs the pseudo-molecule with CH_2 and CH=CH groups and then predicts the property. Scheme B predicts the property of the mixture from the fatty acid chains percentages.

Su et al. [5] give an overview for thermophysical prediction methods relevant for biodiesel manufacturing. For vegetable oils they evaluate the methods for predicting liquid density, vapour pressure, liquid heat capacity and heat of vapourization and perform a comparison. They compare the three vegetable oil characterisation methods from Chang and Liu combined with different property prediction methods and recommend to apply the mixed triglycerides approach. If the vegetable oils composition is only provided as a composition of fatty acids in literature then the simple triglyceride or pseudo-triglyceride approach can be used. The recommended mixed triglyceride approach describes a triglyceride in four fragments as developed by Zong et al. [3]. In Figure 4.1 a general triglyceride structure is shown where the individual fatty acid side-chains R', R'' and R''' are regarded as one fragment each and the glycerol backbone as the fourth fragment.

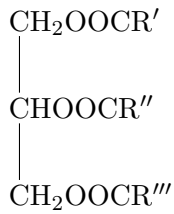


Figure 4.1: General triglyceride structure.

Wallek et al. [6] applied a group contribution method which dissects triglycerides or fatty acids in the structural groups listed in Table 4.2. They extended the group contribution model by Nannoolal et al. [7] with describing the backbone of the triglyceride similar to the fourth fragment by Zong et al. However, the side-chains can be described with the individual structural groups listed in Table 4.2 and are not regarded as single fragments. Wallek et al. report an improvement of the predictive performance for normal boiling point and viscosity estimations with the extended group contribution model. A similar approach has also been applied by Damaceno et al. [8] who extended the modified UNIFAC model with a new glycerol backbone group to improve the performance of phase equilibrium predictions.

Table 4.2: Marrero-Gani first order groups and the updated structural groups set for describing TG, FA and FAME as proposed by Wallek et al. extending the Nannoolal/Rarey/Ramjugernath (NRR) GC model.

Marrero-Gani	Wallek et al. (NRR ID)	Description
COO	(C)-COO-(C) (45)	-COO- connected to two C in a chain (ester)
	=C= + (F ∨ Cl ∨ N ∨ O) (7)	C-atom in a chain connected to at least one F, Cl, N or O atom
	-COO-COO- (189)	Group interaction between two esters
CH		CH UNIFAC subgroup
CH_2	-CH_2- (4)	-CH_2- in a chain
CH_3	-CH_3 (1)	Methyl group not attached to either F, Cl, N or O atom
	GLY	Glycerol backbone
aC		aromatic C
aC-CH_3		aromatic C connected to CH3
aC-OH		aromatic C connected to OH
CH_2(cyc)		CH2 in a ring
C(cyc)		C in a ring
O(cyc)		O in a ring

To this end one can conclude that the structural information of a molecule can improve the predictive performance of a model. Therefore, the next sections will describe current developments with respect to group contribution methods combined with uncertainty analysis and how to combine structural information with the machine learning method known as Gaussian process regression. This allows a purely data-driven model generation where only the experimental values of a certain property and the structural information of the molecules is used as the training data.

4.2 Uncertainty analysis of group contribution methods for property prediction

In this section the Marrero-Gani (MG) group contribution method [9] is discussed with respect to the studies of oleochemical property data and uncertainty analysis. The right hand side of the equation for the MG method is:

$$f(x) = \sum_j M_j C_j + w \sum_k N_k D_k + z \sum_l O_l E_l \tag{4.1}$$

C_j, D_k and E_l are the group contribution values for the first, second and third order groups respectively and M_j, N_k and O_l account for the number of these groups present in the molecular structure of the substance at hand. w and z are the weighting factors for the second and third order groups utilised for the parameter estimation procedure explained later. w and z are set to unity after the group contribution values have been obtained. The left hand side $f(x)$ is the functional representation of the target property x to be predicted. Diaz-Tovar [10] has conducted the regression of MG group contribution values for the 1st, 2nd and 3rd order groups relevant to oleochemicals. Hukkerikar [11] developed a methodology for uncertainty analysis of the MG method. The uncertainty analysis is explained in the following where the MG method is defined in matrix-vector form:

$$f(x) = T\theta \tag{4.2}$$

with T being the occurence matrix of the functional groups (M_j, N_k, O_l) available in the MG method and θ stores the regressed group contribution (GC) values (C_j, D_k, E_l) of each functional group contributing to the property estimation. x is the target property represented by a function which achieves the best possible fit of the experimental data [10]. A collection of these functions for different properties can be found in the works by Diaz-Tovar [10], Constantinou and Gani [12], Hukkerikar et al. [11], and Frutiger et al. [13].

Frutiger et al. [14] compared three regression methods combined with and without outlier treatment, namely ordinary least squares, robust regression and weighted least-squares. Results showed that an outlier treatment should be performed for all regression methods, whereas ordinary-least squares gave best performance statistics for the standard deviation (SD), robust regression

for the average absolute deviation (AAD) and average relative error (ARE), and weighted least-squares for the sum of squared errors (SSE).

From the regression procedure the covariance matrix is obtained for the regressed group contribution values. The uncertainty of the parameter estimates is based on the asymptotic approximation of the covariance matrix COV(θ^*):

$$COV(\theta^*) = \frac{SSE}{n-p}(J(\theta^*)^T J(\theta^*))^{-1} \tag{4.3}$$

where n is the number of points in the data set, p the number of estimated parameters and J(θ) is the Jacobian of the estimated parameters. The parameter estimates are described by a Student's t-distribution:

$$\theta^*_{1-\alpha} = \theta \pm \sqrt{diag(COV(\theta^*))} * t(n-p, \alpha_t/2) \tag{4.4}$$

A first guess for the a priori unknown GC factors is provided by:

$$\hat{\theta} = (T^T T)^{-1} * T^T * f(x) \tag{4.5}$$

A sequential parameter estimation procedure is then applied to estimate the 1st, 2nd and 3rd group contribution values. First w and z in Equation 4.1 are set to zero to estimate the first order group contribution values. Next, w is set to unity and z to zero for estimating the second order group contribution values. And last, w and z are both 1 when the third order group contribution values are estimated. Frutiger et al. introduced a simultaneous parameter estimation step to receive a final estimation of the parameters and to achieve a global optimal solution. The estimates from the sequential estimation step are used as an initial guess for the simultaneous estimation. The final right hand side of the MG equation can then be written with w and z set to unity:

$$y^{pred} = \sum_j M_j C_j + \sum_k N_k D_k + \sum_l O_l E_l \tag{4.6}$$

The confidence interval of the property predictions are calculated with the following equation:

$$y^{pred}_{1-\alpha} = y^{pred} \pm \sqrt{diag(J(\theta^*)COV(\theta^*)J(\theta^*)^T)} * t(n-p, \alpha_t/2) \tag{4.7}$$

The methodology is summarised in Table 4.3 and based on the work by Frutiger et al. [14, 15] who extended and refined the methodology by Hukkerikar et al. [11] and showed that the 2nd and 3rd order groups give only marginally

improved 95% confidence boundaries in comparison to only using 1st order groups. The relative error between the prediction and experimental values was even worse in the case of cis,trans-2,4-hexadiene and improved for acrolein from 0.0094 to 0.0051 when adding second and third groups to the prediction model [14]. Therefore this work investigates the combination of Marrero-Gani first and second order groups with and without molecular descriptors applied in quantitative structure-property relationship (QSPR) models. This evaluation is a preliminary study on gaining more insight on the structural information for prediction purposes and how group contribution models could be enhanced with other features instead of second order groups with respect to the MG method.

Table 4.3: Methodology for parameter estimation and uncertainty analysis of group-contribution methods.

#	Step	Description
1	Experimental data & structural information of molecule	The experimental data is the target (output) and the occurence matrix is the input to the fitting process
2	Definition of LHS and RHS & and predict property with initial GC values	The predicted property (LHS) and the model function (RHS) is defined. Initial guesses for the GC values have to be set
3	Regression	The GC values are estimated for example with robust regression and the performance of the estimation is evaluated (e.g. ARE)
4	Outlier treatment	Outlier treatment via Cock's distance, normal distribution or empirical cumulative distribution
5	Uncertainty based on covariance matrix	Covariance of prediction is approximated with Jacobian and covariance of the estimates
6	Parameter identifiability analysis	Analyse linear correlation coefficients and parameter estimation errors
7	Fitted model can be applied	The obtained model can be used to predict the property of a compound with the occurence matrix and the derived GC values. The uncertainty is calculated from the covariance

4.3 Property prediction with Gaussian process regression and molecular descriptors

Gaussian process regression (GPR) is a machine learning method which can be applied to property prediction. Obrezanova et al. [16] applied GPR to predict absorption, distribution, metabolism and excretion (ADME) properties. Four data sets (a benzodiazepine set with 245 compounds, a blood-brain

barrier data set, a hERG data set and a solubility at pH 7.4 data set) were used to conduct the study. Various Gaussian process models were compared: fixed hyperparameters, hyperparameters obtained by forward variable selection, conjugate gradient optimization and nested sampling. Mondejar et al. [17, 18] implemented a neural network to predict physical properties (T_C, P_C, ω, T_b, $c_{p,0}$) for halogenated organic chemicals. The trained neural network performs for all properties better than the MG and Joback and Reid method. In this work a similar procedure to train a Gaussian process is applied and the steps are listed in Table 4.4. Sola et al. [19] use QSPR to describe the structure of a molecule and to predict the properties boiling point, critical temperature and critical pressure of organic compounds. Since they utilise a linear in the parameters model [20], a general regression model can be expressed with the selected molecular descriptors stored in vector $\boldsymbol{a}$ multiplied with the individual regression factor $\boldsymbol{\beta}$ and an added constant v:

$$\hat{f}(x) = \boldsymbol{a}^T \boldsymbol{\beta} + v \tag{4.8}$$

where $v = \beta_1$ for the linear in the parameters model. The feature vectors $\boldsymbol{a}$ and regression factor vector is given by:

$$\boldsymbol{a}^T = [f_2(x), ..., f_p(x)]; \quad f_1(x) = 1 \tag{4.9}$$

$$\boldsymbol{\beta} = [\beta_2, ..., \beta_p] \tag{4.10}$$

Sola et al. developed a heuristic algorithm to iterate through a database of molecular descriptors and select the descriptors with the best cross-validation statistics with respect to R^2. They show that each property has its individual set of molecular descriptors which give the best prediction results. The two molecular descriptors with the highest correlation in respect to the boiling point of the data set of organic compounds are: the cubic root of the gravitational index and a surface area descriptor ($HDCA_2$/TMSA). The importance of these descriptors to predicting the normal boiling point is also shown by Katritzky et al. [21]. Lam et al. [22] use molecular descriptors to train Gaussian process models with a skin permeability dataset for predicting the percutaneous absorption and apply automatic relevance determination (ARD) to select the highest correlated molecular descriptors. Banchero et al. [23] apply Multi-Linear and Radial-Basis-Function-Neural-Networks (RBFNN) with QSPR predictors to predict T_C, P_C and ω of organic compounds.

Gaussian process regression, due to its stochastic nature, provides confidence bounds for the estimates and neither initial estimates of GC values nor

the individual model function for each property is needed. Instead the parametric structure of the covariance function has to be selected for the data at hand. Bearing this in mind, a distribution over functions with prior information on the real process f(x) is defined as:

$$f(x) \sim GP(\mu(x), k(x, x')) \tag{4.11}$$

and a posterior predictive distribution at x relates to:

$$f(x) \sim N(\mu(x), \sigma^2(x)) \tag{4.12}$$

Thus, Gaussian processes can be described as a linear combination of polynomials and basis functions, where we introduce a mean function μ for the polynomial and a covariance function (also known as kernel) $k(x_i, x_j) = \mathrm{Cov}[f(x_i), f(x_j)]$ for the basis part, to describe a normal distribution over a set of functions:

$$f(x^*) = \mu(\boldsymbol{x^*}) + \boldsymbol{r}_*^T \boldsymbol{K}^{-1}(\boldsymbol{f} - \boldsymbol{1\mu}) \tag{4.13}$$

$\boldsymbol{f}$ is the vector for the observations made at points x_i in $\boldsymbol{x}$ and $[\boldsymbol{r}_*]_i = k(x_i, x^*)$ is the vector of correlations between the current prediction (test) point x^* and the previously sampled data points x_i [24–26]. The mean vector is $[\boldsymbol{\mu}]_i = \mu(x_i)$ and the covariance function is $[\boldsymbol{K}]_{ij} = k(x_i, x_j)$. $\boldsymbol{K}$ is also referred to as the Gram matrix. The joint distribution of noise-free observations can be described with equation (4.13) and the covariance function as:

$$\begin{bmatrix} \boldsymbol{f} \\ f(x^*) \end{bmatrix} \sim N\left(\begin{bmatrix} \boldsymbol{\mu} \\ \mu(x^*) \end{bmatrix}, \begin{bmatrix} \boldsymbol{K} & k(x, x^*) \\ k(x^*, x) & k(x^*, x^*) \end{bmatrix} \right) \tag{4.14}$$

where $\boldsymbol{f}$ is the vector of observations at the previously sampled data points $\boldsymbol{f} = [f(x_1), ..., f(x_n)]^T$. The variance is:

$$\sigma_*^2 = k(x^*, x^*) - \boldsymbol{k}_*^T \boldsymbol{K}^{-1} \boldsymbol{k}_* \tag{4.15}$$

It is common practice to define the mean function(s) as zero, since noise (uncertainty) can be taken into account by modifying the kernel. The joint distribution of noisy-observations can be written as:

$$\begin{bmatrix} \boldsymbol{f} + \boldsymbol{\epsilon} \\ f(x^*) \end{bmatrix} \sim N\left(\begin{bmatrix} \boldsymbol{\mu} \\ \mu(x^*) \end{bmatrix}, \begin{bmatrix} \boldsymbol{K} + \sigma_n^2 \boldsymbol{I} & k(x, x^*) \\ k(x^*, x) & k(x^*, x^*) \end{bmatrix} \right) \tag{4.16}$$

and the predictive mean and variance are respectively:

$$\mu^* = \mu(x^*) + \boldsymbol{k}^T (\boldsymbol{K} + \sigma_n^2 \boldsymbol{I})^{-1} (\boldsymbol{f} + \boldsymbol{\epsilon} - \boldsymbol{1\mu}) \tag{4.17}$$

$$\sigma_*^2 = k(x^*, x^*) - \boldsymbol{k}_*^T(\boldsymbol{K} + \sigma_n^2\boldsymbol{I})^{-1}\boldsymbol{k}_* \tag{4.18}$$

The predictive mean $\hat{\mu}$, variance and log marginal likelihood (LML) are obtained through matrix inversion using Cholesky factorization formulated with the Algorithm 2.1 in the book 'Gaussian Processes for Machine Learning' by Rasmussen and Williams [27]. As already mentioned before, the choice of the kernel $k(x_i, x_j)$ is an important decision to obtain a good fit to the input-output data. The radial basis covariance function (also known as squared exponential kernel) can be regarded as a universal kernel [28] and is defined as follows:

$$k(x_i, x_j) = \sigma_f^2 exp(-\frac{1}{2}\sum_{m=1}^{N}\frac{|x_{i,m} - x_{j,m}|^2}{l_m^2}) \tag{4.19}$$

where σ_f denotes the signal variance and l_m the length scale(s) [27, 29] for the specific feature m in the dimensional space N. These parameters are called hyperparameters and are subject to optimisation while fitting the GP to the training data. In the anisotropic case, where the length-scale hyperparameter $\boldsymbol{l}$ is a vector with the size of the number of features (molecular descriptors), the inverse of $\boldsymbol{l}$ determines the correlation of the individual molecular descriptors. For a large l_m value along a specific input axis the covariance will not be correlated to the referenced molecular descriptor and thus remove it from the inference [27]. This is referred to as ARD [30, 31] and allows to identify the relevant features for predicting a property. The kernel is describing the correlation between two outputs and for noisy observations one can write:

$$\mathrm{COV}(y_i, y_j) = k(x_i, x_j) + \sigma_n^2\delta_{i,j}; \quad \mathrm{COV}(\boldsymbol{y}) = \boldsymbol{K} + \sigma_n^2\boldsymbol{I} \tag{4.20}$$

The noise variance σ_n^2 is also a hyperparameter. $\delta_{i,j}$ is the Kronecker delta and $\boldsymbol{I}$ the identity matrix.

The hyperparameters can be tuned with different methods such as grid search, random search or the expected improvement (EI) algorithm [32]. Grid search and random search are computationally expensive if the number of hyperparameters is high. The EI algorithm finds the next point for training the Gaussian process which results in the highest expected improvement of the model performance:

$$EI(x) = \mathbb{E}[max(0, f(x) - f(\hat{x}))] \tag{4.21}$$

The underlying functional description of EI(x) is [33, 34]:

$$EI(x) = \begin{cases} (\mu(x) - f(\hat{x}))\phi(Z) + \sigma(x)\Phi(Z), & \text{if } \sigma(x) > 0 \\ 0, & \text{if } \sigma(x) = 0 \end{cases} \tag{4.22}$$

where $\phi(Z)$ and $\Phi(Z)$ are the standard normal distribution function and density function. Z is:

$$Z = \frac{\mu(x) - f(\hat{x})}{\sigma(x)} \tag{4.23}$$

This functional description is also known as the acquisition function as a general description for making a trade-off between exploration and exploitation of the hyperparameters' search space. Other acquisition functions are MPI (maximum probability of improvement) and UCB (upper confidence bound) [32].

The methodology to train a Gaussian process for property prediction with molecular descriptors is summarised in Table 4.4. The occurence vector is referred to as a molecular descriptor for a chemical compound where the occurence matrix is the collection of occurence vectors for a set of chemicals. Diaz-Tovar [10] provides estimates of pure component properties for more

Table 4.4: Methodology for training a Gaussian process with combining the functional groups from GC methods and molecular descriptors from QSPR methods as the features to the fitting process.

#	Step	Description
1	Cluster experimental data	Classify experimental data with respect to chemical structures
2	Split experimental data	Divide the experimental data (targets) into a training and testing set
3	Structural information of molecule	Create occurrence matrix of first order groups and evaluate through literature research which relevant molecular descriptors should be chosen with respect to the predicted property
4	Train the Gaussian process	Train the Gaussian process with the training set
5	Test and analyse the trained Gaussian process	Report R^2, MSE, $RMSE$ for predictive performance and analyse which features are important with the elements of the length scale vector l
6	Trained Gaussian process can be applied	If the accuracy is satisfactory the predictive model can be used to predict the property of a compound with the set of important molecular descriptors. The uncertainty of the prediction is provided by the covariance function.

than 200 oleochemical compounds (Tri-, di- and monoglycerides, fatty acids

and fatty esters). Due to a confidentiality agreement the experimental values couldn't be disclosed by Diaz-Tovar. The estimated values in the fatty acid and triglyceride data sets were used to show how a Gaussian process can be trained and provide confidence bounds on the predictions. The methodology is exemplified on the normal boiling point property and molecular descriptors used are: the MG first and second order group occurrences, the number of C atoms in the molecule, the cubic root of the gravitational index ($\sqrt[3]{\mathrm{GI}}$) and 9 surface area descriptors available in the Python package mordred by Moriwaki et al. [35]. The simplified GC method has the following form excluding third order groups:

$$f(x) = \sum_j M_j C_j + \sum_k N_k D_k \tag{4.24}$$

The predictive performance of the GC, the QSPR and the hybrid GC-QSPR Gaussian process models is then compared with respect to the coefficient of determination (R^2) and the mean squared error (MSE). Table 4.7 shows the values for the length scale hyperparameters after the Gaussian process has been trained with the boiling point values for fatty acids. A ranking of the important features can be performed while a predictive model has been obtained including the provision of confidence bounds. The results (Table 4.6) show that for triglycerides, the Gaussian process regression with the molecular descriptors of the group contribution method combined with the QSPR descriptors perform better than all other subset combinations. The results for fatty acids show that the QSPR method performs best followed by the GC method with 1st and 2nd order groups. Some valuable results are also obtained from the length scale hyperparameter to perform a ranking of the descriptors as shown in Table 4.7. Other molecular descriptors could be added to the descriptor set to see if the predictive performance can be improved. If this is the case, QSPR can be used to further enhance the predictive performance. Other improvements could also be achieved with the optimisation procedure of the hyperparameters via nested sampling for example as discussed by Obrezanova et al. [16]. But the case study presented here has to be assessed critically since the data set of experimental values is small and Gaussian processes are known to be sensitive to small data sets and thus over-fitting the regression problem has to be avoided [36]. Therefore a methodology has to be developed subject to the experimental data size and number of features. Feature selection is a valuable tool and has been demonstrated with Gaussian process regression and should be tested with a larger data set. For small data sets, lasso regression could be applied [37].

Table 4.5: List of molecular descriptors.

Descriptor	Abbreviation	Unit
Occurence of CH3 groups	M_{CH3}	-
Occurence of COOH groups	M_{COO}	-
Occurence of HC=CH groups	$M_{HC=CH}$	-
Occurence of CH2 groups	M_{CH2}	-
Number of C atoms	N_C	-
Cubic root of gravitational index	$\sqrt[3]{GI}$	-
Partial negative surface area	PNSA	$Å^2$
Partial positive surface area	PPSA	$Å^2$
Difference in charged partial surface area	DPSA	$Å^2$
Fractional charged partial negative surface area	FNSA	$Å^2$
Fractional charged partial positive surface area	FPSA	$Å^2$
Surface weighted charged partial negative surface area	WNSA	$Å^2$
Surface weighted charged partial positive surface area	WPSA	$Å^2$
Total hydrophobic surface area	TASA	$Å^2$
Total polar surface area	TPSA	$Å^2$

Table 4.6: Comparison of Gaussian process prediction performance for the normal boiling point of fatty acids (26 data points) and triglycerides (64 data points) between different molecular descriptor sets.

Scoring	GC (1st)	GC (1st & 2nd)	QSPR	GC (1st) + QSPR	GC (1st & 2nd) + QSPR
Fatty acids (13 training points)	4 features	7 features	11 features	15 features	18 features
R^2	0.9704	0.9778	0.9835	0.9035	0.9580
MSE	265.41	138.24	124.37	601.23	410.20
	5 features	7 features	11 features	16 features	18 features
Triglycerides (32 training points)					
R^2	0.3949	0.2439	0.1586	0.2360	0.4934
MSE	126.69	189.14	189.14	189.14	101.51

Table 4.7: Length scale hyperparameters for individual features ordered from low to high for Gaussian Process models (GC, QSPR and GC+QSPR) predicting the normal boiling point of fatty acids.

GC (1st & 2nd)		QSPR		GC (1st & 2nd) + QSPR	
Descriptor i (Feature)	l_i	Descriptor i (Feature)	l_i	Descriptor i (Feature)	l_i
M_{CH_3}, M_{COOH}	0.001	FNSA	1.321	M_{CH_3}, M_{COOH}	0.001
$M_{CH_n=CH_m-CH_p=CH_k}$	0.001	$N_{C-Atoms}$	16.092	$N_{CH_n=CH_m-CH_p=CH_k}$	0.001
$M_{CH2-CH_m=CH_n}$	5.42	TPSA	96.52	$N_{CH_m=CH_n-COOH}$	0.21995
M_{CH_2}	7.66	PNSA	463.52	$N_{C-Atoms}$	25.34
$M_{CH_m=CH_n-COOH}$	71447	$\sqrt[3]{\text{GI}}$	94865	$M_{CH=CH}$	25.91
$M_{CH=CH}$	1.35×10^5	WNSA	2.2497×10^5	M_{CH_2}	28.45
		FPSA	1.4074×10^6	WPSA	675.72
		DPSA	1.6802×10^7	DPSA	843.4
		PPSA	1.8794×10^8	PNSA	3018.2
		WPSA	2.9307×10^8	FPSA	13329
		TASA	7.5709×10^{11}	CRGI	64017
				TASA	4.00×10^5
				TPSA	1.82×10^7
				WNSA	9.51×10^7
				PPSA	1.01×10^8
				FNSA	4.27×10^{12}
$\sigma_f = 65.019$, $\sigma_n = 1.3256$		$\sigma_f = 124.24$, $\sigma_n = 7.2252$		$\sigma_f = 75.719$, $\sigma_n = 0.24137$	

4.4 Server based property prediction tool: SAFEPROPS

A new software tool has been developed as part of this work and is named SAFEPROPS. SAFEPROPS provides accurate, reliable and fast predictions using group contribution (GC) methods such as Marrero-Gani, modified UNIFAC (Lyngby, Dortmund) and predictive Soave Redlich Kwong (PSRK). Gaussian Process regression has been implemented to provide a fully data-driven approach where the input features can be a set of molecular descriptors relevant for the interested property and where the output vector holds the experimental values for each row of the input matrix. The software prototype is implemented using Python as the main programming language and a schematic diagram of the framework of the software is shown in Figure 4.2.

The necessary group-contribution values together with the covariance matrix are obtained from the relational database (PostgreSQL) which has been populated using the parameter and error estimation routines described before. Currently the data is stored in the JSONB format in non-relational form which can be improved to make the data access faster since not the whole set of group contribution values and the entire covariance matrix has to be retrieved. If the data itself would be stored in relational form then it would

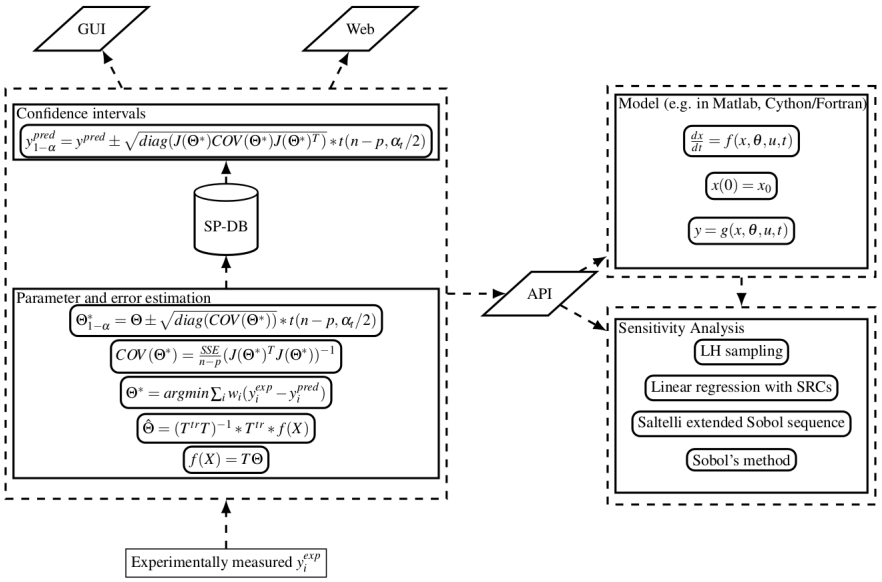


Figure 4.2: Framework of the SAFEPROPS software for applying group contribution methods.

be able to only access the necessary values for the groups the molecule has been dissected into. The entity relational diagram is shown in Figure 4.3 and depicts how one GC method relates to many properties by applying a SQL PRIMARY KEY Constraint. A trained Gaussian process will be serialised for persisting the model on the database. This database structure makes handling new data more convenient for adding other properties or other thermophysical and thermodynamic methods.

The graphical user-interface of SAFEPROPS is a web-interface (Figure 4.4) with a search bar for providing a SMILES string. The submission of the SMILES will let the SMARTS_matching_SMILES algorithm dissect the molecule into the groups defined by the GC method. The set of SMARTS definitions is stored in the gc_data table as a JSONB file (Figure 4.3). Each GC method has its own set of SMARTS.

An application programming interface (API) allows the user to directly connect to the SAFEPROPS software with their Matlab, Python or other programming language scripts (Table 4.8). The code snippets show how a connection to the server can be established with Python or Matlab (Figure 4.5). This allows the user to declare and retrieve the estimates and uncertainty

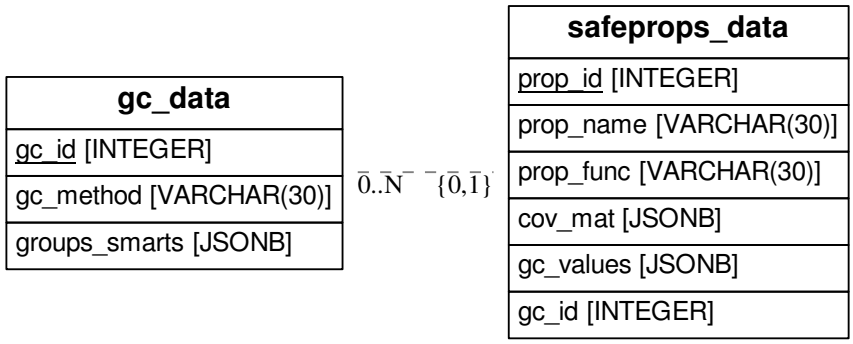


Figure 4.3: Entity relation diagram describing how GC methods are stored in the PostgreSQL database.

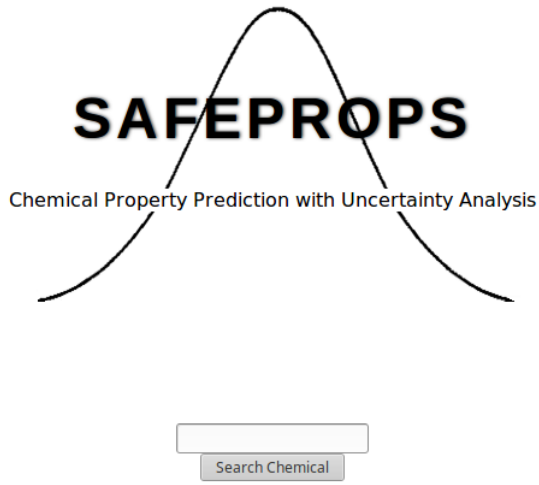


Figure 4.4: Web-interface of the SAFEPROPS property prediction software prototype.

bounds directly in their developed routines.

Table 4.8: API documentation of SAFEPROPS.

Function	Field	Type	Description
single_prediction			Fetch single value prediction for defined SMILES and property
	SMILES	string	SMILES string
	prop	string	Property string
batch_prediction			Fetch multiple value predictions for defined list of SMILES and list of properties
	SMILES_list	list of strings	List of SMILES strings
	prop_list	list of strings	List of property strings
single_group_occurences			Fetch the occurences of the groups for defined SMILES
	SMILES	string	SMILES string

The software can be installed locally on the users own machine and accessed via localhost; or a server can be rented to run SAFEPROPS on e.g. IBM cloud. The docker container allows the easy installation of the software on a server. The future aim to open-source the software and provide the tool as a Git repository gives the PSE community open and free access to SAFEPROPS and issues can be submitted to collectively add new methods and resolve mistakes in the code.

```python
#Python code snippet for property data retrieval from SAFEPROPS
import requests
import json
input = {"SMILE": 'CC', "prop": 'LowerFlammabilityLimit'}
response = requests.get(
"http://localhost:8000/single_prediction", input)
data = response.json()
ypred_value = data['ypred']
ybound_value = data['ybound']
print(ypred_value)
print(ybound_value)
```

.

```matlab
%Matlab code snippet for property data retrieval from SAFEPROPS
url = 'http://localhost:8000/single_prediction';
SMILE_string = 'CCCCCC';
prop_string = 'LowerFlammabilityLimit';
S = webread(url, 'SMILE',SMILE_string, 'prop', prop_string);
disp(S(1));
```

.

Figure 4.5: Examples in Python or Matlab how to connect to SAFEPROPS and retrieve property predictions.

Bibliography

[1] K. Chowdhury, L. A. Banu, Khan, and A. Latif, "Studies on the fatty acid composition of edible oil," *Bangladesh Journal of Scientific and Industrial Research*, vol. 42, no. 3, pp. 311–316, 2007. DOI: 10.3329/bjsir.v42i3.669.

[2] R. O'Brien, *Fats and Oils*. CRC Press, 2004. DOI: 10.1201/9780203483664.

[3] L. Zong, S. Ramanathan, and C. Chen, "Fragment-based approach for estimating thermophysical properties of fats and vegetable oils for modeling biodiesel production processes," *Industrial & Engineering Chemistry Research*, vol. 49, no. 6, pp. 3022–3023, 2010. DOI: 10.1021/ie100160v.

[4] A. Chang and Y. A. Liu, "Integrated process modeling and product design of biodiesel manufacturing," *Industrial & Engineering Chemistry Research*, vol. 49, no. 3, pp. 1197–1213, 2010. DOI: 10.1021/ie9010047.

[5] Y.-C. Su, Y. A. Liu, C. A. Diaz Tovar, and R. Gani, "Selection of prediction methods for thermophysical properties for process modeling and product design of biodiesel manufacturing," *Industrial & Engineering Chemistry Research*, vol. 50, no. 11, pp. 6809–6836, 2011. DOI: 10.1021/ie102441u.

[6] T. Wallek, J. Rarey, J. O. Metzger, and J. Gmehling, "Estimation of pure-component properties of biodiesel-related components: Fatty acid methyl esters, fatty acids, and triglycerides," *Industrial & Engineering Chemistry Research*, vol. 52, no. 47, pp. 16 966–16 978, 2013. DOI: 10.1021/ie402591g.

[7] Y. Nannoolal, J. Rarey, D. Ramjugernath, and W. Cordes, "Estimation of pure component properties: Part 1. estimation of the normal boiling point of non-electrolyte organic compounds via group contributions and group interactions," *Fluid Phase Equilibria*, vol. 226, pp. 45–63, 2004. DOI: 10.1016/j.fluid.2004.09.001.

[8] D. S. Damaceno, O. A. Perederic, R. Ceriani, G. M. Kontogeorgis, and R. Gani, "Improvement of predictive tools for vapor-liquid equilibrium based on group contribution methods applied to lipid technology," *Fluid Phase Equilibria*, vol. 470, pp. 249–258, 2018, SI:John P O'Connell. DOI: 10.1016/j.fluid.2017.12.009.

[9] J. Marrero and R. Gani, "Group-contribution based estimation of pure component properties," *Fluid Phase Equilibria*, vol. 183-184, pp. 183–208, 2001. DOI: 10.1016/S0378-3812(01)00431-9.

[10] C. A. Diaz-Tovar, R. Gani, and B. Sarup, "Computer-aided modeling of lipid processing technology," English, PhD thesis, Jul. 2011.

[11] A. S. Hukkerikar, "Development of pure component property models for chemical product-process design and analysis," English, PhD thesis, 2013.

[12] L. Constantinou and R. Gani, "New group contribution method for estimating properties of pure compounds," *AIChE Journal*, vol. 40, no. 10, pp. 1697–1710, 1994. DOI: 10.1002/aic.690401011.

[13] J. Frutiger, C. Marcarie, J. Abildskov, and G. Sin, "Group-contribution based property estimation and uncertainty analysis for flammability-related properties," eng, *Journal of Hazardous Materials*, vol. 318, pp. 783–793, 2016. DOI: 10.1016/j.jhazmat.2016.06.018.

[14] J. Frutiger, C. Marcarie, J. Abildskov, and G. Sin, "A comprehensive methodology for development, parameter estimation, and uncertainty analysis of group contribution based property models—an application to the heat of combustion," *Journal of Chemical & Engineering Data*, vol. 61, no. 1, pp. 602–613, 2016. DOI: 10.1021/acs.jced.5b00750.

[15] J. Frutiger, "Property uncertainty analysis and methods for optimal working fluids of thermodynamic cycles," English, PhD thesis, 2017.

[16] O. Obrezanova, G. Csányi, J. M. R. Gola, and M. D. Segall, "Gaussian processes: a method for automatic qsar modeling of adme properties," *Journal of Chemical Information and Modeling*, vol. 47, no. 5, pp. 1847–1857, 2007. DOI: 10.1021/ci7000633.

[17] M. E. Mondejar, S. Cignitti, J. Abildskov, J. M. Woodley, and F. Haglind, "Prediction of properties of new halogenated olefins using two group contribution approaches," *Fluid Phase Equilibria*, vol. 433, pp. 79–96, 2017. DOI: 10.1016/j.fluid.2016.10.020.

[18] M. E. Mondejar, J. Frutiger, S. Cignitti, J. Abildskov, G. Sin, J. M. Woodley, and F. Haglind, "Uncertainty in the prediction of the thermophysical behavior of new halogenated working fluids," *Fluid Phase Equilibria*, vol. 485, pp. 220–233, 2019. DOI: 10.1016/j.fluid.2018.12.020.

[19] D. Sola, A. Ferri, M. Banchero, L. Manna, and S. Sicardi, "Qspr prediction of n-boiling point and critical properties of organic compounds and comparison with a group-contribution method," *Fluid Phase Equilibria*, vol. 263, no. 1, pp. 33–42, 2008. DOI: 10.1016/j.fluid.2007.09.022.

[20] S. Boyd and L. Vandenberghe, *Introduction to Applied Linear Algebra*. Cambridge University Press, 2018.

[21] A. R. Katritzky, U. Maran, V. S. Lobanov, and M. Karelson, "Structurally diverse quantitative structure−property relationship correlations of technologically relevant physical properties," *Journal of Chemical Information and Computer Sciences*, vol. 40, no. 1, pp. 1–18, 2000. DOI: 10.1021/ci9903206.

[22] L. T. Lam, Y. Sun, N. Davey, R. Adams, M. Prapopoulou, M. B. Brown, and G. P. Moss, "The application of feature selection to the development of gaussian process models for percutaneous absorption," *Journal of Pharmacy and Pharmacology*, vol. 62, no. 6, pp. 738–749, 2010. DOI: 10.1211/jpp.62.06.0010.

[23] M. Banchero and L. Manna, "Comparison between multi-linear- and radial-basis-function-neural-network-based qspr models for the prediction of the critical temperature, critical pressure and acentric factor of organic compounds," *Molecules*, vol. 23, no. 6, 2018. DOI: 10.3390/molecules23061379.

[24] M. J. Sasena, "Flexibility and efficiency enhancements for constrained global design optimization with kriging approximations," English, PhD thesis, 2002.

[25] J. A. Caballero and I. E. Grossmann, "An algorithm for the use of surrogate models in modular flowsheet optimization," *AIChE Journal*, vol. 54, no. 10, pp. 2633–2650, 2008. DOI: 10.1002/aic.11579.

[26] S. Olofsson, M. Mehrian, L. Geris, R. Calandra, M. P. Deisenroth, and R. Misener, "Bayesian multi-objective optimisation of neotissue growth in a perfusion bioreactor set-up," in *27th European Symposium on Computer Aided Process Engineering*, ser. Computer Aided Chemical Engineering,

vol. 40, 2017, pp. 2155–2160. DOI: https://doi.org/10.1016/B978-0-444-63965-3.50361-5.

[27] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press, 2005.

[28] C. A. Micchelli, Y. Xu, and H. Zhang, "Universal kernels," *J. Mach. Learn. Res.*, vol. 7, pp. 2651–2667, December 2006.

[29] S. Olofsson, L. Hebing, S. Niedenführ, M. P. Deisenroth, and R. Misener, "Gpdoemd: A python package for design of experiments for model discrimination," *Computers & Chemical Engineering*, vol. 125, pp. 54–70, 2019. DOI: 10.1016/j.compchemeng.2019.03.010.

[30] R. M. Neal, *Bayesian Learning for Neural Networks*, eng. Springer New York, 1996. DOI: 10.1007/978-1-4612-0745-0.

[31] D. J. C. Mackay, "Bayesian interpolation," eng, *Neural Computation*, vol. 4, no. 3, pp. 415–447, 1992. DOI: 10.1162/neco.1992.4.3.415.

[32] D. R. Jones, "A taxonomy of global optimization methods based on response surfaces," *Journal of Global Optimization*, vol. 21, pp. 345–383, 2001. DOI: 10.1023/A:1012771025575.

[33] V. Nguyen, S. Gupta, S. Rana, C. Li, and S. Venkatesh, "Regret for expected improvement over the best-observed value and stopping condition," in *Proceedings of the Ninth Asian Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 77, PMLR, 2017, pp. 279–294.

[34] C. Qin, D. Klabjan, and D. Russo, "Improving the expected improvement algorithm," in *Advances in Neural Information Processing Systems 30*, Curran Associates, Inc., 2017, pp. 5381–5391.

[35] H. Moriwaki, Y.-S. Tian, N. Kawashita, and T. Takagi, "Mordred: A molecular descriptor calculator," *Journal of Cheminformatics*, vol. 10, no. 1, p. 4, February 2018. DOI: 10.1186/s13321-018-0258-y.

[36] G. C. Cawley and N. L. C. Talbot, "On over-fitting in model selection and subsequent selection bias in performance evaluation," *Journal of Machine Learning Research*, vol. 11, 2079—2107, 2010.

[37] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 58, no. 1, pp. 267–288, 1996.

CHAPTER 5

Process Design

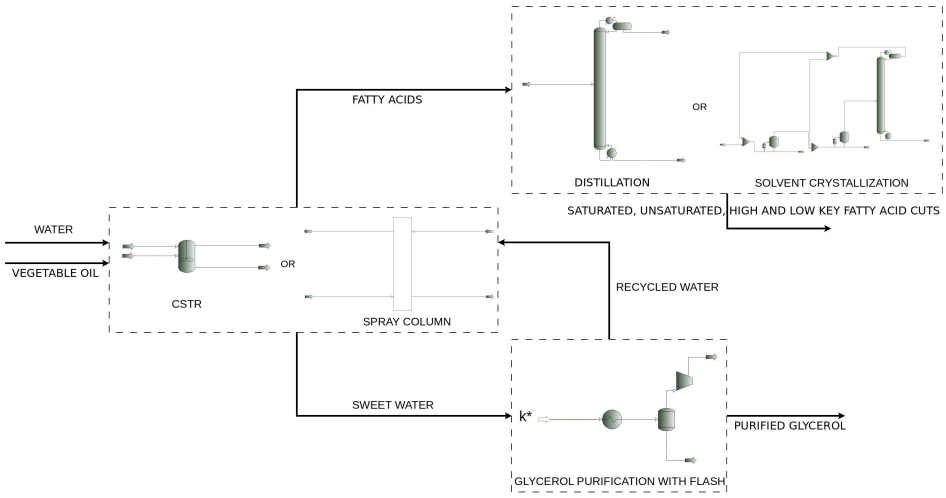


Figure 5.1: Overview of available oleochemical processes in the model library.

The important process tasks identified returning a high-revenue are the hydrolysis of vegetable oil with a counter-current spray column, molecular distillation to recover low concentrated components such as micro-nutrients in triglyceride and fatty acid mixtures and the separation of fatty acids in saturated and unsaturated components. The following process technologies were modelled and included in the model library of this framework: (1) The hydrolysis of triglycerides with a counter-current spray column, (2) molecular distillation of micro-nutrients such as β-carotene and α-tocopherol and (3) the separation of fatty acids through solvent (extractive) crystallisation.

In (1) the spray column was modelled as a finite volume model in Fortran and embedded in the PRO/II process simulator. Additionally, the model was wrapped with Python to perform sensitivity analysis, parameter estimation,

surrogate modelling and multi-criteria optimisation via differential evolution. The parameter estimation procedure allows to fit the model to different experimental data sets and the multi-criteria optimisation algorithm optimises the high pressure steam flow and the distribution over the inlets of the column.

In (2) molecular distillation is modelled and compared to a reference base case from literature. Sensitivity analysis is applied to evaluate if the critical property estimates $T_{C,i}$, $P_{C,i}$ and ω_i for each component in the vegetable oil is accurate enough in respect to the purity of the product.

In (3) the solvent crystallisation process is modelled in Python to separate stearic (saturated) and oleic acid (unsaturated) from each other with acetone as the solvent. The model was validated with a base case and data set from Wale [1] and Singleton [2].

Each section covering the individual unit operations is divided into the following sub-sections:

1. Introduction
2. Model description
3. Analysis
4. Conclusion

5.1 Counter-current spray column

5.1.1 Introduction

The hydrolysis of triglycerides is the reaction to perform if fatty acids or glycerol are the wanted products. Figure 5.2 shows the general hydrolysis scheme of triglycerides where the fatty acid side-chains depicted with the letter R can vary in length and saturation (amount of double bonds). Industrial

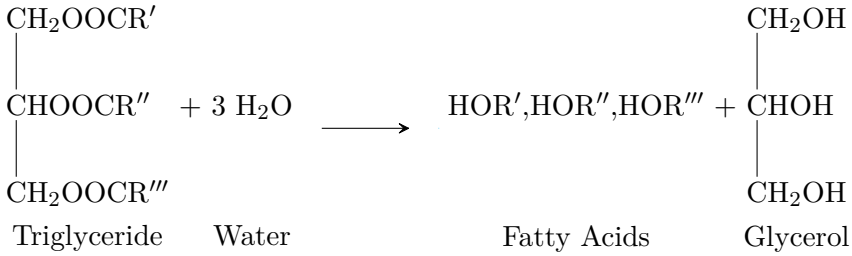


Figure 5.2: Hydrolysis reaction of triglycerides with water to give fatty acids and glycerol.

practice is to perform this reaction in a counter-current spray column where the vegetable oil is fed at the bottom of the column and the high pressurised water steam is injected at the top of the column (Figure 5.4). The continuous oil phase wanders upwards while the dispersed water droplets move to the bottom of the column. Besides acting as a reactant, the non-reacted water also dissolves the glycerol product and leaves the bottom of the column as so called sweet water. Consequently, mass transfer between the continuous and dispersed phase can be exhibited for water and glycerol (Figure 5.6) while the triglycerides and fatty acids remain in the oil phase and are insoluble in water. In this work the steam is assumed to be pure water but to make bio-refineries feasible the sweet water bottom product is subject to purification and therefore water with low concentrations in glycerol could be recycled back to the spray column. The hydrolysis reaction is discussed in this section and the research on counter-current spray columns is elaborated subsequently.

Patil et al. [3] investigated the hydrolysis reaction in a continuous-stirred tank reactor and propose a three-step reversible reaction scheme for the hydrolysis of tri- (TG), di- (DG) and monoglycerides (MG) with water (W) to give fatty acids (FA) and glycerol (G) where DG and MG act as intermediates:

$$\mathrm{TG} + \mathrm{W} \underset{-k_1}{\overset{k_1}{\rightleftharpoons}} DG \; + \; FA$$

$$\mathrm{DG} + \mathrm{W} \underset{-k_2}{\overset{k_2}{\rightleftharpoons}} MG \; + \; FA$$

$$\mathrm{MG} + \mathrm{W} \underset{-k_3}{\overset{k_3}{\rightleftharpoons}} G \; + \; FA$$

These three reactions can be aggregated into a single step reaction where the triglycerides react with water to fatty acids and glycerol:

$$\mathrm{TG} + 3\,\mathrm{W} \underset{-k}{\overset{k}{\rightleftharpoons}} 3\,FA \; + \; G$$

Forero-Hernandez et al. [4] show with the experimental data set from Alenezi et al. [5] that the identified mass transfer coefficient and reaction rate constants are highly correlated. The experimental data is based on the non-catalyzed hydrolysis of sunflower oil in a batch autoclave at 300 °C. A three reaction regime over time is assumed based on the work by Patil et al. [6] and Aniya et al. [7] where the first heterogeneous regime in the interface is mass transfer controlled, the second pseudo-homogeneous regime in the oil phase is controlled by the irreversible fast chemical reaction and the third homogeneous regime in the oil phase is reaching the reversible chemical equilibrium reaction controlled state.

The result of an extensive literature research led to the identification of one data set which could be used for validating the finite volume model in this work. This data set was found in the work of Jeffreys et al. [8]. The analytical model calculations by Jeffreys et al. rely on reaction rate data by Sturzenegger and Sturm [9] (catalyst level of 0.25% zinc oxide) and the value of 0.17 $\frac{1}{min}$ was used as the reaction rate constant. The reaction is assumed to be of pseudo first order and irreversible. It is assumed that the water content in the continuous phase is in excess and constant. Jeffreys et al. imply (referencing Mills and McClair [10]) that the increase of the continuous phase and the decrease with respectively 4% and 7% of the dispersed aqueous mass flow rate is negligible. As a consequence of this assumption the solubility of water in the oil phase will be about 10% at process conditions. The continuous and dispersed phase mass flows are being assumed as constant regardless of the internal column position and the dispersed phase droplets are assumed to travel through the column at the same velocity. In the discussion of their results,

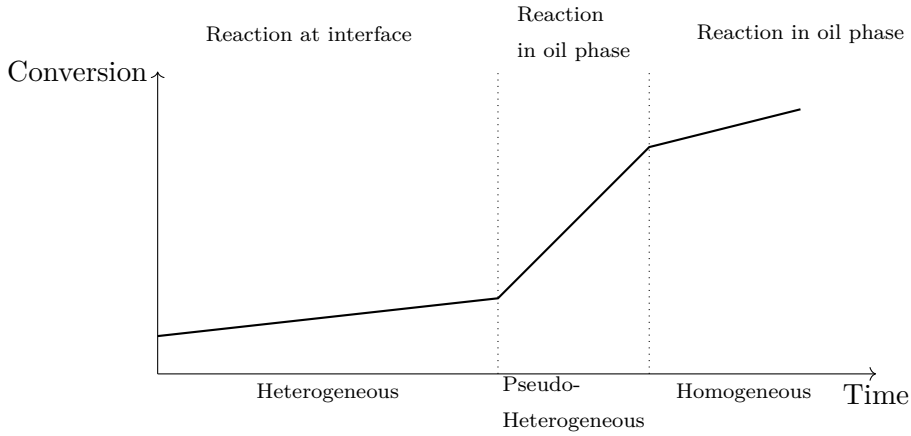


Figure 5.3: Hydrolysis of triglycerides in three step reaction periods.

Jeffreys at al. state that in the lower part of column the chemical reaction is the bottleneck to the mass transfer controlled process. Further, they mention the unfavourable operation at 18% flooding capacity which should rather be 30-40% as suggested by Minard and Johnson [11] to amend the mass tranfer process. Jeffreys et al. present an analytic algebraic equation for the glycerol fraction in the aqueous phase over the height of the column. The overall mass transfer coefficient Ka is obtained for six experimental data sets from a laboratory scale spray column. The work by Jeffreys et al. is also used as the reference to validate and discuss other developed models in the publications by Rifai et al. [12], Namdev et al. [13] and Attarakih et al. [14].

Rifai et al. [12] propose a modified version of the linear, steady-state spray column model established by Jeffreys et al. They present a non-linear model with the water solubility in the continuous phase being a function of composition and the variation of the internal flow rates. The hydrolysis reaction is assumed reversible and second order in nature. Rifai et al. expose with their model calculations that the assumption of irreversible pseudo first order kinetics made by Jeffrey et al. can't be justified. Table 5.1 shows the difference between the assumptions made by the two studies.

In this work we make the following assumptions based on the previously made findings from literature discussed before:

- The hydrolysis of triglycerides with water to fatty acids and glycerol follows a first order reaction to validate the model in this work with the

Table 5.1: Comparison between Jeffreys et al. and Rifai et al. models.

Aspect	Jeffreys et al.	Rifai et al.
Reaction kinetics	irreversible first order $r_{i,k} = (k_i Sh \rho_{Oil}/w_i) x_{TG_i,k}$	reversible second order: $r_{i,k} = (k_i Sh \rho_{Oil}^2/w_i)(x_{W,k} x_{TG_i,k} - \frac{1}{K} x_{GLY,k} x_{FA_i,k})$
Internal flowrates	assumed constant over column height	changes over column height
Water solubility	assumed constant over column height	changes over column height
Hydrodynamic model	-	Beyaert et al. [15] $v_s = \frac{G}{S(1-\epsilon)} + \frac{L}{S\epsilon}$
Solution formulation	Analytical	System of non-linear differential equations

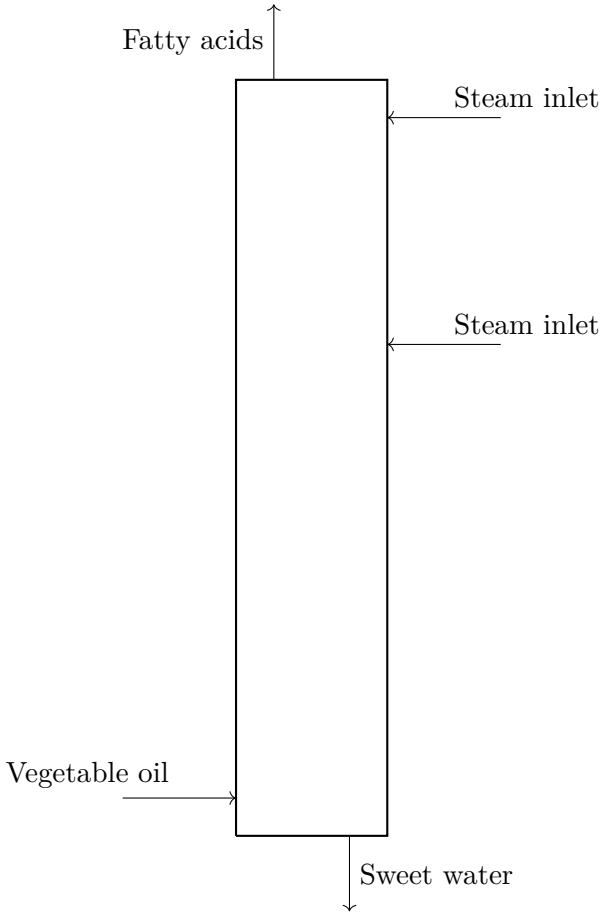


Figure 5.4: Counter-current spray column.

experimental data set from Jeffreys et al.

- Constant mass flow rates are assumed for the continuous and dispersed phases in case of validating the model by Jeffreys et al.

- Variable mass flow rate is assumed for the continuous and dispersed phase and the model is then re-parameterised in respect to the mass transfer rates, reaction rate and the backmixing coefficients.

- The finite volume model also takes back-mixing into account as van Egmond and Goossens [16] showed that they obtain better results when considering axial dispersion.

- The steam injected is assumed to be of pure water.

5.1.2 Model description

Figure 5.5 shows the interaction of the important properties and phenomena to take into account when modelling a spray column [17]. The important physical and temperature-dependent properties in the spray column model are liquid density and liquid viscosity. Liquid density is used in the reaction rate term to calculate the mass based reaction rate in a volumetric element. Liquid density and liquid viscosity enter also the hydrodynamic calculations (correlations or computational fluid dynamics) which are in need of these properties. The hydrodynamic calculations provide slip velocity and the interfacial area between the phases to the mass transfer rate calculations. Further, the backmixing and holdup values get directly included in the mass balance equations of the spray column model. The kinetics describe the stochiometry, reaction order and mechanism of the reaction system. These and the liquid density go into the reaction rate expression which makes up the production or consumption terms in the component mass balances. The process model for hydrolysing triglycerides with water to obtain fatty acids and glycerol is implemented as a finite volumes model. A diagram of the finite volume model can be seen in Figure 5.5 and the equations are described next.

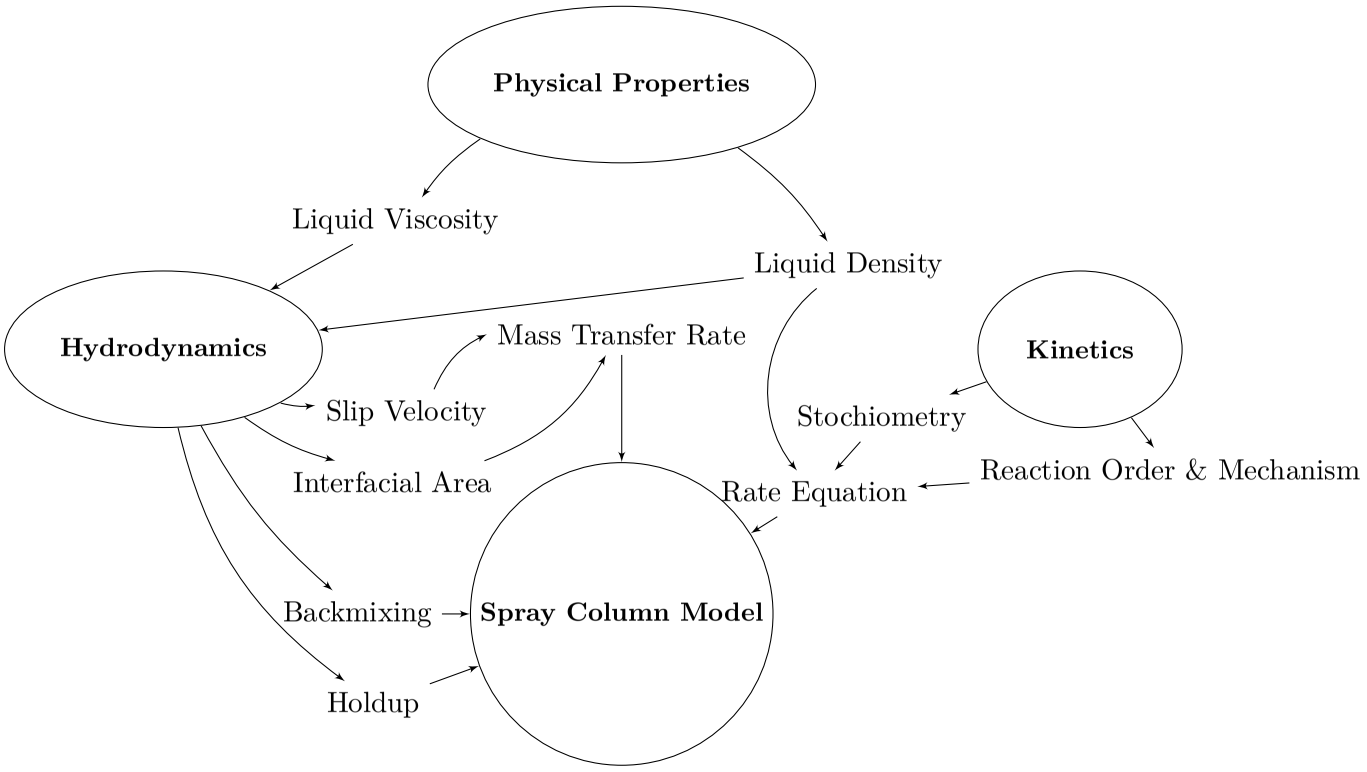


Figure 5.5: Properties and phenomena interaction in respect to a counter-current liquid-liquid spray column under subcritical conditions.

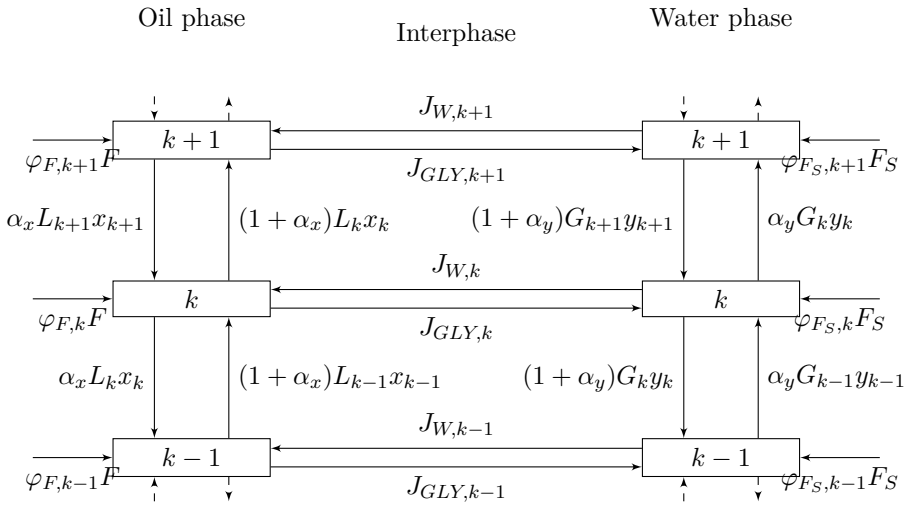


Figure 5.6: Schematic diagram of the finite volume model for the counter-current spray column.

The mass balance of triglycerides in the oil phase is:

$$
\begin{aligned}
m_k \frac{dx_{TG_i,k}}{dt} = 0 = {} & \overbrace{\alpha_x L_{k+1} x_{TG_i,k+1}}^{\text{Backmixing from upper stage}} - \overbrace{(1+\alpha_x)L_k x_{TG_i,k}}^{\text{Comp. flow to upper stage}} \\
& - \underbrace{\alpha_x L_k x_{TG_i,k}}_{\text{Backmixing to lower stage}} + \underbrace{(1+\alpha_x)L_{k-1}x_{TG_i,k-1}}_{\text{Comp. flow from lower stage}} \\
& - \underbrace{k_i Sh \rho_{Oil} x_{TG_i,k}}_{\text{Consumption of TG by 1st order reaction}} + \underbrace{\varphi_{F,k} F x_{F,TG_i}}_{\text{Feed on stage k}}
\end{aligned}
\tag{5.1}
$$

where L_k and G_k are the mass flowrates of oil and water in $\frac{lb}{h}$. h is the length of a stage or respectively a volumetric element since the modelled column doesn't feature any plates, stages or packings. The height of one element is $h = H/N$. $x_{TG_i,k}$ is the mass fraction of the triglyceride species TG_i in the volumetric element k. α_x is the backmixing coefficient for the continuous oil phase. k_i is the reaction coefficient in respect to the triglyceride species i, S is the cross-sectional area of the column, h is the height of a volumetric element, ρ_{Oil} is the density of the oil phase at the given operating temperature, $\phi_{F,k}$ is the fraction of the total feed flowrate F fed to the column at stage k.

The component balance of fatty acids in the oil phase is nearly identical to

the triglyceride balance except for the positive production term:

$$\begin{aligned} m_k \frac{dx_{FA_i,k}}{dt} = 0 = & \overbrace{\alpha_x L_{k+1} x_{FA_i,k+1}}^{\text{Backmixing from upper stage}} - \overbrace{(1+\alpha_x) L_k x_{FA_i,k}}^{\text{Comp. flow to upper stage}} \\ & - \underbrace{\alpha_x L_k x_{FA_i,k}}_{\text{Backmixing to lower stage}} + \underbrace{(1+\alpha_x) L_{k-1} x_{FA_i,k-1}}_{\text{Comp. flow from lower stage}} \\ & + \underbrace{\frac{k_i Sh \rho_{Oil} x_{TG_i,k}}{w_{FA}}}_{\text{Production of FA by 1st order reaction}} + \underbrace{\varphi_{F,k} F x_{F,FA_i}}_{\text{Feed on stage k}} \end{aligned} \tag{5.2}$$

$x_{FA,i,k}$ is the mass fraction of the fatty acid species in the volumetric element k. $w_{FA} = 1.0495$ is the mass related ratio to produce one unit fatty acid from one unit triglyceride. The component balance of glycerol in the oil phase includes the mass transfer of glycerol between the oil and aqueous phase:

$$\begin{aligned} m_k \frac{dx_{GLY,k}}{dt} = 0 = & \overbrace{\alpha_x L_{k+1} x_{GLY,k+1}}^{\text{Backmixing from upper stage}} - \overbrace{(1+\alpha_x) L_k x_{GLY,k}}^{\text{Comp. flow to upper stage}} \\ & - \underbrace{\alpha_x L_k x_{GLY,k}}_{\text{Backmixing to lower stage}} + \underbrace{(1+\alpha_x) L_{k-1} x_{GLY,k-1}}_{\text{Comp. flow from lower stage}} \\ & + \underbrace{\sum_{i=1}^{NoTG} \frac{Sh \rho_{Oil} x_{TG_i,k}}{w_{GLY}}}_{\text{Production of GLY by 1st order reaction}} \\ & - \underbrace{Ka_{GLY} Sh (x^*_{GLY,k} - x_{GLY,k})}_{\text{Mass transfer of GLY from oil to aqueous phase}} + \underbrace{\varphi_{F,k} F x_{F,GLY}}_{\text{Feed on stage k}} \end{aligned} \tag{5.3}$$

where $w_{GLY} = 11.7233$ is the mass related ratio to produce one unit glycerol from one unit triglyceride. For the component balance of glycerol in the aqueous phase the production term can be excluded since the reaction is only taking place in the oil phase:

$$\begin{aligned} m_k \frac{dy_{GLY,k}}{dt} = 0 = & \overbrace{\alpha_y G_{k-1} y_{GLY,k-1}}^{\text{Backmixing from lower stage}} - \overbrace{(\alpha_y+1) G_k y_{GLY,k}}^{\text{Comp. flow to lower stage}} \\ & - \underbrace{\alpha_y G_k y_{GLY,k}}_{\text{Backmixing from upper stage}} + \underbrace{(\alpha_y+1) G_{k+1} y_{GLY,k+1}}_{\text{Backmixing from upper stage}} \\ & + \underbrace{Ka_{GLY} Sh (y^*_{GLY,k} - y_{GLY,k})}_{\text{Mass transfer of GLY from oil to aqueous phase}} + \underbrace{\varphi_{F_S,k} F_S y_{F_S,GLY}}_{\text{Steam injection on stage k}} \end{aligned} \tag{5.4}$$

The internal flowrate for the dispersed (water) phase is defined as:

$$\begin{aligned}\frac{dG_k}{dt} = 0 = & \overbrace{\alpha_y G_{k-1}}^{\text{Backmixing from lower stage}} - \overbrace{(\alpha_y+1)G_k}^{\text{Total mass flow to lower stage}} \\ & + \underbrace{(\alpha_y+1)G_{k+1}}_{\text{Total mass flow from upper stage}} - \underbrace{\alpha_y G_k}_{\text{Backmixing from upper stage}} \\ & - \underbrace{Ka_W Sh((1-y_{GLY,k}) - y^*_{W,k})}_{\text{Mass transfer of water from aqueous to oil phase}} \\ & + \underbrace{Ka_{GLY} Sh(y^*_{GLY,k} - y_{GLY,k})}_{\text{Mass transfer of glycerol from oil to aqueous phase}} \\ & + \underbrace{\varphi_{F_S,k} F_S}_{\text{Steam injection on stage k}}\end{aligned} \tag{5.5}$$

and the internal flowrate of the continuous (oil) phase reads:

$$\begin{aligned}\frac{dL_k}{dt} = 0 = & \overbrace{\alpha_x L_{k+1}}^{\text{Backmixing from upper stage}} - \overbrace{(1+\alpha_x)L_k}^{\text{Total mass flow to upper stage}} \\ & - \underbrace{\alpha_x L_k}_{\text{Backmixing to lower stage}} + \underbrace{(1+\alpha_x)L_{k-1}}_{\text{Total flow from lower stage}} \\ & + \underbrace{Ka_W Sh((1-y_{GLY,k}) - y^*_{W,k})}_{\text{Mass transfer of water from aqueous to oil phase}} \\ & - \underbrace{Ka_{GLY} Sh(x_{GLY,k} - x^*_{GLY,k})}_{\text{Mass transfer of glycerol from oil to aqueous phase}} \\ & + \underbrace{\varphi_{F,k} F}_{\text{Feed on stage k}}\end{aligned} \tag{5.6}$$

The equilibrium concentration of glycerol in the aqueous phase at the interface $y^*_{GLY,k}$ can be expressed with the distribution ratio ψ_{GLY} and the concentration in the oil phase $x_{GLY,k}$:

$$y^*_{GLY,k} = \psi_{GLY} x_{GLY,k} \tag{5.7}$$

The distribution can be calculated for example with the modified UNIFAC model but in this work we use the data for the distribution ratio from the reference case for validation purposes.

The system comprises $(NoC + 2) * N$ equations with $(NoC + 2) * N$ unknown variables being the fractions of the individual triglycerides and fatty acids in the continuous phase, the 2 glycerol fractions in the continuous and dispersed phase and the internal flowrates of both phases in each volumetric element. In this model we assume that each triglyceride has equivalent fatty acids side-chains and consequently one kind of triglyceride will react to one kind of fatty acid. The continuous phase consists of triglycerides, fatty acids and glycerol. The dispersed phase is a mixture of glycerol and water since we assume no mass transfer of triglycerides and fatty acids between the oil and water interface. Thus, the water fraction in the dispersed phase can be derived from the glycerol fraction with the summation rule. Equations (5.1), (5.2) and (5.3) are $(NoC - 1) * N$ equations which gives $NoC * N$ equations when including equation set (5.4). The equation sets (5.5) and (5.6) add $2 * N$ equations and the system has no degrees of freedom with the number of equations being the same as the number of unknown variables. The system of nonlinear equations was solved with a global Newton method (NLEQ1 solver [18]).

5.1.3 Analysis

We compare the results of the proposed finite volume model in this work with the analytical model by Jeffreys et al. (experiment run no. 6) as seen in Figure 5.7. The parameters in Table 5.2 were used to validate the finite volume model against the analytical model by Jeffreys et al. The figure shows that the finite volume model aligns very well with the analytical model. The finite volume model presented in this paper was simulated with 100 volumetric elements. The glycerol content in the sweet water at the bottom of the column is 18.8%.

Global sensitivity analysis

Global sensitivity analysis allows to identify and rank the important parameters of an unit operation model and can also be used to locate sensitive zones in e.g. columns and reactors. In this work we perform variance-based Sobol sensitivity analysis (Appendix 9.3) to evaluate physical (liquid density), thermodynamic (distribution ratio) and phenomena (kinetics and mass transfer) based properties with respect to the sensitivity of the glycerol content in the sweet water stream at the bottom of the column.

Table 5.2: Parameters for the counter-current oil-splitting column in English units and mass based (experimental run number 6 by Jeffreys et al.).

Parameter	Symbol	Nominal Value	Unit
Overall mass transfer coefficient for glycerol	Ka	14.21	$[lb/(ft^3h)]$
Cross-sectional area of tower	S	3.688	$[ft^2]$
Mass flow of extract (aqueous phase)	G	3760	$[lb/h]$
Mass flow of raffinate (oil phase)	L	8540	$[lb/h]$
Glycerol distribution ratio	ϕ_{GLY}	10.32	[-]
Forward reaction rate coefficient	k	10.2	$[1/h]$
Height of column	H	73.5	$[ft]$
Glycerol content in fat	z_0/w_{GLY}	0.0853	[-]
Liquid density of fat	ρ	45.05	$[lb/ft^3]$
Backmixing coefficient of cont. phase (oil)	α_x	0.0	[-]
Backmixing coefficient of disp. phase (water)	α_y	0.0	[-]

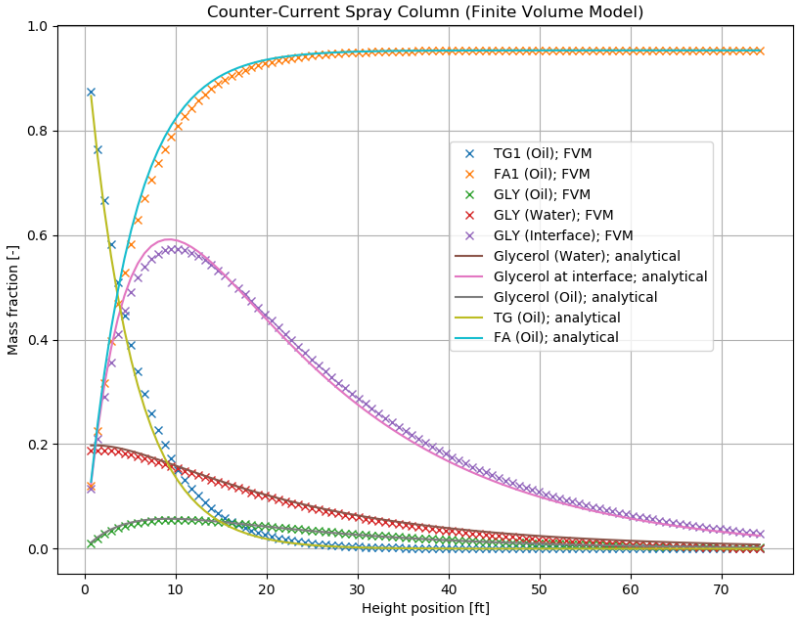


Figure 5.7: Validation of finite volume model (constant flows) with analytical model from Jeffreys et al..

Jeffreys et al. derive from their six experiments a variation in the overall mass transfer coefficient for glycerol from 10.1 to 16.0 $\frac{lb}{ft^2 h}$. These values can be calculated with the following equation:

$$Ka_{GLY} = \frac{G_{median}}{HTU * S} \tag{5.8}$$

where the HTU (height to transfer unit) values have been documented.

First we analyse the experimental data set where the mean value of Ka_{GLY} is 13.0 $\frac{lb}{ft^2 h}$ with a standard deviation of 3.0 ($\pm$23 %). The forward reaction rate is 10.2 $\frac{1}{h}$ with no further estimates or uncertainties given. They also report liquid density values for tripalmitin (C16:0) for each of the six experiments (Table 5.3) which we assume to be the density value at feed temperature on the first stage of the spray column. The mean of these six values is 45.016 with a standard deviation of 0.068 (0.15 %) $\frac{lb}{ft^3}$ and the mean of the distribution ratio is 10.26 with a standard deviation of 1.5 (14.6 %). For the sensitivity analysis we define the means of the parameters as the values from the experimental run number 6 and define a normal distribution of 10 % for each parameter.

Variance-based sensitivity analysis with the Sobol method was performed for the analytical model and the results are depicted in Figure 5.8. The results highlight that the distribution ratio and by this the liquid-liquid phase equilibrium has the highest effect on the glycerol fraction at the bottom product. The overall mass transfer coefficient follows as the second most important parameter and aligns with literature that the unit operation at hand is a mass-transfer driven process whereas sensitivity of the glycerol fraction to the reaction rate coefficient is negligible. The reason is the very slow reaction regime [13]. The liquid density has no effect on the conversion from the starting material (TG and W) to the products (FA and GLY) although the liquid density uncertainty has been set higher than actually analysed before.

Parameter estimation via differential evolution

The initial estimates of the parameters for fitting purposes can be obtained via differential evolution (DE). The sum of squared error is evaluated during the parameter space search via DE by formulating the following objective function

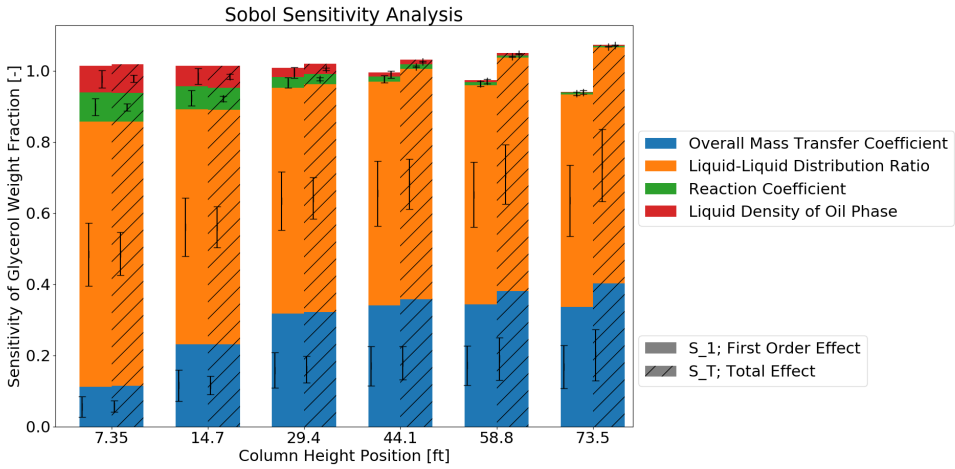


Figure 5.8: Sensitivity analysis of analytical model.

as the sum of squared differences between the predicted and measured data:

$$\sum_{i=1}^{N}[(y_{GLY,i}^{exp} - y_{GLY,i}^{sim})^2 + (x_{GLY,i}^{exp} - x_{GLY,i}^{sim})^2 + (\frac{G_{out,i}^{exp} - G_{out,i}^{sim}}{10000})^2 + (\frac{L_{out,i}^{exp} - L_{out,i}^{sim}}{10000})^2 + (\frac{G_{median,i}^{exp} - G_{median,i}^{sim}}{10000})^2 + (\frac{L_{median,i}^{exp} - L_{median,i}^{sim}}{10000})^2] \tag{5.9}$$

where

$$G_{median,i}^{exp} = \frac{G_{in,i}^{exp} + G_{out,i}^{exp}}{2}; \quad L_{median,i}^{exp} = \frac{L_{in,i}^{exp} + L_{out,i}^{exp}}{2} \tag{5.10}$$

and

$$G_{median,i}^{sim} = \frac{G_{in,i}^{exp} + G_{out,i}^{sim}}{2}; \quad L_{median,i}^{sim} = \frac{L_{in,i}^{exp} + L_{out,i}^{sim}}{2} \tag{5.11}$$

A scaling factor had to be introduced for the mass flow rate differences to scale them to the value range of the mass fractions.

DE is a stochastic direct search method by Storn and Price [19] and the algorithm is summarised in the following:

1. Specify population size, number of generations, crossover probability, mutation factor

2. Initialise vector population where parameters are uniformly distributed within their bounds

3. Evaluate the objective (cost) function for all individuals (vectors) and store in the fitness variable

4. Generation loop until number of generations or fitness of cost function is reached:

   4.1. Mutation (Parameter mixing): Select a target vector, choose randomly three other vectors and create mutant vector $m = v_1 + m_{factor} * (v_2 - v_3)$ where m_{factor} is called the mutant factor or differential weight

   4.2. Recombination: Generate trial vector by a probabilistic swapping (crossover) of elements from current target vector with mutant vector

   4.3. Replacement: Evaluate cost function and replace target vector with trial vector if the cost function is lower with the parameters from the trial vector

5. Parameter vector is returned with best fitness

The model is then fitted with the parameters returned from the DE routine as the first guess. Table 5.3 summarises the data set from Jeffreys et al. which was used for the parameter estimation.

We assumed the same conditions as Jeffreys et al. did in their work. Since Rifai et al. show that variable internal flowrates can't be assumed constant we defined variable continuous (oil) and dispersed (water) stream flowrates for the finite volume model. A parameter estimation has to be performed for Ka_{GLY} and the second mass transfer coefficient Ka_W describing the mass transfer of water between the continuous and dispersed phase. The forward reaction constant k has also been included as a parameter to be estimated. The results of the parameter estimation via DE are summarised in Table 5.4 and compared to the data from Jeffreys et al.

Jeffreys et al. calculate individual mass transfer coefficients of glycerol for each experimental run. In this work all six experiments are used for fitting

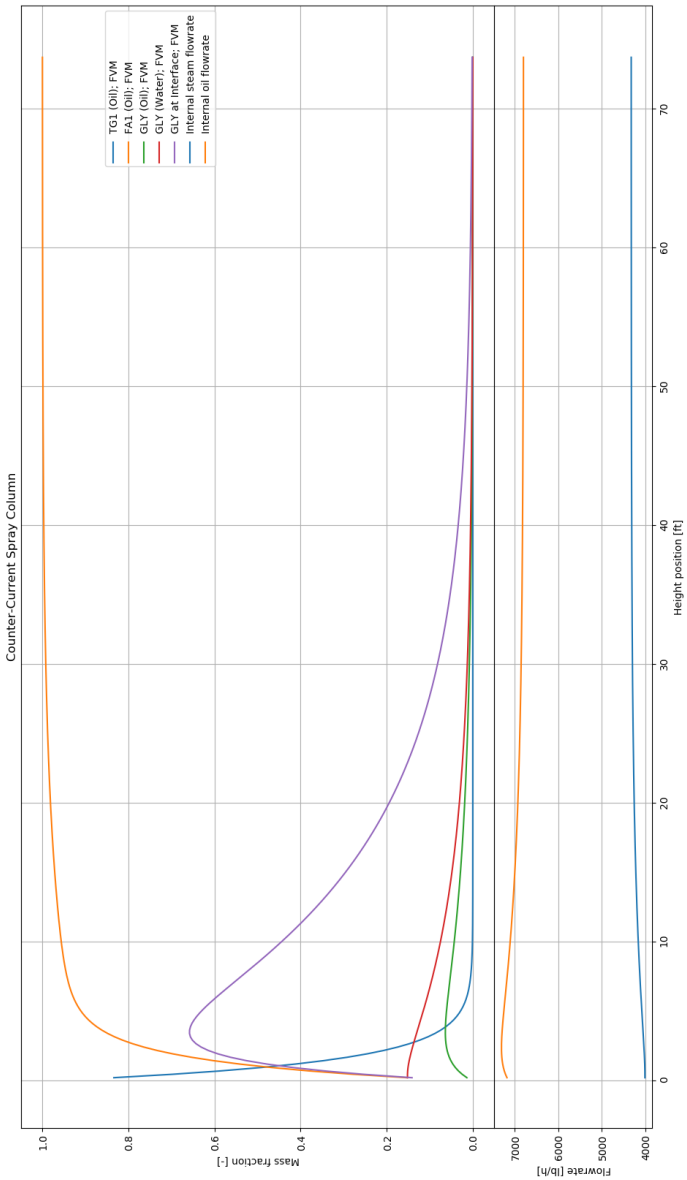


Figure 5.9: Mass fraction profiles after parameter estimation of Ka_W, Ka_{GLY}, k, α_x and α_y.

Table 5.3: Data used for parameter estimation.

Experimental Run	Input L_{in} [lb/h]	G_{in} [lb/h]	ρ_{Oil} [lb/ft^3]	m [-]		
#1	7260	4600	45	10.32		
#2	6490	4440	45.05	9.56		
#3	6905	4300	45	11.38		
#4	7400	3980	45.1	11.67		
#5	6570	4480	44.9	8.32		
#6	8175	4120	45.05	10.32		
Experimental Run	Output y_{GLY} [-]	x_{GLY} [-]	L_{out} [lb/h]	G_{out} [lb/h]	G_{median} [lb/h]	L_{median} [lb/h]
#1	0.1605	0.03	8050	3810	4205	7655
#2	0.1705	0.037	7180	3750	4095	6835
#3	0.189	0.027	7370	3835	4070	7140
#4	0.182	0.019	7770	3610	3795	7585
#5	0.227	0.027	7340	3710	4095	6955
#6	0.188	0.024	8900	3395	3760	8540

the parameters Ka_{GLY}, Ka_W, k, α_x and α_y. The covariance matrix obtained from the parameter estimation is:

$$\text{Cov} = \begin{array}{c} \begin{matrix} \text{Ka}_{GLY} & \text{k} & \alpha_x & \alpha_y & \text{Ka}_W \end{matrix} \\ \begin{pmatrix} 1.07 \times 10^4 & -1.83 \times 10^4 & 8.08 \times 10^2 & -1.71 \times 10^2 & 1.51 \times 10^4 \\ -1.83 \times 10^4 & 3.77 \times 10^4 & -9.59 \times 10^2 & 4.01 \times 10^1 & -2.58 \times 10^4 \\ 8.08 \times 10^2 & -9.59 \times 10^2 & 1.15 \times 10^2 & -3.59 \times 10^1 & 1.13 \times 10^3 \\ -1.71 \times 10^2 & 4.01 \times 10^1 & -3.59 \times 10^1 & 1.43 \times 10^1 & -2.37 \times 10^2 \\ 1.51 \times 10^4 & -2.58 \times 10^4 & 1.13 \times 10^3 & -2.37 \times 10^2 & 2.13 \times 10^4 \end{pmatrix} \end{array} \tag{5.12}$$

The standard deviation of the parameters' mean value is calculated from the covariance matrix and results in:

$$\sigma = \sqrt{diag(\text{Cov})} = \begin{pmatrix} 103.60 \\ 194.05 \\ 10.71 \\ 3.78 \\ 145.94 \end{pmatrix} \tag{5.13}$$

The results show that more experimental data is necessary to provide a satisfactory parameter estimation. Further, changing the kinetic model to a second order reaction may enhance the parameter estimation but if one wants to include a second order reaction model then experimental data of the water concentration in the oil phase is needed.

Table 5.4: Results of parameter estimation via differential evolution for finite volume model with variable internal water and oil flowrates.

Parameter	Re-parameterized model		Jeffreys et al.			
Ka_{GLY}	19.06		14.21			
Ka_W	28.95		-			
k	33.68		10.2			
α_x	0.01		0.0			
α_y	0.10		0.0			
Experiment	y_{GLY}^{sim}	G_{out}^{sim}	y_{GLY}^{exp}	Deviation sim. [%]	G_{out}^{exp}	Deviation sim. [%]
1	0.1305	4026	0.1605	- 18.7	3810	+ 5.7
2	0.1427	3841	0.1705	- 16.3	3750	+ 2.4
3	0.1773	3602	0.189	- 6.2	3835	- 6.1
4	0.1676	3364	0.182	- 7.9	3610	- 6.8
5	0.1758	3758	0.227	- 22.6	3710	+ 1.3
6	0.1462	3553	0.188	- 22.2	3395	+ 4.7
	x_{GLY}^{sim}	L_{out}^{sim}	x_{GLY}^{exp}		L_{out}^{exp}	
1	0.0197	7056	0.03	- 34.3	8050	- 12.3
2	0.0217	6321	0.037	- 41.4	7180	- 12.0
3	0.0188	6706	0.027	- 30.4	7370	- 9.0
4	0.0171	7174	0.019	- 10.0	7770	- 7.7
5	0.0360	6494	0.027	+ 33.3	7340	- 11.5
6	0.0223	7967	0.024	- 7.1	8900	- 10.5
	G_{median}^{sim}	L_{median}^{sim}	G_{median}^{exp}		L_{median}^{exp}	
1	7447	4169	4205	+ 77.1	7655	- 45.5
2	6727	3992	4095	+ 64.3	6835	- 41.6
3	7170	3788	4070	+ 76.2	7140	- 46.9
4	7587	3524	3795	+ 99.9	7585	- 53.5
5	6999	3923	4095	+ 70.9	6955	- 4.2
6	8364	3683	3760	+ 22.4	8540	- 2.0

Multi-criteria optimisation via differential evolution

Energy efficiency is an important aspect to make the economic performance of the spray column more viable and the direct injected steam consumes the largest energy share in this process [20]. The operating cost have to be evaluated in order to optimise the amount of steam fed to the column and how to distribute it between the two steam inlets. The steam cost is summarised in Table 3 [21] with the total cost for steam production being 13.6 $ per 1000 lb steam.

A possible formulation of the environmental objective would be the Eco-indicator 99 [22] which describes the effect of a product or process on the environment over its life cycle in terms of three damage categories: Human health, ecosystem quality and resources. The three damage categories are then

Table 5.5: Fixed and variable cost for high pressure (HP) steam production.

Cost	Unit	per 1000 lb steam
Average boiler fuel	MMBtu	1.56
Fresh water	$	0.02
Water treatment cost	$	0.74
Water preheating and pumping	$	0.62
Deareation steam	$	1.10
FD fan	$	0.05
C_{var} (variable cost)	$	11.9
Boiler capital	MM$	20
R depreciation factor	% of capital	15
Maintenance cost	% of capital	2
Two employees	$/a	120000
Employee cost factor	-	3
C_{fix} (fixed cost)	$	1.7
$C_{ST} = C_{var} + C_{fix}$	$	13.6

Fuel price: 6 $/MMBtu

weighted and normalised to balance or put emphasis on short or long term perspectives [23, 24]. The weighted values of the three damage categories are then summed up to give the Eco-indicator. The measure of the Eco-indicator is performed in points whereas 1 Point aligns with one thousandth of the yearly environmental load of one average European inhabitant. Table 5.6 lists the points per lb of the material/energy flows β_b consumed by the spray column process [25]. With this table an analysis for each damage category can be assessed as a function of the steel used for building the spray column, the steam consumed per year and the electricity needed for feeding the oil to the spray column. First the resource flows β_b are multiplied with the individual impact category values and then summed up for obtaining the impact of the resource usage on the damage category. Then the damage category values referenced to each resource are summed up and subsequently weighted and normalised to obtain the Eco 99 indicator with the final summation. It is noted that the values in Table 5.6 are already normalised in respect to the steel, steam and electricity consumption. The equation for the indicator can

be formulated as follows:

$$\text{Eco } 99 = \sum_b \sum_d \delta_d \omega_d \sum_{k \in K_d} \beta_b \alpha_{b,k} \tag{5.14}$$

where the individual damage factors in the specific impact category d can be summed up:

$$\alpha_{b,d} = \sum_{k \in K_d} \alpha_{b,k} \tag{5.15}$$

and thus the indicator simplifies to:

$$\text{Eco } 99 = \sum_b \sum_d \delta_d \omega_d \beta_b \alpha_{b,d} \tag{5.16}$$

Table 5.6: Impact categories for the eco-indicator and normalised data for steel, steam and electricity.

Impact category	Steel [points/lb]	Steam [points/lb]	Electricity [points/kWh]
Human health (d=1)			
Carcinogenics	2.867×10^{-3}	5.352×10^{-5}	4.360×10^{-4}
Climate change	5.942×10^{-3}	7.257×10^{-4}	3.610×10^{-6}
Ionizing radiation	2.046×10^{-4}	5.126×10^{-4}	8.240×10^{-4}
Ozone layer depletion	2.064×10^{-6}	9.525×10^{-7}	1.210×10^{-4}
Respiratory effects	3.633×10^{-2}	3.570×10^{-7}	1.350×10^{-6}
Ecosystem (d=2)			
Acidification	1.229×10^{-3}	5.488×10^{-3}	2.810×10^{-4}
Ecotoxicity	3.379×10^{-2}	1.270×10^{-3}	1.670×10^{-4}
Resources (d=3)			
Land occupation	1.692×10^{-3}	3.892×10^{-5}	4.680×10^{-4}
Fossil fuels	2.690×10^{-2}	5.670×10^{-2}	1.200×10^{-3}
Mineral extraction	3.366×10^{-2}	4.001×10^{-6}	5.7×10^{-6}

In conclusion, the description of the multi-criteria optimisation problem is:

$$
\begin{aligned}
\underset{\boldsymbol{x}=\{F_S,\phi_{F_S,1}\}}{\text{minimise}} \quad & f(\boldsymbol{x}) = 1/\text{Profit} \\
& = 1/(\text{Revenue} - \text{TAC} - \text{Raw Material Cost}) \\
\text{subject to} \quad & F_S \in [50, 5000] \\
& \phi_{F_S,Pos1} \in [0.1, 1.0] \\
& \phi_{F_S,Pos2} = 1 - \phi_{F_S,Pos1} \\
& \text{Eco}_{99} < 30000 \\
& \text{Product Purity Constraint } x_{FA} \geq 0.95
\end{aligned}
$$

As we applied the differential evolution algorithm to parameter estimation, we apply the same to the optimisation problem. For both, parameter estimation and multi-criteria optimisation, the parameters of the DE algorithm were set to a population size of 15, a mutation range of 0.5-1.0 with dithering enabled and a recombination value of 0.7. These paramater values are the standard setting of the differential_evolution function in scipy.optimize. The two parameters subject to variation are the steam flowrate F_S and the fraction of the total feed injected through the top inlet $\phi_{F_S,Pos1}$ with the position being at the 300th volume element and the second inlet $\phi_{F_S,Pos2}$ being positioned at the 200th element of the column which is modelled with 300 finite volumes. The algorithm evaluates the objective function until it converges against a minimum and the stopping criterion is reached.

Before discussing the results for the multi-criteria optimisation, the response surfaces of the finite volume model for the fatty acid and glycerol fractions at the top and bottom of the column are shown in Figure 5.10 and 5.11. The glycerol fraction increases with decreasing steam flowrates since the glycerol will be more concentrated with lower steam flow rates until it reaches a maximum and then decreases because no water is available for the reaction. We can see a slight increase of the glycerol fraction over the amount of water fed through the first inlet.

Simultaneously when increasing the water flowrate we can see in Figure 5.11 that the fatty acid fraction reaches a plateau at about 2000 $\frac{lb}{h}$. Meaning that the water flowrate of 4120 $\frac{lb}{h}$ in the base case design is too high and dilutes on the one side the glycerol content in the sweet water product and on the other side it doesn't increase the fatty acid content in the top product. The multi-criteria optimisation will therefore find the point where the glycerol and fatty acid fractions are balanced out to gain the highest product revenue with a minimum in operating cost and Eco99 indicator points.

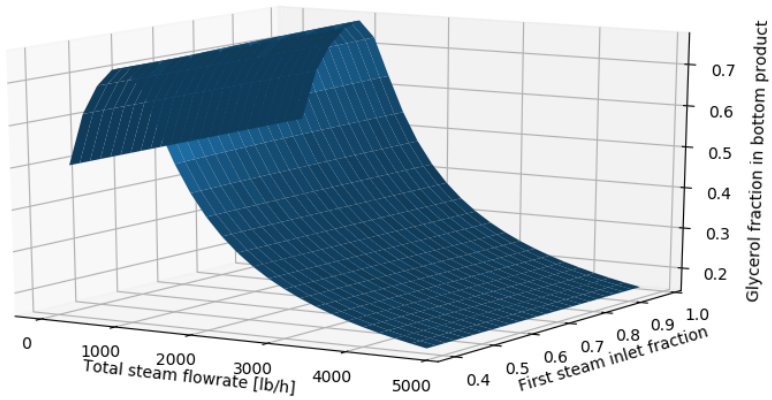


Figure 5.10: Response surface for glycerol fraction in bottom product with variable water flowrate and two inlets for steam.

In regards to the revenue which can be generated from the fatty acid and sweet water product streams, we assume that the product streams will be further purified and therefore set the prices for the palmitic acid product at the top of the spray column to be 0.71 $\frac{US\$}{lb}$ [26] which is the price for high grade palmitic acid. The sweet water product at the bottom of column is assumed to be further purified to high grade glycerol with a price of 0.085 $\frac{US\$}{lb}$ [27].The raw material price of the vegetable oil is 0.2359 $\frac{US\$}{lb}$ [28].

The size of the column is 73.5 ft in height [8], 2.16696 ft in diameter [14] and the column wall thickness is assumed 0.01001 ft (3.05 mm). The material is stainless steel 316 ($\rho_{SS316} = 229.9\ \frac{kg}{ft^3} = 506.84\ \frac{lb}{ft^3}$). For calculating the capital cost we assume the spray column being the shape of a cylinder and thus the weight of the column is 5.11 ft^3 times 506.84 $\frac{lb}{ft^3}$ which gives 116522.5 lb. The price of stainless steel 316 is 4227 $\frac{\text{US\$}}{t}$ [29] and thus we obtain a capital cost of 4966 US$ for the material of the spray column. The sustainability indicator calculation covers the used steel material, steam generation and the electricity for pumping. This results in the following equation for the Eco99 indicator:

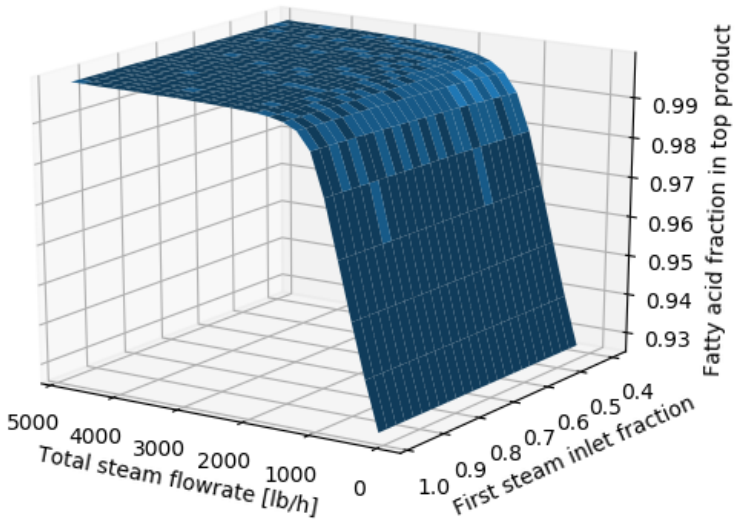


Figure 5.11: Response surface for fatty acid fraction in top product with variable water flowrate and two inlets for steam.

$$\begin{aligned} \text{Eco99} = \sum_{d} \omega_d (\beta_{Steel} \sum_{k \in K_1} \alpha_{Steel,k} \\ + \beta_{Steam} \sum_{k \in K_2} \alpha_{Steam,k} \\ + \beta_{Electricity} \sum_{k \in K_3} \alpha_{Electricity,k}) \end{aligned} \tag{5.17}$$

where β_{Steel} = 1174.8 kg and the pump duty for the feed is $\beta_{Electricity}$ = 440.59 kWh. The steam flowrate is a decision variable subject to change during the differential evolution algorithm. The weighting factors ω_d are set in respect to a hierarchist perspective (human health = 40 %, ecosystem quality = 40 % and resources = 20 %) to $\omega_1 = 0.4$, $\omega_2 = 0.4$ and $\omega_3 = 0.2$. In respect

to the product purity the constraint has been defined as:

$$\text{Product Purity Constraint} = 0.95 - x_{FA,300} \tag{5.18}$$

where $x_{FA,300}$ is the fatty acid content of the finite volume element at the top of the column for the continuous phase. The different criteria have to be scaled appropriately.

The results (Table 5.7) show that the optimisation minimises the steam flowrate to reduce the steam cost. The optimal operating point will achieve a fatty acid fraction in the top product of 0.99 while the glycerol fraction in sweet water will be 0.76 with a top product flowrate of 7904 lb/h and a sweet water flowrate of 767 lb/h.

Table 5.7: Results of multi-criteria optimisation.

Input and objective	Unit	Value	Input bounds
Input			
Steam flowrate	lb/h	1069	[50, 5000]
First steam inlet fraction	-	1.0	[0.1, 1.0]
Objective			
Revenue	\$/a	48211258	
Total annual cost (TAC)	\$/a	127290	
Raw material cost	\$/a	16775116	
Profit	\$/a	31308853	
Eco99 indicator	Points	28416	

5.1.4 Conclusion

The validation of the finite volume model showed that the model can be used to describe a spray column unit operation although more experimental data is needed to fully develop the finite volume model. The experimental setup of a counter-current spray column presented by Cadavid et al. [30] can be established to obtain the necessary data. Combined with the work by Forero-Hernandez et al. [4] to perform rigorous kinetic data analysis and the model presented here, important information about the hydrolysis of vegetable oils in spray columns can be obtained. Future research should be made in regards to computational fluid dynamics (CFD) to describe the hydrodynamics in the spray column. This will allow to generate surrogate functions from the

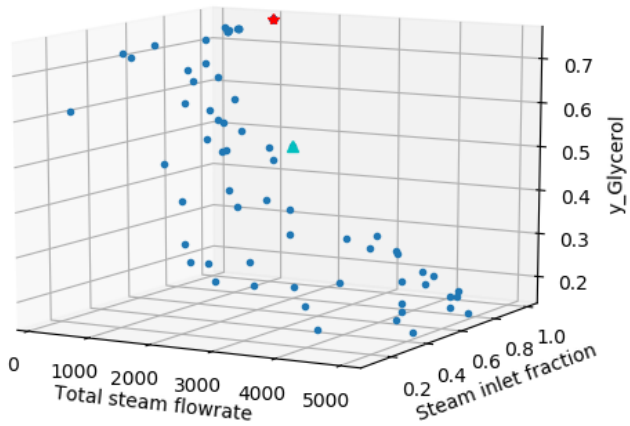


Figure 5.12: Iteration steps for finding the optimal point along the response surface of the glycerol fraction at bottom of the column in respect to steam flowrate and inlet fraction (the red point is the final solution of the differential evolution procedure; the triangle is the starting point).

computational cost-intensive CFD model and include them in the finite volume model.

The presented model allows to be adapted to different spray column setups and gives the engineer a valuable tool to validate, analyse and optimise an industrial scale spray column. The possibility to perform parameter estimation is given if experimental data from an existing plant is provided. Through multi-criteria optimisation sustainable process design can be achieved by including sustainability indicators such as the Eco99 indicator into the objective function. The model enables to test and analyse different scenarios and allows to communicate with packages and tools in line with the concept of digital industries of the future. Further, a proof of concept has been realised to also embed the spray column model in a commercial process simulator such as PRO/II (Figure 5.14).

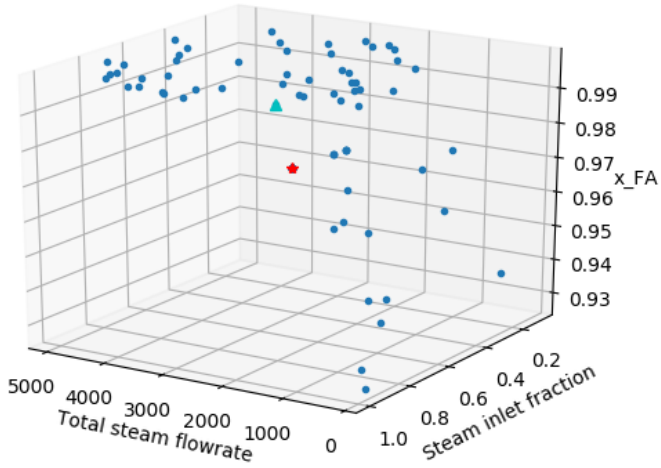


Figure 5.13: Iteration steps for finding the optimal point along the response surface of the fatty acid fraction in the product stream in respect to steam flowrate and inlet fraction (the red point is the final solution of the differential evolution procedure; the triangle is the starting point).

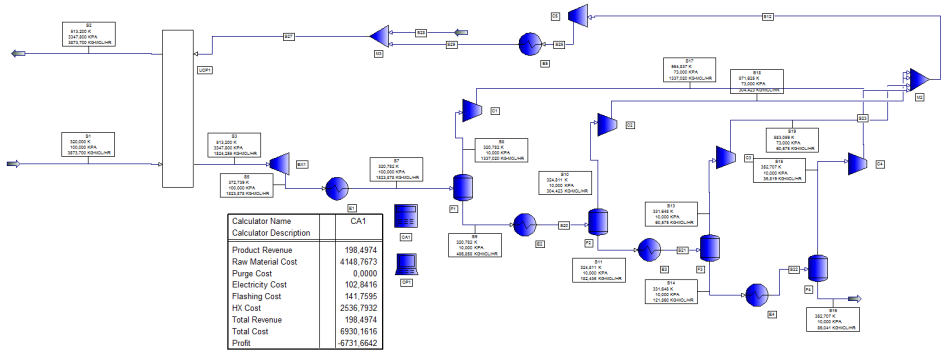


Figure 5.14: Embedded spray column in PRO/II with glycerol purification and recycling of water to the spray column.

5.2 Molecular distillation

5.2.1 Introduction

The name molecular distillation already indicates that the components are separated by their molecule size. The separation is performed at very low pressure which allows a low operating temperature and prevents the thermal decomposition of the components. The evaporated molecules which escape the heated surface of the evaporator don't collide with each other until they condensate. The mean free path of a molecule defines this behaviour and larger molecules have a shorter mean free path [31]. The unit operation is commonly applied in the food industries to recover high-value added or unwanted compounds such as β-carotene, tocopherols, tocotrienols and cholesterol from vegetable oils and fats [32–35]. The presented modelling and analysis of a short path evaporator based molecular distillation unit has the goal to study the effects of the uncertainties in physical property estimates on the product quality.

A methodology has been developed including tools integration to apply advanced uncertainty propagation and sensitivity analysis in connection with commercial process simulation software. The input parameters of the selected thermodynamic model, namely critical temperature, critical pressure and acentric factor, were considered as a source of uncertainty and analysed using Monte Carlo sampling techniques. This enabled the process model output uncertainty to be described as an empirical distribution function with a 95% confidence interval. Variance-based decomposition such as the Sobol method or standard regression were used to analyse the sensitivity of the respective properties. We also show that machine learning methods such as polynomial chaos expansion (PCE) can be applied to reduce the number of necessary process simulations and obtained equivalent results in comparison with the more costly full Monte Carlo based procedure.

5.2.2 Model description

A molecular distillation base case design has been modelled (Figure 5.15) and verified with the experimental data from Unnithan et al. [36] for the β-carotene recovery and the simulation performed by Tehlah et al. [37]. The molecular distillation process consists of a short path distiller (180 °C, 0.008 mbar) with an internal condensation system which operates at 20 °C, being an operating point between the condensing and the melting point of the fatty

acids. The oil is pre-heated to 120 °C before it enters the short path distillation system. The molecular distillation unit is regarded as a VLE flash and heat exchanger (integrated condenser) in the simulation and the top vapour product stream leaving the molecular distillation unit corresponds to the stream leaving the heat exchanger in the simulation.

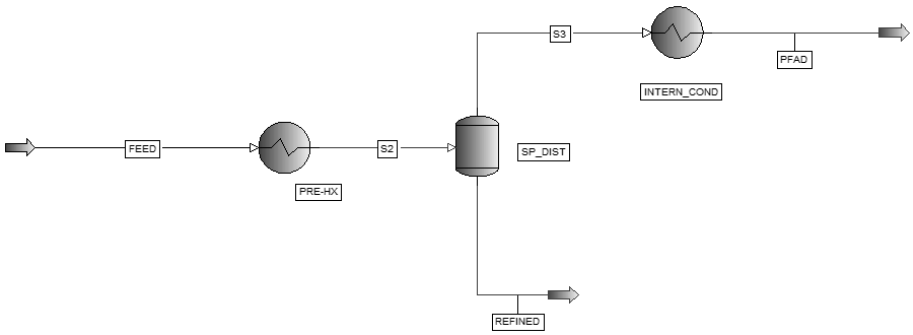


Figure 5.15: PRO/II flowsheet of molecular distillation process.

Tehlah et al. perform their simulation with the Aspen process simulator and apply the Redlich-Kwong-Aspen EOS which is based on the SRK EOS.

Soave-Redlich-Kwong equation of state

In this work the SRK EOS is applied:

$$P = \frac{RT}{V_m - b} - \frac{a}{V_m(V_m + b)} \tag{5.19}$$

The parameters a and b are calculated with mixing rules which can be found in literature in different forms [38, 39].

These rules depend on the component fractions in the mixture, some interaction parameters and the following correction factors describing the attraction and volume of the molecules:

$$a_i = \alpha_i 0.42747 \frac{R^2 T_{C,i}^2}{P_{C,i}} \tag{5.20}$$

$$b_i = 0.08664 \frac{RT_{C,i}}{P_{C,i}} \tag{5.21}$$

Mathias [38] introduced the generalised temperature-dependent function which improves the vapour pressure prediction:

$$\alpha_i(T) = [1 + m_i(1 - T_{r,i}^{1/2})]^2 \tag{5.22}$$

where m_i is a parameter for pure component i dependent on the acentric factor ω_i:

$$m_i = 0.48 + 1.574\omega_i - 0.176\omega_i^2 \tag{5.23}$$

and $T_{r,i}$ is the reduced temperature ($T_{r,i} = T/T_{C,i}$).

For comparison reasons we also applied the SimSci-SRK EOS, which similarly to the Aspen-SRK EOS, relies on a modification of the temperature-dependent function $\alpha_i(T)$ [40]. The K-value describing the distribution ratio of the individual components between the vapour and liquid phase will be derived from the EOS. Thus, the uncertainties in the property parameters will propagate from the EOS to the K-value calculations.

Table 5.8: Simulation results and comparison to experimental values from the patent by Unnithan et al. and simulation results from Tehlah et al. and this work with different equation of state models.

		Tehlah et al.	**This work**	
Bottom component recovery	**Experimental**	**RK-Aspen**	**SRK**	**SRK-SimSci**
β-carotene	95.98 %	98.96 %	95.82 %	95.81 %
α-tocopherol	98.54 %	not reported	40.37 %	40.30 %

Simulation Parameters (Feed flowrate, outlet temperature of pre-heater, operating temperature and pressure of flash): F = 2000 kg/h, T_1 = 120 °C, T_2 = 180 °C, P_2 = 0.008 mbar

The simulation results in Table 5.8 show the difference between the bottom component recoveries of β-carotene and α-tocopherol for the applied SRK models. Tehlah et al. did not report any values for the α-tocopherol recovery. We assume that their results for the top product recovery did not agree with the experimental results by Unnithan et al. [36]. Thus, the model has been validated in regards to the bottom product recovery and for the sensitivity analysis the product purity will be analysed as the output of the model. In the following, the effect of uncertainties on the bottom product of the molecular distillation unit is studied. The parameters ($T_{C,i}$, $P_{C,i}$ and ω_i) are subject to the sensitivity analysis in the next sections and are overwritten with the COM-interface and the values stored in the sampling hypercube for each simulation. The β-carotene product fraction (purity) values are stored for all simulations in the output vector.

5.2.3 Analysis

The strategy to apply the Monte Carlo method with process simulators is based on the methodology by Frutiger et al. [41] for property uncertainty propagation in process models. The methodology has been extended with Sobol (variance-based) sensitivity analysis (SA) in this work:

1) A process model is built in a commercial process simulation software such as PRO/II or Aspen. The property models and parameters for the uncertainty analysis need to be selected and the process variables are specified to satisfy the degrees of freedom.

2) The property uncertainty data is retrieved from databases (e.g. NIST TDE [42], AIChE DIPPR) or literature studies. Property uncertainty information needs to be estimated if not available, for example through calculation of the covariance matrix [43] or a bootstrap method [44].

3) Monte Carlo sampling technique is used to sample property values within its corresponding uncertainty range i.e. 95%-confidence interval using e.g. Matlab (2017b) or Python (3.6). Latin Hypercube Sampling (LHS) [45] or Sobol sequences [46] can be utilized for the probabilistic sampling over the components properties value space [47]. In this study the probability of uncertainty is assumed to follow a normal distribution. However, any other distribution is also possible. The rank-based method for correlation control of Iman and Conover [48] allows taking correlations between the property parameters into account. This is necessary, when parameters are not completely independent, as often is the case for property models.

4) The Monte Carlo samples are evaluated in the process model executed by the process simulator. In this work the PRO/II COM server is used, which provides read and write access to property information in PRO/II. This is done with the Python-COM interface.

5) Uncertainty analysis: The process model output uncertainty is quantified [41]. The Monte Carlo results provide a distribution function for the desired process model output of PRO/II. This can be analysed using mean and percentile calculations. Hence, the 95% confidence interval of the PRO/II output with respect to the corresponding input property values can be obtained.

6) Sensitivity analysis: Several sensitivity analysis methods exist which can be performed via Monte Carlo simulations or machine-learning methods. In this work we present and apply variance based sensitivity analysis (SA) which is either performed via Monte Carlo simulations or by first constructing a surrogate (response surface) model and then calculating the sensitivity indices. The theory on the sensitivity analysis methods applied in this work can be found in the appendix.

To this end the methodology has been presented and Table 5.9 summarises the above steps.

Table 5.9: Methodology for uncertainty and sensitivity analysis using a process simulator such as PRO/II.

#	Step	Description	Output
1	Problem definition and analysis	Description of process and applied property models	$y = f(\theta)$
2	Define set of variables for property models	Input variables of correlations or models	$\theta = \{x_1, ..., x_M\}$
3	Retrieve uncertainty data	Mean and standard deviation	μ_i, σ_i
4	Sampling over properties' confidence intervals	LHS or Sobol random sequences to generate sampling hypercube	X_1, ..., X_k $\boldsymbol{\Omega}$
5	Monte Carlo simulations	Run process simulations and store outputs	$\boldsymbol{y}$
6	Uncertainty analysis	Mean and 95% confidence interval of output due to uncertainties in θ	$y_i \pm \sigma_i$
7	Sensitivity analysis	Retrieval of first, total and interaction sensitivity indices	$S1_i$, ST_i $S_{i,j}$

Uncertainty analysis

The extended Antoine equation coefficients presented by Lim et al. [39] were provided to PRO/II as these correlations were also used by Tehlah et al. [37]. No confidence bounds were provided by Lim et al. and thus the effect of uncertainties in the vapour pressure estimates couldn't be assessed in this study. The uncertainties of the critical temperature, critical pressure and acentric factor in respect to each component are calculated from the data provided by different sources. Estimated values for T_C and P_C (without uncertainties) were taken from Diaz-Tovar [49]. The publication by Lim et al. [39] applied the prediction methods by Dorn and Brunner [50] and Pitzer [51] providing the property values for T_C, P_C and ω without uncertainties. Further, we used the property prediction tool ProPred [52] which is part of the KT-Consortium

software ICAS. With this application T_C, P_C and ω are predicted with the Marrero-Gani [53] or the Constantinou-Gani [54] group contribution models. No uncertainties were reported with any of these data resources or prediction models and therefore we used the experimental and predicted values to calculate the mean and standard deviation for each property as seen in Table 5.10 and 5.11.

In PRO/II 10.1 the Bio-Lib 10.1 (BIOFUELS) database stores the property values for tripalmitin and triolein, the PRO/II SIMSCI database provided the values for oleic acid and the KT-Consortium LIPIDS database holds the property data for α-tocopherol and β-carotene. The fill from structures option was selected in PRO/II to obtain values for the ideal gas enthalpy, liquid/-vapour thermal conductivity, liquid/vapour viscosity and surface tension for β-carotene and α-tocopherol.

The mean and standard deviation are retrieved from the data and used for the following sensitivity analysis.

Table 5.10: Experimental data and predicted data of properties for individual components of raw material palm oil.

	Prediction methods with regressed data sets									
	Diaz-Tovar		**Dohrn and Brunner**		**Pritzer**	**Marrero-Gani**		**Constantinou-Gani**		
Component	T_C	P_C	T_C	P_C	ω	T_C	P_C	T_C	P_C	ω
Tripalmitin	1017.47	753.85	947.10	396.82	1.6500	1056.51	870	-	-	2.177
Triolein	1039.12	726.53	954.10	360.15	1.8004	1088.68	847	-	-	2.299
Oleic acid	781	1390	813.56	1250.2	0.8104	841.41	1449	-	-	1.151
alpha-Tocopherol	857.4	1070	936.93	838.45	1.1946	962.85	1207	-	-	-
beta-Carotene	905.4	708.15	1031.1	678.41	1.6255	-	-	905.4	503	1.336

Table 5.11: Mean and standard deviation of properties for individual components of raw material palm oil calculated from the experimental data and predicted data in Table 5.10.

	Mean and standard deviation ($\mu \pm \sigma$)		
Component	T_C [K]	P_C [kPa]	ω [-]
Tripalmitin	1007.03 ± 45.27	674 ± 201	1.914 ± 0.264
Triolein	1027.3 ± 55.6	645 ± 207	2.050 ± 0.249
Oleic acid	811.99 ± 24.69	1363 ± 83	0.981 ± 0.170
alpha-Tocopherol	919.06 ± 44.87	1038 ± 152	1.195 ± 0.0001
beta-Carotene	947.3 ± 59.3	630 ± 91	1.481 ± 0.145

Sensitivity analysis

Six simulation runs were performed with sample sizes N of 32, 65, 98, 131, 313, 2188 and sampled values based on Saltelli's extension of the Sobol sequence [55]. These were normally distributed with the means and standard deviations for each of the 3 property parameters of each of the 5 components (Table 5.11). The generated sampling hypercube is used for performing Sobol sensitivity analysis and the obtained sensitivity indices $S1$ and ST are summarised in Table 5.12. The standard deviation is calculated with $\sigma = \sqrt{\sum_j \sigma_j^2}$. Standard deviation values with a negative $S1$ value are neglected in the summation since they have converged close to zero and are treated as non-influential parameters.

The 95% confidence interval of the β-carotene fraction at the bottom is 0.0008169 and 0.0008193 with the mean being 0.0008181 based on the simulation with a sample size of N=2188. This allows the engineer to assess if the needed purity for the β-carotene product lies within the uncertainty bounds. If this is not the case then the sensitivity analysis in this section will help to identify the properties which have to be revised through experiments or literature research to improve the output prediction of the process at hand.

As can be seen in Figure 5.16 from the first order sensitivity index, the critical temperature of β-carotene ($T_{C,Carotene}$), the critical pressures of tripalmitin ($P_{C,Tripalmitin}$) and triolein ($P_{C,Triolein}$) have a main effect on the β-carotene product fraction followed by $P_{C,Carotene}$ and $\omega_{Carotene}$. This shows, apart from the uncertainty of the property $T_{C,Carotene}$, that $P_{C,Tripalmitin}$ and $P_{C,Triolein}$ can be properties which have to be revised although they don't belong to the wanted product component fraction which we use as the output for the sensitivity analysis. We can also see that there is a high amount of interactions between the property parameters when comparing the values between S1 and ST for each parameter. This conclusion we can also draw because the sum of $S1_{i,j}$ doesn't add up to 1. Table 9.6 in the appendix presents the interaction matrix. Figure 5.17 visualises the summation of $T_{C,i}$, $P_{C,i}$ and ω_i, indicating which chemical species have the highest effect on the output. From these results we can conclude that β-carotene has the highest effect on the β-carotene fraction at the bottom of the molecular distillation unit and also a high interaction between $T_{C,Carotene}$ and $P_{C,Carotene}$ or $\omega_{Carotene}$ can be identified although these values are highly uncertain and even a higher sample size N is needed to obtain more accurate values for the contribution of the individual interactions.

When tracking the time for evaluating all sample sets we can observe a

Table 5.12: Sobol SA of property variables used by the SRK EOS and it's effect on beta-carotene recovery with 70016 model evaluations (sample size N=2188).

Component	$S1_{j,k} \pm \sigma_{S1,j,k}$	$ST_{j,k} \pm \sigma_{ST,j,k}$
$T_{C,Tripalmitin}$	-0.000587 ± 0.004592	0.016667 ± 0.017378
$P_{C,Tripalmitin}$	0.090428 ± 0.044442	0.221925 ± 0.044754
$\omega_{Tripalmitin}$	0.002334 ± 0.007700	0.032921 ± 0.018697
$\sum_j S1_{j,Tripalmitin}$	0.092762 ± 0.045104	0.271513 ± 0.051522
$T_{C,Triolein}$	0.004282 ± 0.006385	0.014645 ± 0.015544
$P_{C,Triolein}$	0.088856 ± 0.040798	0.214896 ± 0.046402
$\omega_{Triolein}$	-0.002196 ± 0.006423	0.025806 ± 0.017340
$\sum_j S1_{j,Triolein}$	0.093138 ± 0.041295	0.255347 ± 0.051918
$T_{C,OleicAcid}$	0.002537 ± 0.006041	0.012168 ± 0.008537
$P_{C,OleicAcid}$	0.001808 ± 0.002213	0.000584 ± 0.000635
$\omega_{OleicAcid}$	-0.000080 ± 0.000146	0.000269 ± 0.000608
$\sum_j S1_{j,OleicAcid}$	0.004345 ± 0.006434	0.013021 ± 0.008582
$T_{C,Tocopherol}$	-0.000058 ± 0.000197	0.000275 ± 0.000608
$P_{C,Tocopherol}$	-0.000092 ± 0.000155	0.000270 ± 0.000608
$\omega_{Tocopherol}$	-0.000080 ± 0.000146	0.000269 ± 0.000608
$\sum_j S1_{j,Tocopherol}$		0.000814 ± 0.001053
$T_{C,Carotene}$	0.427228 ± 0.084424	0.634004 ± 0.072671
$P_{C,Carotene}$	0.025124 ± 0.027615	0.169324 ± 0.059240
$\omega_{Carotene}$	0.058044 ± 0.028075	0.185940 ± 0.033751
$\sum_j S1_{j,Carotene}$	0.510396 ± 0.093157	0.989268 ± 0.099647
$\sum_k \sum_j S1_{j,k}$	0.7006 ± 0.111621	1.529963 ± 0.123912

Table 5.13: Comparison between Sobol SA with different sampling hypercube sizes.

	Run 1	Run 2	Run 3	Run 4	Run 5	Run 6
Sample size N	32	65	98	131	313	2188
Number of evaluations (N*(2*M+2))	1024	2080	3136	4192	10016	70016
Run time [min]	69.73	143.52	226.25	298.27	865.45	6104.99

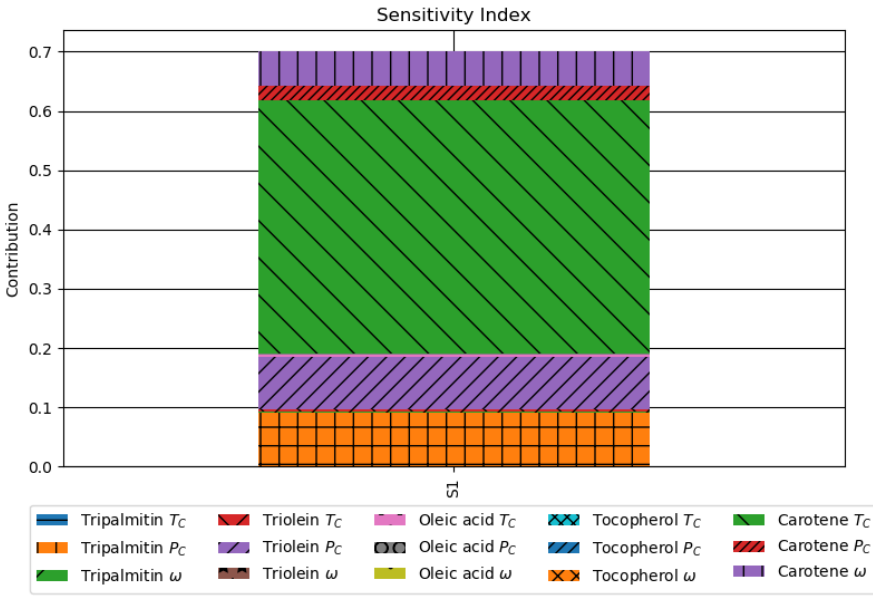


Figure 5.16: Sensitivity index bar plot for $S1_{i,j}$.

linear increase (Figure 5.18) of the run time. The bottle neck is the Monte Carlo simulation step encompassing the data transmission between the COM-interface and PRO/II, populating the property variables within the process simulator, simulating the process and storing the final output vector. The sensitivity analysis is performed with the generated input-output data and takes less than 10 seconds. The Monte Carlo procedure won't be feasible for commercial process simulators if the evaluations are not run in parallel on multiple processors. The total time which the evaluations takes for run 6 is 6105 min (about 101 h) and even on a multicore computer with 4 processors would take too long performing the full Monte Carlo simulations if the results are needed within seconds or several minutes. This situation can of course change if computers for the general industrial user will be on the market with a high number of processors.

But since methods exist which can reduce the number of evaluations needed, the next section shows that polynomial chaos expansion is a promising alternative to conduct sensitivity analysis with less computational effort

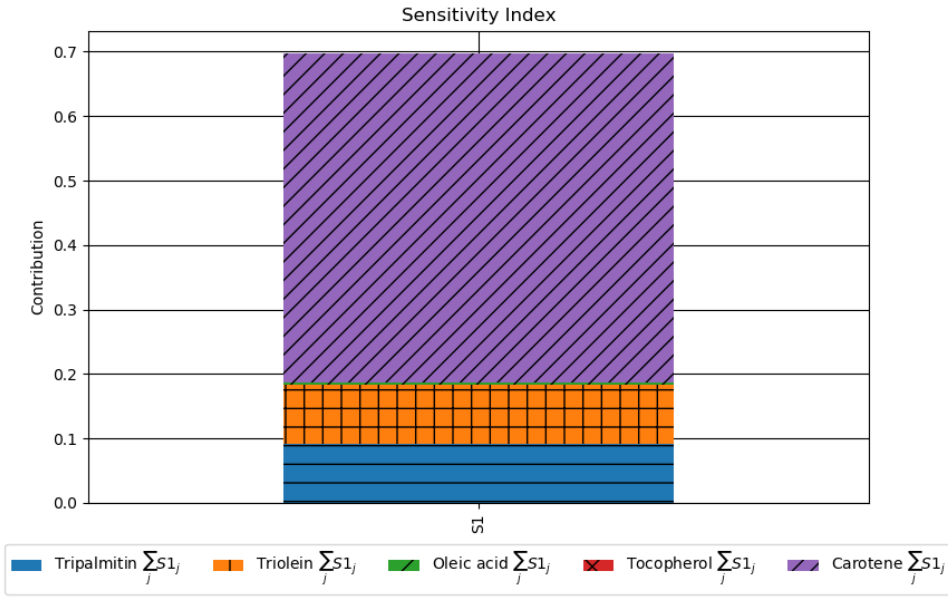


Figure 5.17: Sensitivity index bar plot for $\sum_j S1_{i,j}$.

and close to the results obtained with Monte Carlo based SA.

PCE based sensitivity analysis

The previous section showed that the retrieval of Sobol sensitivity indices from the full Monte Carlo approach resulted in high confidence intervals on the sensitivity indices for low sample size numbers and a high number of samples is needed to reduce these (Figure 5.19). We therefore applied a surrogate based sensitivity analysis where first an expansion of polynomial terms is regressed to 1024 process evaluations. Our problem at hand has a dimensionality of M=15 and the polynomial expansion we obtain fits the data with a coefficient of determination R^2=0.9148. The polynomial expansion has a degree of p=3 and 89 coefficients. In the second step the PCE is used to retrieve the sensitivity indices. Table 5.14 compares the obtained sensitivity indices with the previous obtained indices from the full Monte Carlo (MC) sensitivity analysis (SA). The results show that we are able to identify the same property parameters

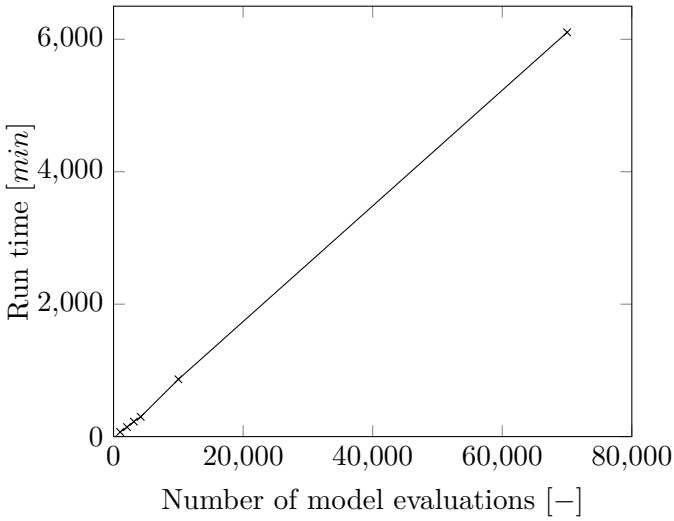


Figure 5.18: Run time over number of model evaluations with increasing model evaluations for Monte Carlo based sensitivity analysis.

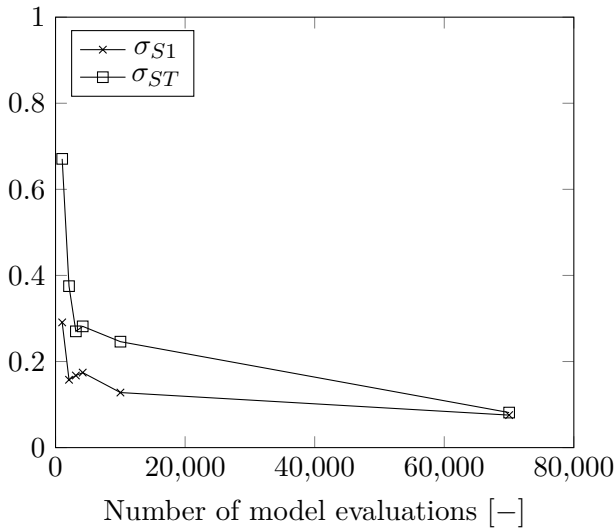


Figure 5.19: Standard deviation of S1 and ST for $T_{C,Carotene}$ with increasing model evaluations for Monte Carlo based sensitivity analysis.

compared to MC SA with $T_{C,Carotene}$ being the most important. These results are obtained via PCE based SA with about 68-fold less model evaluations.

Table 5.14: Comparison between direct Monte Carlo based Sobol SA and PCE based Sobol SA.

	PCE SA		**MC SA**	
Property	S1	ST	S1	ST
$T_{C,Tripalmitin}$	-	0.00	-	0.02
$P_{C,Tripalmitin}$	0.04	0.13	0.09	0.22
$\omega_{Tripalmitin}$	0.00	0.00	0.00	0.03
$T_{C,Triolein}$	0.00	0.00	0.00	0.01
$P_{C,Triolein}$	0.03	0.12	0.09	0.21
$\omega_{Triolein}$	0.00	0.00	-	0.03
$T_{C,OleicAcid}$	-	0.00	0.00	0.01
$P_{C,OleicAcid}$	-	-	0.00	0.00
$\omega_{OleicAcid}$	-	0.01	-	0.00
$T_{C,Tocopherol}$	-	0.00	-	0.00
$P_{C,Tocopherol}$	-	0.00	-	0.00
$\omega_{Tocopherol}$	-	0	-	0.00
$T_{C,Carotene}$	0.55	0.69	0.43	0.63
$P_{C,Carotene}$	0.08	0.16	0.03	0.17
$\omega_{Carotene}$	0.07	0.15	0.06	0.19
$\sum S_{j,k}$	0.77	1.26	0.75	1.52
Model evaluations	1024		70016	

5.2.4 Conclusion

It was shown that machine-learning based methods such as polynomial chaos expansion can reduce the computational time needed to perform sensitivity analysis. Fully data-driven methods use the input-output data and treat the process simulator as a black box. Therefore, the multiple evaluation of a process flowsheet should be performed in parallel with sequential process simulators. To this date no modular process simulator such as Aspen or PRO/II supports this functionality and this would be a major improvement to adapt Monte Carlo and machine-learning based methods by commercial process simulators. It is therefore recommended to implement parallel evaluations of process models on multiple cores (e.g. with Co-array Fortran and/or MPI [56–

58]) to reduce the computational time due to the high number of evaluations in case of the full Monte Carlo approach or to increase the speed of machine learning methods. This is necessary and important to realise the growing potential of these methods. It would be worthwile to test the performance of different machine learning techniques in respect to sensitivity analysis methods in the future, e.g. the comparison between neural networks, Gaussian process regression and polynomial chaos expansion.

Monte Carlo and machine learning methods can be integrated in commercial process simulators and applied by industrial users. Uncertainties of experimental data or estimated values should always be reported and we suggest that process simulators should take the uncertainties in properties as given and evaluate process models in respect to these uncertainty ranges. It was demonstrated in this work that small changes in property uncertainties can have a major effect on the process flowsheet output.

The study presented a Monte Carlo based methodology and tools integration with scripting languages such as Python or Matlab for enabling property uncertainty and sensitivity analysis with a commercial process simulator. The framework enables process design engineers to perform robustness analysis of process design effectively and increases the value of commercial process simulators since considering uncertainty gains more and more importance in the process systems engineering community. The study shows that it is possible to use Monte Carlo techniques with commercial process simulators, which is currently not state-of-the-art in industrial practice. Further we highlight that machine learning based techniques can be applied to reduce the computational expensive full Monte Carlo approach. This was exemplified with polynomial chaos expansion. To this end, the generic nature of the methodology was successfully implemented with respect to a molecular distillation process. The uncertainties in properties such as critical temperature $T_{C,i}$, critical pressure $P_{C,i}$ and acentric factor ω_i were propagated through the SRK EOS and the process models in PRO/II. Analysis of the molecular distillation showed that the uncertainty of the recovery of β-carotene can be apportioned mostly to $T_{C,Carotene}$ as the input properties' uncertainties propagate from the SRK EOS through the calculations of the molecular distillation unit. The mostly higher values of the total order indices indicate a high degree of interaction between the parameters. This can be verified by evaluating the interaction sensitivity indices as shown in Table 9.6 in the appendix. The methodology we presented is also applicable to the case of analysing multiple outputs of a process.

5.3 Solvent (extractive) crystallisation

5.3.1 Introduction

Crystallisation allows to separate compounds by means of cooling, evaporation or through a chemical reaction between two phases which increase a species in the solution until it is supersaturated. Thus, the driving force of a solvent crystallisation process is the solubility gradient over temperature which differs from substance to substance. The liquid mixture of components are in solid-liquid equilibrium where the solid phase crystallises in three steps [1]. In the first step supersaturation is achieved which is the driving force for nucleation (step two) and as the third step crystal growth will increase the product yield. The equilibrium is reached where the Gibbs free energy is at its minimum and the solid and liquid phases are stable. In the next sections the calculation of the solid-liquid equilibrium with the Wilson activity coefficient model is explained. Thereafter, Michelsen's tangent plane criteria for the phase stability test and the formulation of the Gibbs energy minimisation algorithm is presented. The solvent crystallisation model is then validated with experimental data from literature. The industrial crystallisation of fatty acids with methanol as the polar solvent is also known as the Emersol process [59, 60] where a multitubular crystalliser with scraper blades [61] is utilised. In this work only pure solid phases are assumed and polymorphism is not taken into account.

5.3.2 Model description

Solid-liquid equilibrium and Wilson activity coefficient model

The thermodynamic equilibrium is defined as the state where the chemical potentials of all phases in a mixture are equal. Thus, the solid-liquid equilibrium can be defined as [62]:

$$\mu_i^j = \mu_i^l \tag{5.24}$$

where μ_i^j and μ_i^l are the chemical potentials in the solid phase j and liquid phase l for component i. There exist N components and P phases where P - 1 phases are solid and the remaining phase is liquid [63].

The chemical potential μ_i at a certain temperature is dependent on temperature T, the liquid activity coefficient γ_i and the component fraction of species i. Further, the chemical potential is related to a reference potential

$\mu_{0,i}$ and the expressions for the liquid and solid chemical potentials are:

$$\mu_i^l = \mu_{0,i}^l + RTln(\gamma_i^l x_i^l) \tag{5.25}$$

$$\mu_i^j = \mu_{0,i}^j + RTln(\gamma_i^j x_i^j) \tag{5.26}$$

Equations 5.16, 5.17 and 5.18 give the following relation:

$$ln(\frac{\gamma_i^j x_i^j}{\gamma_i^l x_i^l}) = \frac{\mu_{0,i}^l - \mu_{0,i}^j}{RT} \tag{5.27}$$

The differential change in chemical potential is defined by [62]:

$$d\mu_i = -S_i dT + V_i dP \tag{5.28}$$

where S_i and V_i are the entropy and volume for species i. As Clausius showed, the heat transfer along an isotherm of a system follows proportional behaviour and thus entropy S_i was defined as an extensive state variable of a thermodynamic system [64]. The proportional relation can be written in differential form as:

$$\Delta S_i = \Delta H_i / T \tag{5.29}$$

The enthalpy of fusion is calculated from the molar enthalpy of fusion, heat capacity and the difference between the temperature T of the system and the temperature of fusion $T_{f,i}$ of species i:

$$\Delta H_{f,i} = \Delta H_{f,m,i} + \Delta c_{p,i}(T - T_{f,i}) \tag{5.30}$$

With equations 5.19 to 5.22 we can formulate the following expression:

$$ln(\frac{\gamma_i^j x_i^j}{\gamma_i^l x_i^l}) = \frac{\Delta H_{f,i}^j}{R}(\frac{1}{T} - \frac{1}{T_{f,i}}) - \frac{\Delta c_{p,i}}{R}\frac{T_{f,i} - T}{T} + \frac{\Delta c_{p,i}}{R} ln(\frac{T_{f,i}}{T}) \tag{5.31}$$

As stated before, the solid phase is assumed being a pure component and the relation $\gamma_i^j x_i^j = 1$ can be considered [65]. The Δc_p values and $T_{f,i} - T$ term for triglycerides and fatty acids are comparable small and thus the previous equation simplifies to:

$$ln(\frac{1}{\gamma_i^l x_i^l}) = \frac{\Delta H_{f,i}^j}{R}(\frac{1}{T} - \frac{1}{T_{f,i}}) \tag{5.32}$$

The activity coefficients can be calculated with an activity or equation of state model (e.g. Wilson, UNIFAC, Soave-Redlich-Kwong or Margules [65–67]). In this work the Wilson model is applied. The needed binary interaction parameters for the Wilson model are defined as [1]:

$$A_{ij} = \frac{V_j}{V_i} exp(\frac{-(\lambda_{ij} - \lambda_{ji})}{RT}) \tag{5.33}$$

V_j and V_i are the molar volumes of component j and i in the liquid phase whereas λ_{ij} and λ_{ji} are the binary energy parameters.

The liquid activity coefficient is calculated with [1]:

$$ln\gamma_k = -\sum_{j=1}^{N} x_j A_{kj} + 1 - \sum_{i=1}^{N} \frac{x_i A_{ik}}{\sum_{j=1}^{N} x_j A_{ij}} \tag{5.34}$$

Phase stability

A phase stability test has to be performed to evaluate if the addition of a new phase to the thermodynamic system in equilibrium will lead to a decrease in the Gibbs free energy ($\Delta G < 0$). The difference between the two energy situations can be expressed with the equations ...:

$$G^{(II)} - G^{(I)} = \sum^{L} \sum^{n} N_j y_i^j \mu_i^j - \sum^{n} n_i \mu_i^I = \sum^{L} (\sum^{n} y_i^j (\mu_i^j - \mu_i^l)) = \sum^{L} N_j F_j \tag{5.35}$$

For this a tangent to the Gibbs free energy curve can be drawn at the point of the current composition and if the Gibbs curve lies above this tangent then the mixture is stable. The stability criterion is:

$$F_L = \sum y_i^L (\mu_i^L - \mu_i^{(I)}) \geq 0 \tag{5.36}$$

F_L is positive for any composition y^L if the minimum of F_L is positive and therefore the derivative $\frac{dF_L}{dy^L}$ is set to zero to obtain the composition at the minimum value of F_L.

$$K = \mu_i - \mu_i^{(I)} = \mu_j - \mu_j^{(I)} \tag{5.37}$$

is constant for $\min(F_L)$ and therefore stability is provided for $K > 0$.

$$k = \frac{K}{RT} = \frac{\mu_i^{0,L} - \mu_i^I}{RT} + ln(\gamma_i^L) + ln(y_i^L) \tag{5.38}$$

if the addition of an infinitesimal amount of a new phase and k becomes lower than 0 then the Gibbs energy will decrease and the current system without the additional phase is unstable. The composition of the new trial phase is:

$$y_i = \frac{y_i^L e^{-k}}{\sum_i y_i^L e^{-k}} \tag{5.39}$$

The tangent plane criterion by Michelsen [68] is defined as:

$$ln(Y_i) = ln(y_i^L e^{-k}) = \frac{\mu_i^{(I)} - \mu_i^{0,L}}{RT} - ln(\gamma_i^L) \tag{5.40}$$

where k is the stability criterion variable. A stable equilibrium is present if $k > 0$. The term $y_i^L e^{-k}$ is depicted as Y_i in Figure 5.20 and has to be lower than zero for a stable equilibrium.

Gibbs free energy minimisation

The phase equilibrium is described by a set of nonlinear equations which have to be solved by minimising the total Gibbs energy as the objective function to satisfy the equilibrium condition (Eq. 5.21).

The Gibbs free energy can be expressed as [1]:

$$G = \Delta g_{solid} + \Delta g_{liq} \tag{5.41}$$

where Δg_{liq} and Δg_{solid} are:

$$\Delta g_{solid} = RT \sum_{i=1}^{N} \phi_{solid,i} \frac{\Delta H_{f,i}}{R} (\frac{1}{T} - \frac{1}{T_{f,i}}) \tag{5.42}$$

$$\Delta g_{liq} = \phi_{liq} RT \sum_{i=1}^{N} x_i ln(\gamma_i x_i) \tag{5.43}$$

For any given system the mole balance constraints must be satisfied in respect to the total moles in each phase of species i and the total moles of species i in the whole system. Thus, the Gibbs free energy minimisation problem can be formulated as follows:

$$\begin{aligned} \text{minimise} \quad & G = \Delta g_{solid} + \Delta g_{liq} \\ \text{subject to} \quad & \sum_{i=1}^{N} x_i^j = 1 \\ & \sum_{j=1}^{P} x_i^j \phi^j = z_i \end{aligned} \tag{5.44}$$

and is solved in this work via the sequential least squares programming method in the scientific Python package SciPy [69].

Solid flash algorithm

The algorithm for the composition calculation of the liquid phase of the product of the crystalliser is depicted with the flow diagram in Figure 5.20 and can be divided into the following steps:

1. A first estimation is made of the number of phases, the present amount of each phase ϕ_j and of their compositions x_i^j.

2. Calculation of the activity coefficient(s).

3. Perform new estimation of the amount of phases and the compositions with the newly calculated activity coefficients and repeat the previous and this step until the Gibbs energy converged to the global minimum.

4. Perform the stability test and check if estimates give a stable mixture.

5. If the mixture is unstable repeat the previous steps with an additional phase.

6. If the mixture is stable the final result is obtained.

The input parameters to the algorithm are the interaction parameters of the Wilson model $\Delta\lambda_{ij}$, the operating temperature of the crystalliser T, the heat of fusion $\Delta H_{f,i}$, the temperature of fusion $T_{f,i}$ and the molar volumes V_i of each component. The independent variables are x_i^j and ϕ^j and the dependent variables are A_{ij} and γ_i^j.

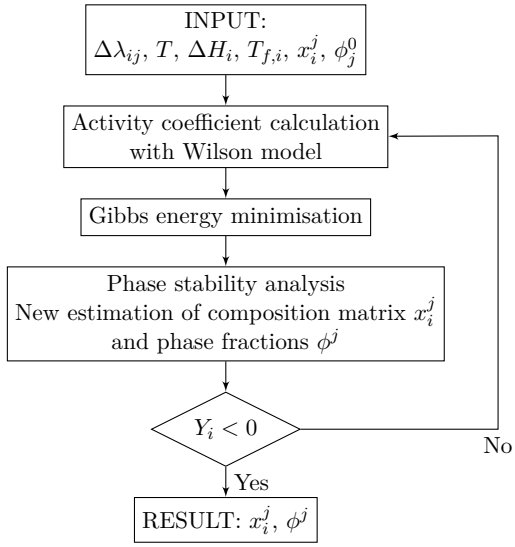


Figure 5.20: Solid flash algorithm.

5.3.3 Analysis

The model is validated with the property data given in Table 5.15. The Wilson parameters are documented in the appendix. Wales obtained the energy parameters by regressing experimental data sets for solid-liquid equilibria [70].

Table 5.15: Property data for crystallisation calculations.

Components	$T_{f,i}$ [°C]	V_i [cm^3/mol]	$\Delta H_{f,i}$ [J/mol]
Palmitic acid (16:0)	62.8	300.97	53973
Stearic acid (18:0)	69.6	302.38	61222
Oleic acid (18:1)	16.30	315.61	39598
Linoleic acid (18:2)	-5.20	311.609	31200
Acetone	-95	74.03	5720

Process configurations

Four process configurations have been studied to separate a mixture of stearic, palmitic, linoleic and oleic fatty acids (Figure 5.21) for a feed flowrate of 36483.125 mol/h. The solvent for the crystallisation process is acetone. The

Table 5.16: Feed composition to crystalliser.

Components	Feed composition
Palmitic acid (16:0)	0.005790
Stearic acid (18:0)	0.003613
Oleic acid (18:1)	0.02709
Linoleic acid (18:2)	0.04520
Acetone	0.918305

Table 5.17: Validation of model with results of the composition of the liquid product from the crystalliser.

	Experimental data			Predictions by Wale			Predictions by model in this work		
Temperature [°C]	$x_{Stearic}$	x_{Oleic}	$x_{Acetone}$	$x_{Stearic}$	x_{Oleic}	$x_{Acetone}$	$x_{Stearic}$	x_{Oleic}	$x_{Acetone}$
0	0.0009	0.0124	0.9867	0.0006	0.0124	0.9870	0.0011	0.0124	0.9864
-10	0.0003	0.0146	0.9851	0.0003	0.0141	0.9856	0.0004	0.0146	0.9850
-20	0.0001	0.0143	0.9856	0.0002	0.0151	0.9847	0.00016	0.0267	0.9731
-30	2.0e-5	0.0061	0.9939	2.9e-5	0.0043	0.9957	3.75e-5	0.00649	0.9934
-40	6e-5	0.0025	0.9975	3.1e-5	0.0018	0.9981	1.03e-5	0.007	0.9930

equipment cost were calculated on the basis of Guthrie's module costing method. The operating cost for the distillation and crystallisation units were calculated with the following relations,

$$C_{OP}^{Dist} = Q_{Cond} c_{CW} + Q_{Reb} c_{Steam} \tag{5.45}$$

Q_{cond} and Q_{Reb} are the heat duties of the condenser and reboiler. c_{CW} and c_{Steam} are the price coefficients for cooling water and steam.

$$C_{OP}^{Cryst} = Q_C c_C \tag{5.46}$$

Q_C is the energy for cooling and c_C is the price coefficient for the cooling medium.

Q_C is calculated with the overall heat transfer coefficient U, the heat transfer area A and the logarithmic mean temperature difference ΔT_{LM}:

$$Q_C = U A \Delta T_{LM} \tag{5.47}$$

Condenser and reboiler duties for the distillation columns, density and heat capacity values were calculated with the PRO/II process simulator. Data for the price coefficients was gathered from literature [71] and regressed to estimate the cost coefficient for a certain cooling temperature.

The results of this preliminary, qualitative economic evaluation showed that the first process structure is the most economic in regards to equipment and operating cost. The detailed information of the economic evaluation can be found in the work by Aktas [72]. The product purity of oleic acid, stearic acid and palmitic acid are 99% and the product purity of linoleic acid is 96% respectively.

Table 5.18: Price coefficients c_C for cooling medium.

Temperature [°C]	Price [$/GJ]
15	4.0
5	5.0
-20	8.0
-50	14.0

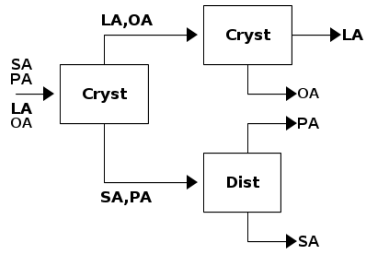
Table 5.19: Operating conditions of the unit operations for the different process configurations.

Process structure	Operating conditions
1	$T_{Cryst1} = 242.2$ K, $T_{Cryst2} = 230$ K, $P_{Dist} = 1.2$ kPa
2	$T_{Cryst1} = 274$ K, $T_{Cryst2} = 243.5$ K, $T_{Cryst3} = 230$ K, $P_{Dist} = 1.2$ kPa
3	$T_{Cryst1} = 274$ K, $T_{Cryst2} = 242.15$ K, $T_{Cryst3} = 230$ K, $P_{Dist} = 1.2$ kPa
4	$T_{Cryst1} = 246$ K, $T_{Cryst2} = 220$ K, $P_{Dist} = 1.2$ kPa

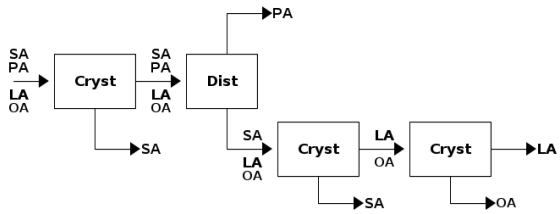
5.3.4 Conclusion

A solid flash algorithm has been developed to model a solvent crystallisation process. The results aligned well with the experimental data. Four different process structures have been evaluated to gain some preliminary knowledge on how solvent crystallisation can be implemented for separating fatty acid mixtures. The process with two crystallisers in series and a distillation column after the first crystalliser in parallel to the second crystalliser, gave the most feasible process in regards to capital and operating cost. An optimisation loop has to still be implemented to find the optimal operation conditions.

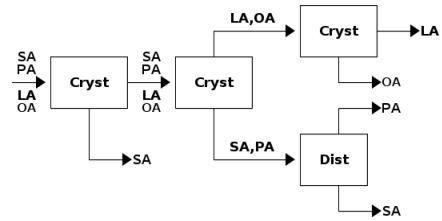
Process structure 1:



Process structure 2:



Process structure 3:



Process structure 4:

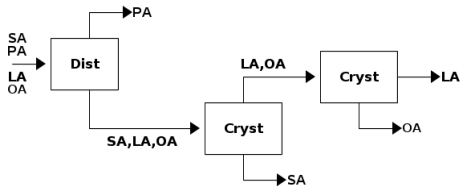


Figure 5.21: Process configurations for separating a mixture of stearic (SA), palmitic (PA), linoleic (LA) and oleic (OA) fatty acids.

The algorithm can be extended to support polymorphic crystallisation into the major α, β and β' forms. For this extension, the number of trial splits needs to be increased from one binary flash to four flash calculations (β-β, β-β', β-α and β-liquid split) [73, 74]. Solvent selection and design is also an important aspect to solvent crystallisation and computer-aided molecular design (CAMD) can be applied to find suitable economic and sustainable solvents.

Bibliography

[1] S. N. Wale, "Separation of fatty acids by extractive crystallization," English, PhD thesis, 1995.

[2] W. S. Singleton, "Phase investigations of fats. ii. systems containing oleic and stearic acids and an organic solvent," *Journal of the American Oil Chemists' Society*, vol. 25, no. 1, pp. 15–20, 1948. DOI: 10.1007/BF02553634.

[3] T. A. Patil, T. S. Raghunathan, and H. S. Shankar, "Thermal hydrolysis of vegetable oils and fats. 2. hydrolysis in continuous stirred tank reactor," *Industrial Engineering Chemistry Research*, vol. 27, no. 5, pp. 735–739, 1988. DOI: 10.1021/ie00077a002.

[4] H. Forero-Hernandez, M. N. Jones, B. Sarup, A. D. Jensen, and G. Sin, "A simplified kinetic and mass transfer modelling of the thermal hydrolysis of vegetable oils," *Computer Aided Chemical Engineering*, vol. 40, pp. 1177–1182, 2017.

[5] R. Alenezi, G. A. Leeke, R. C. D. Santos, and A. R. Khan, "Hydrolysis kinetics of sunflower oil under subcritical water conditions," *Chemical Engineering Research and Design*, vol. 87, no. 6, pp. 867–873, 2009.

[6] T. A. Patil, D. N. Butala, T. S. Raghunathan, and H. S. Shankar, "Thermal hydrolysis of vegetable oils and fats. 1. reaction kinetics," *Industrial Engineering Chemistry Research*, vol. 27, no. 5, pp. 727–735, 1988.

[7] V. K. Aniya, R. K. Muktham, K. Alka, and B. Satyavathi, "Modeling and simulation of batch kinetics of non-edible karanja oil for biodiesel production," *Fuel*, vol. 161, pp. 137–145, 2015.

[8] G. V. Jeffreys, V. G. Jenson, and F. R. Miles, "The analysis of a continiuous fat-hydrolysing column," *Transactions of the Institution of Chemical Engineers*, vol. 39, no. nil, pp. 389–396, 1961.

[9] A. Sturzenegger and H. Sturm, "Hydrolysis of fats and high temperatures," *Industrial and Engineering Chemistry*, vol. 43, no. 2, pp. 510–515, 1951.

[10] V. Mills and H. K. McClain, "Fat hydrolysis," *Industrial and Engineering Chemistry*, vol. 41, pp. 1982–1985, 1949.

[11] G. W. Minard and A. I. Johnson, "Limiting flow and holdup in a spray extraction column," *Chemical Engineering Progress*, vol. 48, no. 62, 1952.

[12] Rifai, Elnashaie, and Kafafi, "Analysis of a countercurrent tallow splitting column," *Trans. Instn. Chem. Eng*, vol. 55, pp. 59–63, 1977.

[13] P. D. Namdev, T. A. Patil, T. S. Raghunathan, and H. S. Shankar, "Thermal hydrolysis of vegetable oils and fats. 3. an analysis of design alternatives," *Industrial Engineering Chemistry Research*, vol. 27, pp. 739–743, 1988.

[14] M. Attarakih, T. Albaraghthi, M. Abu-Khader, Z. Al-Hamamre, and H. Bart, "Mathematical modeling of high-pressure oil-splitting reactor using a reduced population balance model," *Chemical Engineering Science*, vol. 84, pp. 276–291, 2012.

[15] B. O. Beyaert, L. Lapidus, and J. C. Elgin, "The mechanics of vertical moving liquid-liquid fluidized systems: Ii. countercurrent flow," *AIChE Journal*, vol. 7, no. 1, pp. 46–48, 1961.

[16] L. C. van Egmond and M. L. Goossens, "Berekningen aan axiale dispersie in een operationele vetsplitter," Laboratorium voor Chemische Technologie, Tech. Rep., 1982.

[17] R. S. Ettouney, M. A. El-Rifai, A. O. Ghallab, and A. K. Anwar, "Mass transfer fluid flow interactions in perforated plate extractive reactors," *Separation Science and Technology*, vol. 50, no. 12, pp. 1794–1805, 2015. DOI: 10.1080/01496395.2015.1014057.

[18] U. Nowak and L. Weimann, "A family of newton codes for systems of highly nonlinear equations," Konrad-Zuse-Zentrum für Informationstechnik Berlin, Tech. Rep., December 1991.

[19] R. Storn and K. Price, "Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 11, no. 4, pp. 341–359, December 1997. DOI: 10.1023/A:1008202821328.

[20] W.-C. Wang, T. L. Turner, W. L. Roberts, and L. F. Stikeleather, "Direct injection of superheated steam for continuous hydrolysis reaction," *Chemical Engineering and Processing: Process Intensification*, vol. 59, pp. 52–59, 2012. DOI: 10.1016/j.cep.2012.04.003.

[21] F. Zhu, "Determining true steam prices," in *Energy and Process Optimization for the Process Industries*. John Wiley & Sons, Ltd, 2013, ch. 17, pp. 366–385. DOI: 10.1002/9781118782507.ch17.

[22] M. Goedkoop and R. Spriensma, "The eco-indicator99: A damage oriented method for life cycle impact assessment: Methodology report," PRe Consultants B.V., Tech. Rep., Jun. 2001, pp. 1–144.

[23] M. Thompson, R. Ellis, and A. Wildavsky, *Cultural Theory*. Westview Press, 1990.

[24] P. Hofstetter, *Perspectives in life cycle impact assessment :* eng. Kluwer, 1998, 484 s.

[25] G. P. Rangaiah and S. Sharma, *Differential Evolution in Chemical Engineering*. World Scientific, 2017. DOI: 10.1142/10379.

[26] L. Landress, "Fatty acids (North America)," ICIS Pricing, Tech. Rep., 2014.

[27] ——, "Glycerine (US Gulf)," ICIS Pricing, Tech. Rep., 2014.

[28] Malaysian Palm Oil Council, *CPO vs SPO price*, March 2019.

[29] M. International, *Meps - world stainless steel prices*, October 2018.

[30] J. G. Cadavid, R. D. Godoy-Silva, P. C., M. Camargo, and C. Fonteix, "Biodiesel production in a counter-current reactive extraction column: Modelling, parametric identification and optimisation," *Chemical Engineering Journal*, vol. 228, pp. 717–723, 2013. DOI: 10.1016/j.cej.2013.05.040.

[31] J. Lutišan and J. Cvengroš, "Mean free path of molecules on molecular distillation," *The Chemical Engineering Journal and the Biochemical Engineering Journal*, vol. 56, no. 2, pp. 39–50, 1995. DOI: https://doi.org/10.1016/0923-0467(94)02857-7.

[32] C. Batistella, E. Moraes, R. Maciel, and M. Maciel, "Molecular distillation - rigorous modeling and simulation for recovering vitamin e from vegetable oils," *Applied Biochemistry and Biotechnology*, vol. 98, pp. 1187–1206, 2002.

[33] E. B. de Moraes, P. F. Martins, C. B. Batistella, M. E. T. Alvarez, R. M. Filho, and M. R. W. Maciel, "Molecular distillation," in *Twenty-Seventh Symposium on Biotechnology for Fuels and Chemicals*. Humana Press, 2006, pp. 1066–1076. DOI: 10.1007/978-1-59745-268-7_90.

[34] P. Martins, V. Ito, C. Batistella, and M. Maciel, "Free fatty acid separation from vegetable oil deodorizer distillate using molecular distillation process," *Separation and Purification Technology*, vol. 48, no. 1, pp. 78–84, 2006. DOI: https://doi.org/10.1016/j.seppur.2005.07.028.

[35] S. Wang, Y. Gu, Q. Liu, Y. Yao, Z. Guo, Z. Luo, and K. Cen, "Separation of bio-oil by molecular distillation," *Fuel Processing Technology*, vol. 90, no. 5, pp. 738–745, 2009. DOI: 10.1016/j.fuproc.2009.02.005.

[36] U. R. Unnithan, *Refining of edible oil rich in natural carotenes and vitamin E*, Patent, US, August 1999.

[37] N. Tehlah, P. Kaewpradit, and I. M. Mujtaba, "Development of molecular distillation based simulation and optimization of refined palm oil process based on response surface methodology," *Processes*, vol. 5, no. 3, 2017. DOI: 10.3390/pr5030040.

[38] P. M. Mathias, "A versatile phase equilibrium equation of state," *Industrial & Engineering Chemistry Process Design and Development*, vol. 22, no. 3, pp. 385–391, 1983. DOI: 10.1021/i200022a008.

[39] C. S. Lim, Z. A. Manan, and M. R. Sarmidi, "Simulation modeling of the phase behavior of palm oil-supercritical carbon dioxide," *Journal of the American Oil Chemists' Society*, vol. 80, no. 11, pp. 1147–1156, November 2003. DOI: 10.1007/s11746-003-0834-6.

[40] P. M. Mathias, H. C. Klotz, and J. M. Prausnitz, "Equation-of-state mixing rules for multicomponent mixtures: The problem of invariance," *Fluid Phase Equilibria*, vol. 67, pp. 31–44, 1991. DOI: 10.1016/0378-3812(91)90045-9.

[41] J. Frutiger, "Property uncertainty analysis and methods for optimal working fluids of thermodynamic cycles," English, PhD thesis, 2017.

[42] M. Frenkel, R. D. Chirico, V. Diky, X. Yan, Q. Dong, and C. Muzny, "Thermodata engine (tde): software implementation of the dynamic data evaluation concept," *Journal of Chemical Information and Modeling*, vol. 45, no. 4, pp. 816–838, 2005. DOI: 10.1021/ci050067b.

[43] J. Frutiger, C. Marcarie, J. Abildskov, and G. Sin, "A comprehensive methodology for development, parameter estimation, and uncertainty analysis of group contribution based property models—an application to the heat of combustion," *Journal of Chemical & Engineering Data*, vol. 61, no. 1, pp. 602–613, 2016. DOI: 10.1021/acs.jced.5b00750.

[44] J. Frutiger, I. Bell, J. P. O'Connell, K. Kroenlein, J. Abildskov, and G. Sin, "Uncertainty assessment of equations of state with application to an organic rankine cycle," *Molecular Physics*, vol. 115, no. 9-12, pp. 1225–1244, 2017. DOI: 10.1080/00268976.2016.1275856.

[45] M. McKay, R. Beckman, and W. Conover, "Comparison the three methods for selecting values of input variable in the analysis of output from a computer code," *Technometrics; (United States)*, May 1979. DOI: 10.1080/00401706.1979.10489755.

[46] I. M. Sobol', D. Asotsky, A. Kreinin, and S. Kucherenko, "Construction and comparison of high-dimensional sobol' generators," *Wilmott*, vol. 2011, no. 56, pp. 64–79, 2011. DOI: 10.1002/wilm.10056.

[47] G. Sin, K. V. Gernaey, and A. E. Lantz, "Good modeling practice for pat applications: Propagation of input uncertainty and sensitivity analysis," *Biotechnology Progress*, vol. 25, no. 4, pp. 1043–1053, 2009. DOI: 10.1002/btpr.166.

[48] R. L. Iman and W. J. Conover, "A distribution-free approach to inducing rank correlation among input variables," *Communications in Statistics - Simulation and Computation*, vol. 11, no. 3, pp. 311–334, 1982. DOI: 10.1080/03610918208812265.

[49] C. A. Diaz-Tovar, R. Gani, and B. Sarup, "Computer-aided modeling of lipid processing technology," English, PhD thesis, Jul. 2011.

[50] R. Dohrn and G. Brunner, "An estimation method to calculate Tb, Tc, Pc and omega from the liquid molar volume and the vapor pressure," ser. Proceedings of the 3rd International Symposium on Supercritical Fluids, 2016, pp. 241–248.

[51] K. S. Pitzer, "The volumetric and thermodynamic properties of fluids. i. theoretical basis and virial coefficients1," *Journal of the American Chemical Society*, vol. 77, no. 13, pp. 3427–3433, 1955. DOI: 10.1021/ja01618a001.

[52] A. S. Hukkerikar, "Development of pure component property models for chemical product-process design and analysis," English, PhD thesis, 2013.

[53] J. Marrero and R. Gani, "Group-contribution based estimation of pure component properties," *Fluid Phase Equilibria*, vol. 183-184, pp. 183–208, 2001. DOI: 10.1016/S0378-3812(01)00431-9.

[54] L. Constantinou and R. Gani, "New group contribution method for estimating properties of pure compounds," *AIChE Journal*, vol. 40, no. 10, pp. 1697–1710, 1994. DOI: 10.1002/aic.690401011.

[55] A. Saltelli, "Making best use of model evaluations to compute sensitivity indices," *Computer Physics Communications*, vol. 145, no. 2, pp. 280–297, 2002. DOI: 10.1016/S0010-4655(02)00280-1.

[56] S. Garain, D. S. Balsara, and J. Reid, "Comparing coarray fortran (CAF) with MPI for several structured mesh PDE applications," *Journal of Computational Physics*, vol. 297, pp. 237–253, 2015. DOI: 10.1016/j.jcp.2015.05.020.

[57] A. Sharma and I. Moulitsas, "MPI to coarray fortran. experiences with a cfd solver for unstructured meshes," *Scientific Programming*, vol. 2017, no. 3409647, 2017. DOI: 10.1155/2017/3409647.

[58] F. T. Tracy, T. C. Oppe, and M. K. Corcoran, "A comparison of MPI and co-array Fortran for large finite element variably saturated flow simulations," *Scalable Computing: Practice and Experience*, vol. 19, no. 4, pp. 423–432, 2018. DOI: 10.12694/scpe.v19i4.1468.

[59] R. L. Demmerle, "Emersol process: A staff report," *Industrial & Engineering Chemistry*, vol. 39, no. 2, pp. 126–131, 1947. DOI: 10.1021/ie50446a011.

[60] G. Haraldsson, "Separation of saturated/unsaturated fatty acids," *Journal of the American Oil Chemists' Society*, vol. 61, no. 2, pp. 219–222, February 1984. DOI: 10.1007/BF02678772.

[61] K. T. Zilch, "Separation of fatty acids," *Journal of the American Oil Chemists' Society*, vol. 56, no. 11Part1, 739A–742A, 1979. DOI: 10.1007/BF02667432.

[62] J. Prausnitz, R. Lichtenthaler, and E. d. Azevedo, *Molecular thermodynamics of fluid-phase equilibria*. Prentice-Hall, 1986.

[63] L. H. Wesdorp, J. A. Van Meeteren, S. De Jong, R. V. Giessen, P. Overbosch, P. A. Grootscholten, M. Struik, E. Royers, A. Don, T. De Loos, C. Peters, and I. Gandasasmita, "Liquid-multiple solid phase equilibria in fats: Theory and experiments," eng, *Fat Crystal Networks*, pp. 481–709, 2004.

[64] I. Dincer and M. A. Rosen, "Chapter 1 - thermodynamic fundamentals," in *Exergy (Second Edition)*, 2013, pp. 1–20. DOI: https://doi.org/10.1016/B978-0-08-097089-9.00001-2.

[65] M. C. Costa, M. P. Rolemberg, L. A. D. Boros, M. A. Krähenbühl, M. G. de Oliveira, and A. J. A. Meirelles, "Solid−liquid equilibrium of binary fatty acid mixtures," *Journal of Chemical & Engineering Data*, vol. 52, no. 1, pp. 30–36, 2007. DOI: 10.1021/je060146z.

[66] G. Kontogeorgis and G. Folas, *Thermodynamic Models for Industrial Applications: From Classical and Advanced Mixing Rules to Association Theories*, English. Wiley, 2010.

[67] M. C. Costa, L. A. D. Boros, J. A. Souza, M. P. Rolemberg, M. A. Krähenbühl, and A. J. A. Meirelles, "Solid–liquid equilibrium of binary mixtures containing fatty acids and triacylglycerols," *Journal of Chemical & Engineering Data*, vol. 56, no. 8, pp. 3277–3284, 2011. DOI: 10.1021/je200033b.

[68] M. L. Michelsen and J. Mollerup, *Thermodynamic Modelling: Fundamentals and Computational Aspects*. Tie-Line Publications, 2004, 330 s.

[69] E. Jones, T. Oliphant, P. Peterson, *et al.*, *Scipy: Open source scientific tools for Python*, 2001–.

[70] A. Bailey, *Melting and Solidification of Fats*, ser. Fats and oils. Interscience Publishers, 1950.

[71] M. Peters and K. Timmerhaus, *Plant design and economics for chemical engineers*. McGraw-Hill, 1991.

[72] Y. Aktas, *Superstructure generation of hybrid solvent crystallization and distillation processes for fatty acid separation process*, 2018.

[73] L. H. Wesdorp, J. A. van Meeteren, S. De Jong, R. V. D. Giessen, P. Overbosch, P. A. M. Grootscholten, M. Struik, E. Royers, and A. Don, "Liquid-multiple solid phase equilibria in fats: Theory and experiments," in *Fat Crystal Networks*, CRC Press. DOI: 10.1201/9781420030549.

[74] J. Hjorth, “Mathematical modeling of vegetable oil crystallization,” English, PhD thesis, 2014.

CHAPTER 6

Superstructure Optimisation with Surrogate Models

The formulation of an optimisation problem in form of a general disjunctive program (GDP) has been researched extensively [1–3]. GDPs for superstructure optimisation allows the identification of optimal process flowsheet structure and point of operation given a set of possible alternatives. The superstructure optimisation incorporates selection and interconnection of each unit operation in form of disjunctions and with the objective to maximise profit or to minimise total cost. In this paper we highlight the surrogate building step of the methodology with a rigorous counter-current spray column model and a continuous stirred tank reactor (CSTR). We assess the performance of different surrogate modelling methods such as multivariate regression splines, polynomial chaos expansion and Gaussian process regression in respect to the coefficient of determination (R^2), the root- and mean squared error (RMSE, MSE). The GDP is solved by transforming it to a MINLP via convex-hull transformation and then solving the problem with a suitable solver.

6.1 Methodology for surrogate-based superstructure optimisation

The steps of the methodology are presented as follows:

1. **Sampling of the design space**: The independent variables (i.e. design degrees of freedom) of the process under research are identified and the boundaries specified. Different experimental designs exist to cover the

domain of variation such as Monte Carlo samples obtained with random generators, Latin hypercube samples or quasi-random (low-discrepancy) sequences, e.g. Sobol, Hammersley or Halton sequences. A hypercube is generated and used for the next step to perform Monte Carlo simulation. Due to optimization (step 6) the bounds of the independent variables will get iteratively updated to ensure feasible solutions.

2. **Monte Carlo simulation of rigorous process**: The model is evaluated for each sample and the matrix of observations (model outputs) is stored for the surrogate modelling step.

3. **Build surrogate models**: To build a surrogate model the sampling hypercube and matrix of observations obtained from step 1 and 2 are used to apply methods such as multivariate regression splines, polynomial chaos expansion or Gaussian process regression.

4. **Surrogate-based superstructure generation**: A superstructure formulation of the interconnected surrogates and all possible combinations of process structures with operating set points are generated through enumeration or transforming a disjunctive program into a mixed integer non-linear program (MINLP).

5. **Multi-criteria economic & sustainability evaluation**: The different process structures are evaluated in terms of capital, operating and total annual costs and total environmental impact is assessed to take the sustainability of a process into account.

6. **Optimisation**: In this step the optimisation problem is solved where the economic and environmental objective function is formulated as the minimisation of total annual cost and environmental impact subject to boundary conditions.

6.2 Surrogate modelling

Different methods of generating surrogate models have been applied in literature. Fernandes [4] used neural networks to model the Fischer–Tropsch process. Neural networks were also used to generate a surrogate model for each unit operation in the process superstructure by Henao and Maravelias [5] and to optimise the problem. Cremaschi [6] combined second-order surrogate models and a steepest descent routine iteratively in a post-combustion CO2

removal process and neural networks for several other processes. Carpio et al. [7] present surrogate modelling with Gaussian processes (Kriging) combined with the probability of improvement method for constrained global optimisation. They apply their methodology also to three chemical engineering problems. Schweidtmann and Mitsos [8] perform optimisation with neural networks in respect to a fermentation process , a compressor plant and a cumene process. It is important research to perform optimisation with surrogate models to treat existing processes as black boxes, but it has to be questioned if steady-state models should be substituted by a surrogate to optimise only for the operating point. Generating a surrogate will increase the uncertainty of the estimated model output to some extent and in respect to neural networks drop outs have to be implemented to retrieve the confidence bounds. In case of Gaussian processes (Kriging) the uncertainties are provided and one could argue that surrogate methods allow to estimate the uncertainty of a process output without using a full Monte Carlo approach as discussed in the molecular distillation process in this work (Section 5.1.3). However, surrogate models are popular to substitute black-box models which are either computationally expensive to evaluate, have noisy output behaviour or don't supply gradients [9]. Such models treated as black boxes can be for example computational fluid dynamics (CFD) models or dynamic process control models. Especially for multi-scale modelling applications surrogate modelling will play an important role in process systems engineering. Therefore we evaluate the possibility of generating surrogate functions from rigorous steady-state unit operation models and embed them in a superstructure optimisation problem. The superstructure is then solved to determine the best process route and operating point.

6.2.1 Comparison between surrogate modelling methods

The spray column model presented in section 5.1.1 serves as the rigorous model to be substituted by surrogate functions. The design space of the model is summarized in Table 6.1. The same boundaries are defined to generate identical hypercubes for all surrogate modelling methods. The sample size of the hypercube is 500 and populated with Sobol sequences. The model is solved with the input hypercube and the output (overall conversion) is stored for the surrogate modelling step. The sampling and observation data was split half wise for training and testing. The surrogate models were built with

Table 6.1: Design parameters with design space boundaries for the counter-current spray column model.

Parameter	Unit	Lower Bound	Upper Bound
Feed flow rate (x_1)	kmol/h	2000	3500
Solvent (water) to feed ratio (x_2)	-	1.4	1.9
Operating temperature (x_3)	K	473.15	533.15

the train set and the predictions were then compared with the test set. The coefficient of determination (R^2), the mean squared error (MSE) and the root mean squared error (RMSE) are given in Table 6.2 to compare the different surrogate modelling methods as well as the accuracy of mapping the rigorous (black-box) model to a surrogate model. Multivariate regression splines, polynomial chaos expansion (PCE) and Gaussian process regression (GPR) were applied. The theory covering GPR and PCE can be found in section 4.3 and 8.3. The surrogate modelling methods assume:

$$y_j = f_j(x_1, ..., x_p) + \epsilon_j \tag{6.1}$$

as the output to input relation of the system under research where f is the approximating function and ϵ the error.

For multivariate regression splines a training data set is fit to obtain $\hat{f}_j$ to each function f_j with the expansion of basis functions:

$$\hat{f}(x_1, ..., x_p) = \sum_{m=0}^{M} a_m B_m(x_1, ..., x_p) \tag{6.2}$$

Least-squares minimisation gives the coefficients a_m weighting the basis functions which take the form of a hinge function $h(x_i)$. The linear combination of a constant, a linear function and several hinge functions is iteratively calculated with a two-step forward/backward algorithm. In the first forward step the surrogate model is being built by adding hinge functions while maximizing the goodness of fit until a user-defined level of complexity has been reached. The second pruning and pass step will compare subsets of the potential hinge functions by means of a true cross-validation score (CVS) to prevent overfitting. The subset with the lowest CVS is chosen.

In respect to GPR Figure 6.1 shows how the cross-validation score, in this case R^2 , behaves as a function of the number of samples used for training the model. With a training set of more than 15 samples the fitted model will

Table 6.2: Comparison between different surrogate modelling methods based on the test set.

Surrogate modelling method	R^2	MSE	RMSE
Multivariate regression splines	96.2	0.1e-03	0.01
Polynomial chaos expansion	99.7	0.55e-05	0.00235
Gaussian process regression (Matérn kernel)	99.9	0.40e-05	0.002

give predictions with R^2 greater 95 %. The model described with multivariate regression splines is:

$$\begin{aligned} f(x_1, x_2, x_3) = {} & 0.1318 + 0.0016h(x_3 - 510.767) \\ & - 3.3521 \times 10^{-8} h(x_1 - 2.4716 \times 10^6) \\ & + 6.5607 \times 10^{-8} h(2.4716 \times 10^6 - x_1) - 0.0529 h x_2 \\ & + 0.0021h(x_3 - 487.095) - 0.0010h(487.095 - x_3) \end{aligned} \tag{6.3}$$

The surrogate obtained through polynomial chaos expansion is as follows:

$$\begin{aligned} f(x_1, x_2, x_3) = {} & 4.3580 - 0.0230x_3 + 0.5354x_2 + 3.110910^{-5} x_3^2 \\ & - 0.0015x_2x_3 + 4.040210^{-7} x_1 - 1.150610^{-9} x_1x_3 \\ & 0.0418x_2^2 + 2.878210^{-8} x_1x_2 + 1.598210^{-14} x_1^2 \end{aligned} \tag{6.4}$$

The surrogate functions obtained from PCE are chosen to be embedded in the superstructure formulation. The reason is the polynomial nature of these surrogates which the solvers can better handle than the terms obtained from Gaussian process regression. Figure 6.2 and 6.3 show the surrogate functions for the glycerol fraction at the bottom of the spray column and for the operating cost. The bottom fraction is described by a quadratic function and the operating cost by a bilinear term. The bottom fraction showed no sensitivity to the feed flowrate.

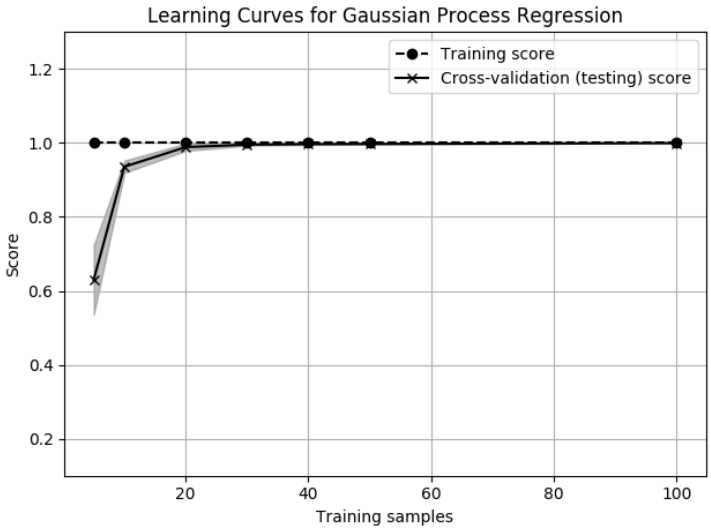


Figure 6.1: Comparison between the behaviour of training score and testing score as a function of the used training samples for Gaussian process regression of spray column model.

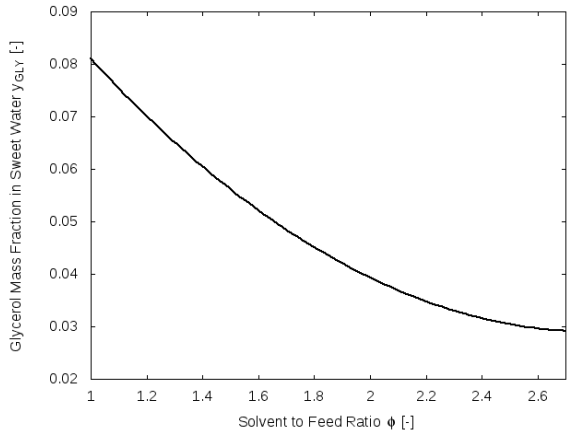


Figure 6.2: Plot of surrogate function for the glycerol mass fraction in respect to the sweet water product at the bottom of the spray column.

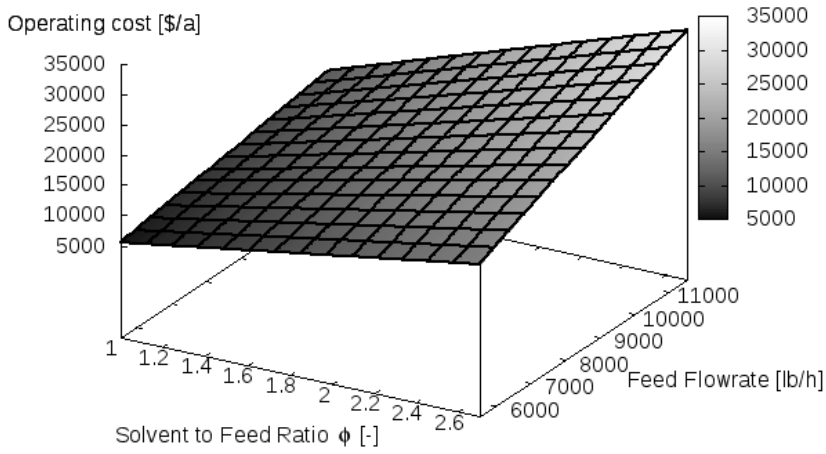


Figure 6.3: Surrogate function for operating cost of the spray column.

6.3 General disjunctive programming

A general disjunctive program can be formulated with the following equations [10]:

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & Z = \sum_{u \in U} c_u + f(x) \\ \text{subject to} \quad & h(x) = 0 \\ & g(x) \leq 0 \\ & \begin{bmatrix} Y_u \\ h_u(x) = 0 \\ r_u(x) \leq 0 \\ c_u = \gamma_u \end{bmatrix}, \forall u \in U^P \end{aligned} \tag{6.5}$$

$$\Omega(Y) = True, \quad Y_u \in \{True, False\}, \quad x \in X, \quad c_u \geq 0 \tag{6.6}$$

x and c_u are continuous variables where x encompasses e.g. flowrates, pressures and temperatures and c_u and γ_u represent costs and fixed charges. $h(x)$ and $g(x)$ are the global equality and inequality constraints of the whole problem formulation (e.g. mass balances) whereas $h_u(x)$ are the constraints subject to the individual unit. Y_u is a binary variable indicating either a permanent process unit in the set U^P or the activation or deactivation ($\neg Y_u$) of a conditional process unit in the set U^C. Conditional units are useful for process synthesis purposes [5] and each conditional unit can also include conditional finite elements (e.g. to describe trays in a distillation column or reactors in series) [11] in the set U^C_{FE}:

$$\begin{bmatrix} Y_u \\ h_u(x) = 0 \\ r_u(x) \leq 0 \\ c_u = \gamma_u + \sum c_{uv} \\ \begin{bmatrix} Y_{u,v} \\ h_u(x) = 0 \\ r_{uv} \leq 0 \\ c_{uv} = \gamma_{uv} \end{bmatrix} \vee \begin{bmatrix} \neg Y_{u,v} \\ B^{uv}x = 0 \\ c_{uv} = 0 \end{bmatrix}, \forall v \in U^C_{FE} \end{bmatrix} \vee \begin{bmatrix} \neg Y_u \\ B^u x = 0 \\ c_u = 0 \end{bmatrix}, \forall u \in U^C \tag{6.7}$$

Different methods exist to solve a GDP. One possibility is to tranform the GDP via convex-hull transformation and then solve the obtained MINLP with a nonlinear solver. Logic-based outer approximation is an other approach to solve a GDP which subdivides the GDP into NLP subproblems and into one MINLP master problem. The next two sections present the two methods more in depth.

6.4 Convex-hull transformation

Convex-hull transformation applies relaxations on the nonlinear disjunctions of the GDP to reformulate the problem as a MINLP. Hereby multipliers (λ_u) are introduced to relax the equality and inequality constraints in the disjunctions. This leads to the linearisation of the equality constraints and the introduction of bi-linearity in the inequality constraints. The non-convex inequality constraints have to be further treated by defining a new variable ($\nu^u = x\lambda_u$) the following can be defined with the convexity of λ_u:

$$\sum_{u \in U} x\lambda_u = x = \sum_{u \in U} \nu^u \tag{6.8}$$

$$\sum_{u \in U} c\lambda_u = c = \sum_{u \in U} \gamma_u \lambda_u \tag{6.9}$$

Rearranging the equations leads then to convex inequality constraints [12]:

$$\lambda_u r_u(\nu^u/\lambda_u) \leq 0 \tag{6.10}$$

and ν^u is bounded by:

$$0 \leq \nu^u \leq \lambda_u U_u \tag{6.11}$$

where U_u are the upper bounds.
Thus, the relaxation of the following disjunction:

$$\vee_{u \in U} \begin{bmatrix} Y_u \\ r_u(x) \leq 0 \\ c_u = \gamma_u \end{bmatrix} \tag{6.12}$$

would be [13]:

$$x = \sum_{u \in U} \nu^u, \quad c = \sum_{u \in U} \lambda_u \gamma_u \tag{6.13}$$

$$0 \leq \nu^u \leq \lambda_u U_u \tag{6.14}$$

$$\sum_{u} \lambda_u = 1, \quad 0 \leq \lambda_u \leq 1 \tag{6.15}$$

$$\lambda_u r_u(\nu^u/\lambda_u) \leq 0 \tag{6.16}$$

$$x, \nu^u, c \geq 0 \tag{6.17}$$

6.5 Logic-based outer approximation

Logic-based outer approximation (LOA) requires the solution of NLP sub-problems (only for the existing units) by fixing binary variables. The solution of these sub-problems provides an upper bound Z_U. The MILP master problems provide lower bounds and new values for the integer variables [14]. This allows solving reduced space NLP sub-problems rather than the full-space representations encountered when solving MINLPs, thereby improving robustness [15]. The NLP sub-problems and the MILP master problems are solved until the bounds of both problems converge. During iteration the nonlinear objective functions and constraints of the sub-problems are linearized at the solution points received from the NLP subproblems. The MILP master problem is set up and solved for the lower bound Z_L then.

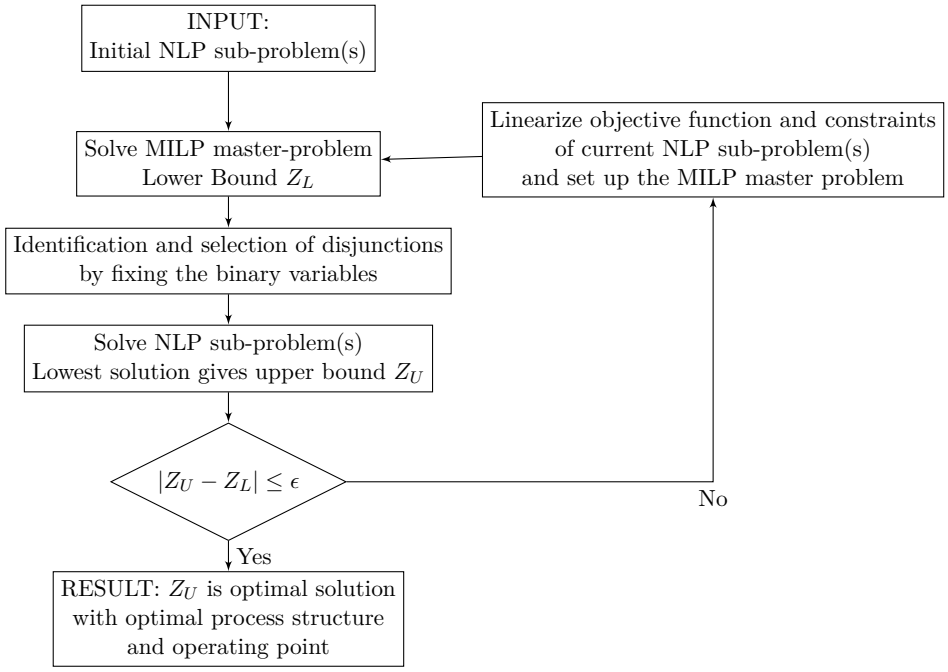


Figure 6.4: Logic-based outer approximation flow diagram.

6.6 Problem formulation for reactor networks

Given is a triglyceride feed which has to be processed by means of the hydrolysis reaction. An interesting question is if either the very efficient spray column with high conversion rates up to 99 per cent should be installed or a series of CSTRs which provide the same conversion of the raw material and investigate which configuration will give the lowest total annual cost. This task can be performed with describing the reactor superstructure (Figure 6.5) with a general disjunctive program (GDP) and include the conversion and cost functions as surrogate models obtained from the rigorous unit operation models. For this a set of reactants (triglyceride, water) and a set of products (fatty acid, glycerol) are specified. Additionally, either the feed rates along with purity requirements or the required production rates must be specified. A set of surrogate functions is given to provide the product composition of the bottom of the spray column as a function of solvent to feed ratio, whereas the surrogate function of the operating cost is a function of the feed flowrate and feed to solvent ratio. The objective is to minimise total annual cost and hereby find the optimal process structure and point of operation.

The objective is to minimise total annual cost which is composed of operating and capital costs for each unit ($u \in U$), the raw material ($r \in R$) costs and the revenue of the products ($p \in P$):

$$\min \text{ TAC} = \sum_{u \in U} C_u^{OP} + \frac{C_u^C}{3} + \sum_{r \in R} f_r * c_r - \sum_{p \in P} f_p * r_p$$

subject to the following global constraints:

- Fixed raw material compositions and component flowrates:

$$\begin{aligned} x_{1,4} &= 1.0 \\ x_{2,1} &= 1.0 \\ f_{1,1:3} &= 0 \\ f_{2,2:4} &= 0 \end{aligned} \tag{6.18}$$

- Material balance around units and mixers:

Figure 6.5: Superstructure representation of oleochemical hydrolysis process with spray column or CSTR.

$$\begin{aligned} f_3 + f_4 &= f_7 + f_8 \\ f_9 &= f_5 + f_6 \\ f_7 &= f_{7,3} + f_{7,4} \\ f_8 &= f_{8,1} + f_{8,2} \\ f_1 &= f_3 + f_6 \\ f_2 &= f_4 + f_5 \end{aligned} \tag{6.19}$$

- Closing equation:

$$f_{i,j} = x_{i,j} \sum_{j \in J} f_{i,j} \tag{6.20}$$

- Maximum product demand for fatty acid:

$$f_{8,2} + f_{9,2} \leq D \tag{6.21}$$

The disjunction for the spray column is:

$$\begin{bmatrix} Y_1 \\ f_4 = f_3 * \phi_1 \\ f_3 = f_1 \\ f_{3,j} = f_{1,j} \\ x_{3,j} = x_{1,j} \\ f_4 = f_2 \\ f_{4,j} = f_{2,j} \\ x_{4,j} = x_{2,j} \\ x_{7,GLY} = \hat{P}(\phi_1) \\ f_7 = f_{7,GLY} + f_{7,W} \\ f_{7,W} = f_{3,W} - 3 f_7 x_{7,GLY} \\ f_{8,TG} = f_{4,TG} - f_7 x_{7,GLY} \\ f_{8,FA} = 3 f_7 x_{7,GLY} \\ C_1^{OP} = \hat{P}(f_4, \phi_1) \\ C_1^C = 137220 \end{bmatrix} \vee \begin{bmatrix} \neg Y_1 \\ f_4 = 0 \\ f_3 = 0 \\ f_{3,j} = 0 \\ x_{3,j} = 0 \\ f_7 = 0 \\ f_{7,j} = 0 \\ f_8 = 0 \\ f_{8,j} = 0 \\ C_1^{OP} = 0 \\ C_1^C = 0 \end{bmatrix}, Y_1 \in U^C \tag{6.22}$$

It is assumed that the fatty acids are not miscible in the water phase. Streams are denoted with $f_{i,j}$ where i is the stream number and j the specifier for the species. The indices slightly change for the CSTR finite elements ($f_{i^*,i,j}$) where the first two indices are the identifiers for the stream and the third index is specifying the species. The disjunction for the CSTRs in series is:

$$
\left[\begin{array}{r}
Y_2 \\
f_5 = f_6\phi_2 \\
f_6 = \sum_i^{FE} f_i^{6^*} \\
f_{6,j} = f_{1,j} \\
x_{6,j} = x_{1,j} \\
f_5 = f_2 \\
f_{5,j} = f_{2,j} \\
x_{5,j} = x_{2,j} \\
f_{9,j} = f_{FE,j}^{5^*} \\
x_{9,j} = x_{FE,j}^{5^*} \\
C_2^{OP} = \sum_i^{FE} C_{2,i}^{OP} \\
C_2^{C} = \sum_i^{FE} C_{2,i}^{C} \\
\left[\begin{array}{r} Y_{2,1} \\ f_1^{5^*} = f_5 \\ x_{1,j}^{5^*} = x_{5,j} \\ f_1^{6^*} = f_1^{5^*}\phi_{2,1} \\ x_{1,j}^{6^*} = x_{6,j} \\ f_2^{5^*} = f_1^{5^*} + f_1^{6^*} \\ C_{2,1}^{OP} = \hat{P}_{OP,2}(\boldsymbol{x}) \\ C_{2,1}^{C} = 30000 \end{array}\right] \vee \left[\begin{array}{r} \neg Y_{2,1} \\ f_1^{5^*} = 0 \\ x_{1,j}^{5^*} = 0 \\ f_1^{6^*} = 0 \\ x_{1,j}^{6^*} = 0 \\ f_2^{5^*} = 0 \\ C_{2,1}^{OP} = 0 \\ C_{2,1}^{C} = 0 \end{array}\right] \\
\left[\begin{array}{r} Y_{2,v} \\ f_v^{5^*} = f_{v-1}^{5^*} + f_6 - \sum_i^{v-1} f_i^{6^*} \\ f_v^{6^*} = f_v^{5^*}\phi_{2,v} \\ x_{v,j}^{6^*} = x_{6,j} \\ x_{v,GLY}^{5^*} = \hat{P}(\phi_{2,v}) \\ f_{v,TG}^{5^*} = f_{v-1,TG}^{5^*} + f_{v-1,TG}^{6^*} - f_v^{5^*} x_{v,GLY}^{5^*} \\ f_{v,W}^{5^*} = f_{v-1,W}^{5^*} + f_{v-1,W}^{6^*} - 3 f_v^{5^*} x_{v,GLY}^{5^*} \\ C_{2,v}^{OP} = \hat{P}_{OP,2}(\boldsymbol{x}) \\ C_{2,v}^{C} = 30000 \end{array}\right] \vee \left[\begin{array}{r} \neg Y_{2,v} \\ f_v^{5^*} = 0 \\ f_v^{6^*} = 0 \\ x_{v,j}^{6^*} = 0 \\ x_{v,GLY}^{5^*} = 0 \\ f_{v,TG}^{5^*} = 0 \\ f_{v,W}^{5^*} = 0 \\ C_{2,v}^{OP} = 0 \\ C_{2,v}^{C} = 0 \end{array}\right], v \in [2, ..., FE]
\end{array}\right]
\vee \left[\begin{array}{r} \neg Y_{2,v} \\ f_5 = 0 \\ f_6 = 0 \\ f_{6,j} = 0 \\ x_{6,j} = 0 \\ f_5 = 0 \\ f_{5,j} = 0 \\ x_{5,j} = 0 \\ f_{9,j} = 0 \\ x_{9,j} = 0 \\ C_2^{OP} = 0 \\ C_2^{C} = 0 \end{array}\right], Y_2 \in U^C
\tag{6.23}
$$

Surrogate functions are introduced to the GDP for calculating the operational cost of the spray column and to map the reactor conversion to the glycerol mass fraction of the sweet water product. Superstructure optimisation was applied to assess the choice between a spray column and a CSTR

given to the global constraints. Surrogate models were obtained through the simulation data of the rigorous models and embedded in the GDP based superstructure. The goal therefore is to develop a design methodology which allows the generation of an optimised oleochemical process where the applying engineer has to define the raw material composition and product specification and the here proposed design toolbox generates the optimal process flowsheet structure and operating condition.

6.7 Optimisation

The GDP is formulated in Pyomo and transformed to a MINLP to then be solved. The operating cost for the spray columns is embedded in the problem formulation. The problem is solved by maximising the following objective function:

$$\begin{aligned} \text{Profit} &= \text{Revenue} - \text{Material Cost} - \text{Operating Cost} \\ &= f_{7,3} * p_2 + f_{8,2} * p_1 + f_{9,3} * p_2 + f_{9,2} * p_1 - f_1 * c_1 - f_2 * c_2 \\ &- C^{OP}_{\text{Spray Column}} - C^{OP}_{\text{CSTR}} \end{aligned} \tag{6.24}$$

where $S_{i,j}$ and S_i are the component streams and total streams. p_1 and p_2 are the selling price for the fatty acid and glycerol. c_1 and c_2 are the cost for the vegetable oil and high pressure steam. $C^{OP}_{\text{Spray Column}}$ and C^{OP}_{CSTR} are the operating costs of the spray column and CSTR respectively.

6.8 Results

The results in Table 6.3 show that the optimisation problem is solved correctly by selecting the spray column as the unit operation and finds the optimal point of operation for the feed to solvent ratio ($\frac{1}{\phi}$) at 0.58. The operating cost for this process structure is 1512401 \$/a. The streams connected to the CSTR are zero.

6.9 Discussion

The first steps for performing superstructure optimisation with surrogate models has been presented by formulating a general disjunctive program incorpo-

Table 6.3: Component stream table after superstructure optimisation.

Component stream	Flowrate value [mol/h]
$f_{1,1}$	0
$f_{1,2}$	0
$f_{1,3}$	0
$f_{1,4}$	4509.7
f_1	4509.7
$f_{2,1}$	1570.7
$f_{2,2}$	0
$f_{2,3}$	0
$f_{2,4}$	0
f_2	1570.7
$f_{3,1}$	0
$f_{3,2}$	0
$f_{3,3}$	0
$f_{3,4}$	4509.7
f_3	4509.7
$f_{4,1}$	1570.7
$f_{4,2}$	0
$f_{4,3}$	0
$f_{4,4}$	0
f_4	1570.7
$f_{7,1}$	0
$f_{7,2}$	0
$f_{7,3}$	1366.5
$f_{7,4}$	410.3
f_7	1776.8
$f_{8,1}$	204.2
$f_{8,2}$	4099.5
$f_{8,3}$	0
$f_{8,4}$	0
f_8	4303.7

rating a surrogate function from a rigorous spray column model. Next developments steps include a more flexible and faster way of dealing with Pyomo as an equation based modelling environment by providing an interface to set initial estimates and automate the calculations of the initial estimates of the unknown variables. Also the definition of ports for each unit will make the superstructure formulation more easy to extend to various unit operations. For this task the Pyomo Network package can be used. The formulation of a concrete model to an abstract model would be the final step to make superstructure formulations highly generic and to connect the routine to process simulators from which the surrogates of various unit operations or sub-processes can then be retrieved from.

Bibliography

[1] Q. Chen and I. E. Grossmann, "Effective generalized disjunctive programming models for modular process synthesis," *Industrial & Engineering Chemistry Research*, null, 2019. DOI: 10.1021/acs.iecr.8b04600.

[2] J. H. Ghouse, Q. Chen, M. A. Zamarripa, A. Lee, A. P. Burgard, I. E. Grossmann, and D. C. Miller, "A comparative study between gdp and nlp formulations for conceptual design of distillation columns," in *13th International Symposium on Process Systems Engineering (PSE 2018)*, ser. Computer Aided Chemical Engineering, 2018, pp. 865–870. DOI: 10.1016/B978-0-444-64241-7.50139-7.

[3] E. S. Rawlings, Q. Chen, I. E. Grossmann, and J. A. Caballero, "Kaibel column: Modeling, optimization, and conceptual design of multi-product dividing wall columns," *Computers & Chemical Engineering*, vol. 125, pp. 31–39, 2019. DOI: 10.1016/j.compchemeng.2019.03.006.

[4] F. A. N. Fernandes, "Optimization of fischer-tropsch synthesis using neural networks," *Chemical Engineering & Technology*, vol. 29, no. 4, pp. 449–453, 2006. DOI: 10.1002/ceat.200500310.

[5] C. A. Henao and C. T. Maravelias, "Surrogate-based superstructure optimization framework," *AIChE Journal*, vol. 57, no. 5, pp. 1216–1232, 2011. DOI: 10.1002/aic.12341.

[6] S. Cremaschi, "A perspective on process synthesis – challenges and prospects," *Computers & Chemical Engineering*, vol. 81, pp. 130–137, 2015. DOI: 10.1016/j.compchemeng.2015.05.007.

[7] R. R. Carpio, R. C. Giordano, and A. R. Secchi, "Enhanced surrogate assisted framework for constrained global optimization of expensive black-box functions," *Computers & Chemical Engineering*, vol. 118, pp. 91–102, 2018. DOI: 10.1016/j.compchemeng.2018.06.027.

[8] A. M. Schweidtmann and A. Mitsos, "Deterministic global optimization with artificial neural networks embedded," *Journal of Optimization Theory and Applications*, vol. 180, no. 3, pp. 925–948, March 2019. DOI: 10.1007/s10957-018-1396-0.

[9] N. Quirante, J. Javaloyes, and J. A. Caballero, "Rigorous design of distillation columns using surrogate models based on kriging interpolation," *AIChE Journal*, vol. 61, no. 7, pp. 2169–2187, 2015. DOI: 10.1002/aic.14798.

[10] Q. Chen and I. E. Grossmann, "Recent developments and challenges in optimization-based process synthesis," *Annual Review of Chemical and Biomolecular Engineering*, vol. 8, no. 1, pp. 249–283, 2017.

[11] B. Pahor, Z. Kravanja, and N. I. Bedenik, "Synthesis of reactor networks in overall process flowsheets within the multilevel minlp approach," *Computers & Chemical Engineering*, vol. 25, pp. 765–774, 2001.

[12] J. .-.-B. Hiriart-Urruty and C. Lemarechal, *Convex Analysis and Minimization Algorithms I*, eng. Springer Berlin Heidelberg, 1993, vol. 305, 430 s.

[13] S. Lee and I. E. Grossmann, "New algorithms for nonlinear generalized disjunctive programming," *Computers & Chemical Engineering*, vol. 24, no. 9, pp. 2125–2141, 2000. DOI: 10.1016/S0098-1354(00)00581-0.

[14] M. Türkay and I. E. Grossmann, "Logic-based minlp algorithms for the optimal synthesis of process networks," *Computers & Chemical Engineering*, vol. 20, no. 8, pp. 959–978, 1996.

[15] Q. Chen, E. Johnson, J. D. Siirola, and I. E. Grossmann, "Pyomo.GDP: Disjunctive models in python," *Proceedings of the 13th International Symposium on Process Systems Engineering*, vol. 1, pp. 1–2, 2018.

CHAPTER 7

Conclusion and perspectives

This thesis project presented a comprehensive multi-scale framework covering property prediction, process design, flowsheeting and optimisation. The methodologies and tools were applied to the oleochemical domain. The results show that industrial relevant processes with detailed models can be generated combined with advanced analysis and optimisation methods.

The proposed data-driven property prediction methodology with Gaussian process regression (GPR) is a promising method to retrieve stochastic models from experimental data. This will allow users in the future to train prediction models with molecular descriptors and the experimental values without the necessity to provide initial estimates of the group contribution values and reducing the empiric nature of group contribution methods. The structural information of the experimental component is automatically generated for training the Gaussian process. Markov Chain Monte Carlo (MCMC) algorithms could be used to apply full Bayesian inference by defining prior distributions over all unknowns. These unknowns would be the length scale hyperparameter, each molecular descriptor and the noise variance in respect to the example of this work.

The modelling procedure presented proved to be highly flexible and allowed to embed and connect the models of processes with a wide range of tools. Thus, the established framework marks the foundation for the development of a highly flexible software prototype. From the practical standpoint the database, data structures and data pipeline have to be further improved, standardised and adapted to the CAPE-OPEN Binary Interop Architecture (COBIA).

Three oleochemical processes have been presented in this thesis to study different aspects of important analysis and optimisation methods. The spray column model showed that it is possible and useful to make use of the Fortran90 programming language to embed the model in a process simulator but

also wrap it in a higher-level language. This allowed to perform parameter estimation and optimisation with differential evolution and Sobol sensitivity analysis which wouldn't have been possible by only using a process simulator. The spray column model was successfully validated with the only available experimental data set for a spray column from the 60s. The parameter estimation and optimisation routine makes this contribution valuable for industry. Combined with a more sophisticated kinetic model and knowledge from CFD results, the finite volume model will allow to make use of the full simulation capabilities it holds and provide accurate design specifications and operating conditions.

The molecular distillation model showed that uncertainty and sensitivity analysis can be performed with a process simulator. This is an important aspect for the industrial practice to build processes with a feasible and robust design with the desired characteristics of the product. Since the full Monte Carlo approach to sensitivity analysis showed that the time of evaluating all sampled data points is infeasible, polynomial chaos expansion based sensitivity analysis was able to reduce the needed number of evaluations drastically and also could handle a number of input parameters up to 15 in the presented example. It is highly recommendable that commercial process simulators implement the proposed methodology. Further studies should be made in respect to different machine-learning based sensitivity analysis methods. Also the assessment of the limited number of input parameters (curse of dimensionality) of different machine-learning methods should be performed.

The implemented crystallisation process model for separating saturated and unsaturated fatty acids agreed with experimental results. Different evaluated process structures indicated how crystallisation units should be combined with distillation columns. The methods which have been applied to the spray column and molecular distillation study should also be applied to the crystallisation process to identify the optimal operating temperature of each crystallisation unit and reduce the needed amount of solvent. Further, solvent design may enhance the process considerably. The polymorphic behaviour of the oil crystals hasn't been taken into account in this work, therefore the extension of the model to the capability of predicting the crystal forms is advisable.

The use of polynomial surrogate functions in superstructure optimisation has been successfully demonstrated. A formulation of the decision problem between two reactors has been solved correctly. The work on superstructure optimisation needs to be extended and made more generic. This means that the working example of the concrete model has to be formulated as an abstract model in Pyomo and a more generic port connection system needs to

be implemented which the Pyomo Network is capable of. This will allow the user to perform superstructure optimisation with a set of process tasks in which several different unit operations can be chosen from. All the necessary information for performing the optimisation is stored in surrogate functions obtained from the rigorous process models. Therefore the abstract model needs to provide an interface to retrieve and store the surrogate models. The future perspective would be to have a connection with a process simulator which performs the multiple evaluations of all unit operations or sub-processes. The obtained surrogates are then automatically sent to the superstructure optimisation to evaluate the optimal process structure and point of operation.

CHAPTER 8

Appendix

8.1 Wilson parameters of fatty acids and acetone

The energy parameters are taken from Wale [2] who obtained the parameters through regression of the data from Barley [] and Brown.

Table 8.1: Wilson energy parameters for binary mixtures of fatty acids.

i-j	λ_{ij} [J/mol]	λ_{ji} [J/mol]
Stearic-Palmitic	6595.9	-3332.9
Stearic-Oleic	-260.0	869.62
Stearic-Linoleic	-1465.8	2744.7
Palmitic-Oleic	5141.9	-2496.5
Palmitic-Linoleic	2001.5	-1269.7
Oleic-Linoleic	-2788.1	6041.8

Table 8.2: Wilson energy parameters for binary mixtures of fatty acids and acetone.

i-j	λ_{ij} [J/mol]	λ_{ji} [J/mol]
Stearic-Acetone	-843.7	4930.3
Palmitic-Acetone	2584.9	1872.0
Oleic-Acetone	6127.3	-125.9
Linoleic-Acetone	6222.7	-3334.4

8.2 Full Monte Carlo and PCE based sensitivity analysis (Sobol method)

Variance-based sensitivity analysis is described in the following. The analysis of variance (ANOVA-HDMR) decomposition gives the relation:

$$\sum_i S_i + \sum_i \sum_{j>i} S_{ij} + \sum_i \sum_{j>i} \sum_{l>j} S_{ijl} + ... + S_{123...M} = 1 \tag{8.1}$$

For a model with 3 input variables under study, the total effect ST_1 of input variable x_1 is analytically defined as the sum of the first order effect and higher order interactions:

$$ST_1 = S_1 + S_{12} + S_{13} + S_{123} = S1_1 + S_{12} + S_{13} + S_{123} \tag{8.2}$$

The first-order sensitivity index is a measure for the contribution of the i-th input parameter to the variance of the output $V(Y)$. We can describe the index in probabilistic form as:

$$S1_i = \frac{V[E(y|x_i)]}{V(y)} \tag{8.3}$$

where $V[E(y|x_i)]$ is the conditional variance and for $S1_i$ the following condition holds:

$$0 \leq S1_i \leq 1 \quad \wedge \quad \sum S1_i \leq 1 \tag{8.4}$$

$S1_1 = 1$ would imply that all variance of y is affected by x_1 and fixing it also determines y.

We generate two independent sampling matrices A and B where the row index denotes the simulation number and the column index references the input factor. $A_B^{(i)}$ is a matrix which is copied from A except the i-th column which is copied from B. With these matrices we can calculate the first-order index [3]:

$$S1_i = \frac{\frac{1}{N}\sum_{j=1}^{N} y(B)_j (y(A_B^{(i)})_j - y(A)_j)}{V(y)} \tag{8.5}$$

The total effect index describes the amount of interactions between the input factors. In probabilistic form, the definition of the index is:

$$ST_i = \frac{E[V(y|x_{\sim i})]}{V(y)} \tag{8.6}$$

$2^M - 1$ total sensitivity indices exist for a given input-output model with M number of inputs. We calculate the first order and total effect indices with A, B and $A_B^{(i)}$ [4] [5, 6]:

$$ST_i = \frac{\frac{1}{2N}\sum_{j=1}^{N}(y(A)_j - y(A_B^{(i)})_j)^2}{V(y)} \tag{8.7}$$

where for both indices $V(y)$ is the variance of the model output [3, 7]:

$$V(y) = \frac{1}{N}\sum_{j=1}^{N}(y(A)_j)^2 - (\frac{1}{N}\sum_{j=1}^{N}y(A)_j)^2 \tag{8.8}$$

2N simulations have to be performed to obtain the output y from the matrices A and B. For computing $y(A_B^{(i)})$, M times N simulations are needed. The overall model evaluations would be N(M+2) [5]. To also calculate the interaction effects the number of needed model evaluations would be N(2M+2) [6].

The other approach we want to discuss in this work is the retrieval of the sensitivity indices from a surrogate (response surface) model which we generate from the input-output data via polynomial chaos expansion (PCE). A PCE can be expressed by the following equation [8, 9]:

$$\hat{y}(\boldsymbol{\xi}) = \sum_{i\in\mathbb{N}^M}^{\infty} c_i\Phi_i(\boldsymbol{\xi}) \tag{8.9}$$

A truncation scheme is applied limiting the expansion order to a degree of p to make the method computational feasible. In this work we generate the polynomial basis with the Wiener-Askey scheme [10] which defines the type of univariate polynomial to select for a given standard distribution of ξ_i. If the input parameters x_i can not be interpreted via the Wiener-Askey scheme, copula theory [11] can be applied or a isoprobalistic transformation has to be performed via Rosenblatt transformation [9]. To put it simply, we derive from the independent input random vector $\boldsymbol{x}$ a set of realizations which we compute the basic random vector $\boldsymbol{\xi}$ from and for which we have a look up table to choose the univariate polynomials (e.g. Hermite polynomials for normal distributions and Legendre polynomials for uniform distributions). We can construct the M-variate orthogonal polynomial basis (multivariate polynomials $\Phi_i(\boldsymbol{\xi})$) as a tensor product of the univariate orthonormal polynomials $\phi_{\alpha_j}^{(j)}(\xi_i)$ [12]:

$$\Phi_i(\boldsymbol{\xi}) = \prod_{j=1}^{M} \phi_{\alpha_j}^{(j)}(\xi_i) \tag{8.10}$$

For high-dimensional problems (M $\geq$ 10) an adaptive sparse PCE can be generated to compensate the curse of dimensionality [13, 14]. These adaptive algorithms select a subset of the polynomial basis during the truncation step and then evaluate if the regression method used, e.g. least angle regression (LARS) [15], gives the minimum leave-one-out-error. The coefficients are then estimated with least squares regression to minimize the L_2-norm.

The mean μ and variance σ^2 of the model output are retrieved from the coefficients of the individual basis functions:

$$\mu = c_0 \tag{8.11}$$

$$\sigma^2 = \sum_{i \neq 0} c_i^2 \Phi_i^2 = \sum_{i \neq 0} c_i^2 \tag{8.12}$$

The sensitivity indices are also calculated with the coefficients and the uni-/multivariate polynomials from the PCE [16]:

$$S1_i = \frac{\sum_{\alpha \in \mathrm{Im}_i} c_\alpha^2 \phi_\alpha^2}{\sum_{k=1}^{p} c_k^2 \Phi_k^2} \tag{8.13}$$

$$ST_i = \frac{\sum_{\alpha \in \jmath_i} c_\alpha^2 \phi_\alpha^2}{\sum_{k=1}^{p} c_k^2 \Phi_k^2} \tag{8.14}$$

Note the difference between the calculation of $S1_i$ and ST_i, namely the sets of indices Im_i and $\jmath_i$:

$$\mathrm{Im}_i = \left\{ \begin{array}{lll} & \alpha_k \quad \mathrm{k} = 1, ..., \mathrm{n} & \mathrm{k} \in (i_1, ..., i_s) \\ \boldsymbol{\alpha}\text{:} & & \\ & \alpha_k \quad \mathrm{k} = 1, ..., \mathrm{n} & \mathrm{k} \notin (i_1, ..., i_s) \end{array} \right\} \tag{8.15}$$

$$\jmath_{i_1, i_2, ..., i_s} = \{\boldsymbol{\alpha} : \alpha_k > 0 \quad \forall k = 1, ..., n \quad k \in (i_1, i_2, ..., i_s)\} \tag{8.16}$$

The integer set of the tuple $\boldsymbol{\alpha}$ represents each term in the expansion, being the tensor product of univariate orthogonal polynomials , and is defined as follows:

$$\boldsymbol{\alpha} = (\alpha_1, ..., \alpha_n); \quad \alpha_i \geq 0; \quad \sum_{i=1}^{M} \alpha_i \leq P \tag{8.17}$$

See also Table 9 for more clarification on how the multi-index maps to the basis functions.

The following relation holds for the number of polynomials P of an expansion with degree p and the input vector dimensionality of M:

$$P = \begin{pmatrix} M+p \\ p \end{pmatrix} = \frac{(M+p)!}{M!p!} \tag{8.18}$$

For example, the PCE of a two dimensional input vector and a polynomial degree of three leads to the number of polynomials of P=10. As mentioned before, Table 9 lists the basis functions and the corresponding multi-index for this case.

Table 8.3: Mapping from multiple indices to single index via multi-index for the two dimensional case and a polynomial degree of three.

m	p	Basis functions	Multi-index
0	0	$\Phi_0(\xi_1,\xi_2) = 1$	$\alpha_0 = (0,0)$
1	1	$\Phi_1(\xi_1,\xi_2) = \phi_1(\xi_1)$	$\alpha_1 = (1,0)$
2	1	$\Phi_2(\xi_1,\xi_2) = \phi_1(\xi_2)$	$\alpha_2 = (0,1)$
3	2	$\Phi_3(\xi_1,\xi_2) = \phi_2(\xi_1)$	$\alpha_3 = (2,0)$
4	2	$\Phi_4(\xi_1,\xi_2) = \phi_1(\xi_1)\phi_1(\xi_2)$	$\alpha_4 = (1,1)$
5	2	$\Phi_5(\xi_1,\xi_2) = \phi_2(\xi_2)$	$\alpha_5 = (0,2)$
6	3	$\Phi_6(\xi_1,\xi_2) = \phi_3(\xi_1)$	$\alpha_6 = (3,0)$
7	2	$\Phi_7(\xi_1,\xi_2) = \phi_2(\xi_1)\phi_1(\xi_2)$	$\alpha_7 = (2,1)$
8	2	$\Phi_8(\xi_1,\xi_2) = \phi_1(\xi_1)\phi_2(\xi_2)$	$\alpha_8 = (1,2)$
9	3	$\Phi_9(\xi_1,\xi_2) = \phi_3(\xi_2)$	$\alpha_9 = (0,3)$

Table 8.4: Interaction sensitivity indices describing the effect of parameters between each other.

Component	$T_{C,Tripalmitin}$	$P_{C,Tripalmitin}$	$\omega_{Tripalmitin}$	$T_{C,Triolein}$	$P_{C,Triolein}$	$\omega_{Triolein}$	$T_{C,OleicAcid}$	$P_{C,OleicAcid}$	$\omega_{OleicAcid}$	$T_{C,Tocopherol}$	$P_{C,Tocopherol}$	$\omega_{Tocopherol}$	$T_{C,Carotene}$	$P_{C,Carotene}$	$\omega_{Carotene}$
$T_{C,Tripalmitin}$	0	-0.001012 ± 0.010009	-0.002872 ± 0.008885	-0.000499 ± 0.006962	0.000516 ± 0.010158	-0.000480 ± 0.006792	0.000292 ± 0.007102	-0.000061 ± 0.007324	-0.000085 ± 0.007295	-0.000092 ± 0.007284	-0.000084 ± 0.007292	-0.000085 ± 0.007295	0.006502 ± 0.012055	0.002999 ± 0.008984	0.000500 ± 0.005875
$P_{C,Tripalmitin}$		0	-0.035791 ± 0.065794	-0.027521 ± 0.069174	-0.011838 ± 0.073258	-0.033325 ± 0.065804	-0.025538 ± 0.067758	-0.027382 ± 0.067525	-0.027476 ± 0.067439	-0.027459 ± 0.067440	-0.027467 ± 0.067434	-0.027476 ± 0.067439	-0.005904 ± 0.079848	-0.018482 ± 0.066615	-0.018463 ± 0.070776
$\omega_{Tripalmitin}$			0	0.003808 ± 0.014801	0.007271 ± 0.018744	0.004190 ± 0.015002	0.002263 ± 0.016664	0.003215 ± 0.015490	0.003891 ± 0.014805	0.003854 ± 0.014812	0.003890 ± 0.014807	0.003891 ± 0.014805	0.002799 ± 0.015344	0.002756 ± 0.017293	0.003664 ± 0.014012
$T_{C,Triolein}$				0	-0.000379 ± 0.013071	-0.000155 ± 0.011425	-0.002102 ± 0.010245	-0.002205 ± 0.009816	-0.002412 ± 0.009764	-0.002423 ± 0.009763	-0.002413 ± 0.009764	-0.002412 ± 0.009764	-0.008074 ± 0.013898	-0.000965 ± 0.009608	-0.002226 ± 0.010997
$P_{C,Triolein}$					0	-0.040702 ± 0.067030	-0.043563 ± 0.067623	-0.043247 ± 0.066265	-0.042160 ± 0.065215	-0.042180 ± 0.065216	-0.042148 ± 0.065215	-0.042160 ± 0.065215	-0.018967 ± 0.065284	-0.029145 ± 0.068216	-0.028444 ± 0.065761
$\omega_{Triolein}$						0	0.005894 ± 0.007906	0.005757 ± 0.007741	0.005517 ± 0.007610	0.005518 ± 0.007601	0.005516 ± 0.007608	0.005517 ± 0.007610	0.006847 ± 0.010653	0.006268 ± 0.009754	0.006596 ± 0.008606
$T_{C,OleicAcid}$							0	-0.004758 ± 0.012633	-0.004002 ± 0.011637	-0.004019 ± 0.011659	-0.004002 ± 0.011640	-0.004002 ± 0.011637	0.004438 ± 0.011010	-0.006574 ± 0.013759	-0.003803 ± 0.009392
$P_{C,OleicAcid}$								0	-0.002054 ± 0.002137	-0.002051 ± 0.002136	-0.002054 ± 0.002137	-0.002054 ± 0.002137	-0.002182 ± 0.002279	-0.002023 ± 0.002185	-0.001944 ± 0.002147
$\omega_{OleicAcid}$									0	0.000080 ± 0.000129	0.000080 ± 0.000129	0.000080 ± 0.000129	0.000080 ± 0.000129	0.000080 ± 0.000129	0.000080 ± 0.000129
$T_{C,Tocopherol}$										0	0.000030 ± 0.000232	0.000030 ± 0.000231	-0.000231 ± 0.000593	0.000006 ± 0.000248	0.000002 ± 0.000231
$P_{C,Tocopherol}$											0	0.000078 ± 0.000143	0.000062 ± 0.000174	0.000073 ± 0.000144	0.000094 ± 0.000150
$\omega_{Tocopherol}$												0	0.000080 ± 0.000129	0.000079 ± 0.000129	0.000079 ± 0.000129
$T_{C,Carotene}$													0	0.056279 ± 0.114991	0.067831 ± 0.124408
$P_{C,Carotene}$														0	0.002970 ± 0.039410
$\omega_{Carotene}$															0

8.3 Python-PRO/II interface

The following Python script establishes a COM server connection with the process simulator PRO/II. The sampling hypercube has to be generated by the user and stored as a numpy data object depicted in the code as X.npy. For storing and loading .mat files the scipy.io class is recommended. The Python packages SALib and Chaospy were used to conduct sensitivity analysis and to generate the polynomial chaos expansion and to post-process the SA. Better results (January 2019) were obtained with the UQLab Matlab package for the polynomial chaos expansion since this package has the adaptive algorithm implemented which is mentioned in the previous section.

```
import numpy as np
import win32com.client as win32
import os
import time


#Connect via COM interface
def COMconnect(db_path):
    pro2   = "Nothing"
    pro2db = "Nothing"
    pro2 = win32.Dispatch("SimSciDbs.Database.101")

    pro2.Initialize()

    pro2.SetOption("showInternalObjects", "1")
    pro2.SetOption("DoublePrecision", "1")

    pro2.Import(os.path.splitext(db_path)[0]+'.inp')

    pro2db = pro2.OpenDatabase(db_path)

    #Get a security license (for better performance)
    pro2.GetSecuritySeat(2)

    return pro2, pro2db


#Disconnect COM interface
def COMdisconnect(pro2, pro2db):
    #Release the security license
    pro2.ReleaseSecuritySeat()

    #Shut down the connection to the COM server
    pro2db = "Nothing"
```

```
    pro2   = "Nothing"

    return pro2, pro2db


#Evaluate flowsheet with new input vector/matrix
def evaluate(db_path, NOC, X):

    pro2, pro2db = COMconnect(db_path)
    pro2check    = pro2db.CheckData
    pro2db       = "Nothing"
    pro2run      = pro2.RunCalcs(db_path)
    pro2db       = pro2.OpenDatabase(db_path)


    #Provide array in which the results are stored
    Stream_Product_Bottom_arr = np.zeros(shape=(len(X),1))

    Component_Name_List = [
      "TRIPALM",
      "TRIOLEIN",
      "OLEIC",
      "A-TOCOPH",
      "B-CAROTN"]

    #Check data and run simulations
    for sim_id, sim_id_ in enumerate(Stream_Product_Bottom_arr):

        print("Simulation Number:", sim_id)

        prop_idx = 0 # Property index

        for comp_id, comp_name in enumerate(Component_Name_List):

            print("Component Name:", comp_name)

            Tc_value    = X[sim_id, prop_idx]

            #Change critical temperature in database of component i
            Comp_i = pro2db.ActivateObject("CompIn", comp_name)
            Comp_i.PutAttribute(Tc_value, "CritTempIn")
            Comp_i.Commit(True)
            Comp_i = "Nothing"

            prop_idx += 1

            Pc_value = X[sim_id, prop_idx]
```

```
82
83              #Change critical pressure in database of component i
84              Comp_i = pro2db.ActivateObject("CompIn", comp_name)
85              Comp_i.PutAttribute(Pc_value, "CritPressIn")
86              Comp_i.Commit(True)
87              Comp_i = "Nothing"
88
89              prop_idx += 1
90
91              omega_value = X[sim_id, prop_idx]
92
93              #Change acentric factor in database of component i
94              Comp_i = pro2db.ActivateObject("CompIn", comp_name)
95              Comp_i.PutAttribute(omega_value, "AcenFactorIn")
96              Comp_i.Commit(True)
97              Comp_i = "Nothing"
98
99              prop_idx += 1
100
101             print("Property values perturbed to...
102             T_c:", Tc_value,
103             "P_c:", Pc_value,
104             "omega:", omega_value)
105
106         pro2check   = pro2db.CheckData
107         pro2check   = pro2db.DbsSaveDb
108         print("Database saved:", pro2check)
109         pro2db = "Nothing"
110
111         print("Return Value of CheckData
112         (0=no error, 1=error):", pro2check)
113
114         #Print error messages
115         nMsg = pro2.MsgCount
116         if nMsg > 0:
117             print(nMsg)
118             for i in range(0, nMsg):
119                 print("Error message:", pro2.MsgText(i))
120
121         #Run simulation
122         pro2run = pro2.RunCalcs(db_path)
123         print("Return Value of RunCalc:", pro2run)
124         Solutions[sim_id] = pro2run
125
126         nMsg = pro2.MsgCount
127         if nMsg > 0:
128             print(nMsg)
```

```
            for i in range(0, nMsg):
                print("Error message:", pro2.MsgText(i))

        #OUTPUT
        pro2db = pro2.OpenDatabase(db_path)

        pro2db.CalculateStreamProps("REFINED")

        #beta-carotene fraction at bottom
        Stream_Product_Bottom = pro2db.ActivateObject("Stream", "
            REFINED")
        Stream_Product_Bottom_arr[sim_id] =
        Stream_Product_Bottom.GetAttribute("LiquidComposition",4)

        print("Stream_Product_Bottom:", Stream_Product_Bottom)
        print("Stream Product Bottom Total Composition Value:",
        Stream_Product_Bottom_arr[sim_id])

        Stream_Product_Bottom = "Deactivate"
        Stream_Product_Bottom = "Nothing"

        np.save('Y_temp.npy', Stream_Product_Bottom_arr)
    #----End of simulation loop----

    pro2, pro2db = COMdisconnect(pro2, pro2db)

    return Stream_Product_Bottom_arr

if __name__ == "__main__":

    np.set_printoptions(threshold=np.inf)

    #Path to the ProII file
    db_path = "C:\\Users\\foo\\bar.prz"

    #Number of components
    NOC = 5

    X = np.load("X.npy")
    print("Size of loaded sample hypercube:", X.shape)

    start_time = time.time()

    Y = evaluate(db_path, NOC, X)

    #Print how much time the evaluations took
    print("--- \%s seconds ---" \% (time.time() - start_time))
```

```
175
176     #Save results
177     np.save('Y.npy', Y)
```

.

8.4 User-added unit operations and subroutines in PRO/II

The user can declare data structures with so called derived types in Fortran combined with the possibility to perform high-performance computing by implementing algorithms in parallel. This allows to perform computational expensive model simulations. Such a model can then be wrapped by the Python programming language to apply different tools and transfer the generated data to the next simulation tool such as a process simulator in order to simulate a process flowsheet. Further, commercial process simulators such as Aspen and PRO/II so far only support user-added unit operations as Fortran models which can be directly used in the graphical user interface of the simulator. The Fortran 2018 standard (N2162) has been published by the International Organization for Standardization (ISO) and now also supports parallel programming with coarrays without any additional message passing interface such as OpenMPI.

After a Fortran model has been developed, the code has to be adapted to the specific data structures of PRO/II. A good modelling practice would be to identify how the code should be implemented as a pure Fortran model to make the conversion to a Fortran-PRO/II implementation as easy as possible. Example Fortran-PRO/II files for user-added unit operations (UAOP) or subroutines (UAS) are found in the folder path C:/Program Files/SIMSCI/PROII/User/UAS/. Configuration files for the UAOP or UAS (.xml, *.dat and p2uasreg.ini) are located or have to be created in the ../ProII/System folder.

The model has to be compiled to a shared library in form of a dynamic linked library (.dll) so that the model can be accessed with PRO/II. For this only the Microsoft Virtual Studio development environment can be used with the Intel Fortran compiler. It is not possible to compile the Fortran code with an open-source compiler such as gfortran as stated by the PRO/II manual. But it could be tried to use MinGW to not be dependent on the Intel Fortran

compiler. More information can be found in the user-added unit operation manual of PRO/II.

Bibliography

[1] C. A. Diaz-Tovar, R. Gani, and B. Sarup, "Computer-aided modeling of lipid processing technology," English, PhD thesis, Jul. 2011.

[2] S. N. Wale, "Separation of fatty acids by extractive crystallization," English, PhD thesis, 1995.

[3] A. Saltelli, M. Ratto, T. Andres, F. Campolongo, J. Cariboni, D. Gatelli, M. Saisana, and S. Tarantola, *Global Sensitivity Analysis: The Primer*. Wiley, 2008.

[4] M. Jansen, "Analysis of variance designs for model output," *Computer Physics Communication*, vol. 117, no. 1-2, pp. 35–43, 1999. DOI: 10.1016/S0010-4655(98)00154-4.

[5] A. Saltelli, P. Annoni, I. Azzini, F. Campolongo, M. Ratto, and S. Tarantola, "Variance based sensitivity analysis of model output. design and estimator for the total sensitivity index," *Computer Physics Communications*, vol. 181, no. 2, pp. 259–270, 2010. DOI: 10.1016/j.cpc.2009.09.018.

[6] J. Herman and W. Usher, "Salib: An open-source python library for sensitivity analysis," *Journal of Open Source Software*, vol. 2, no. 9, 2017.

[7] A. Zubov and G. Sin, "Multiscale modeling of poly(lactic acid) production: From reaction conditions to theology of polymer melt," *Chemical Engineering Journal*, vol. 336, pp. 361–375, 2018. DOI: 10.1016/j.cej.2017.12.033.

[8] B. Sudret, "Global sensitivity analysis using polynomial chaos expansions," *Reliability Engineering and System Safety*, vol. 93, no. 7, pp. 964–979, 2008. DOI: 10.1016/j.ress.2007.04.002.

[9] J. Feinberg and H. P. Langtangen, "Chaospy: An open source tool for designing methods of uncertainty quantification," *Journal of Computational Science*, vol. 11, pp. 46–57, 2015. DOI: 10.1016/j.jocs.2015.08.008.

[10] D. Xiu and G. Karniadakis, "The wiener–askey polynomial chaos for stochastic differential equations," *SIAM Journal on Scientific Computing*, vol. 24, no. 2, pp. 619–644, 2002. DOI: 10.1137/S1064827501387826.

[11] R. B. Nelsen, *An Introduction to Copulas*, ser. Lecture Notes in Statistics. Springer-Verlag, 2006. DOI: 10.1007/0-387-28678-0.

[12] A. Alexanderian, "On spectral methods for variance based sensitivity analysis," *Probab. Surveys*, vol. 10, pp. 51–68, 2013. DOI: 10.1214/13-PS219.

[13] G. Blatman and B. Sudret, "An adaptive algorithm to build up sparse polynomial chaos expansions for stochastic finite element analysis," *Probabilistic Engineering Mechanics*, vol. 25, no. 2, pp. 183–197, 2010.

[14] J. Slim, F. Rathmann, A. Nass, H. Soltner, R. Gebel, J. Pretz, and D. Heberling, "Polynomial chaos expansion method as a tool to evaluate and quantify field homogeneities of a novel waveguide rf wien filter," *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 859, pp. 52–62, 2017. DOI: 10.1016/j.nima.2017.03.040.

[15] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani, "Least angle regression," *Ann. Statist.*, vol. 32, no. 2, pp. 407–499, April 2004. DOI: 10.1214/009053604000000067.

[16] O. Garcia-Cabrejo and A. Valocchi, "Global sensitivity analysis for multivariate output using polynomial chaos expansion," *Reliability Engineering & System Safety*, vol. 126, pp. 25–36, 2014. DOI: 10.1016/j.ress.2014.01.005.