

University of Alberta

DEVELOPING A KINETIC MODEL FOR HYDROCONVERSION PROCESSING
OF VACUUM RESIDUE

by

Shiva Shams

A thesis submitted to the Faculty of Graduate Studies and Research
in partial fulfillment of the requirements for the degree of

Master of Science

in

Process Control

Department of Chemical and Materials Engineering

©Shiva Shams
Spring 2011
Edmonton, Alberta

Permission is hereby granted to the University of Alberta Libraries to reproduce single copies of this thesis and to lend or sell such copies for private, scholarly or scientific research purposes only. Where the thesis is converted to, or otherwise made available in digital form, the University of Alberta will advise potential users of the thesis of these terms.

The author reserves all other publication and other rights in association with the copyright in the thesis and, except as herein before provided, neither the thesis nor any substantial portion thereof may be printed or otherwise reproduced in any material form whatsoever without the author's prior written permission.

Abstract

One of heavy oils upgrading processes is hydroconversion. As it is a complex process involving many chemical reactions, the mathematical model of hydroconversion process often has more kinetic parameters than can be estimated from the data. In this thesis, a model for hydroconversion processing of vacuum residue is proposed. It is proved that the model is structurally identifiable, but shown that it is inestimable and good parameter estimates may be impossible to obtain even if the model fit is good. As a proof to the model inestimability, it is shown that literature data can be fitted using a subset of only three (of seven) parameters. To improve parameter estimability, a method is proposed for designing additional experiments. The method is based on designing experiments that provide data that is complementary (in an appropriate sense) to existing data. The approach is illustrated using the hydroconversion model. For the hydroconversion model, using two additional experiments provides a good balance between parameter estimation and experimental effort.

Acknowledgements

I would like to thank all the people who helped me finish this thesis in one way or another, as this thesis would not have been completed without their help and guidance.

First and foremost, I would like to thank my supervisor, Dr. Amos Ben-Zvi, who helped me all the way to the end. He taught me how to think as a researcher, how to approach a problem and how to be as a person. Without his guidance and patience, I would not have completed my master's.

Next, I would like to express my sincere thanks to Dr. Murray Gray and Dr. William McCaffrey, who were always available to answer my questions with such patience, and were ready to share their invaluable knowledge and experience with me.

I also would like to give my thanks to Hector De la Hoz Siegler, who was always ready to answer my questions patiently, and whose support made my research go smoothly.

Finally, I would like to convey my deepest gratitude for my mother, my sister and my friend, Reihaneh, who were all such a great support to me.

Contents

1	Introduction	1
1.1	Objectives and contributions	3
1.2	Thesis outline	5
	Bibliography	6
2	Background	8
2.1	Hydroconversion process	8
2.2	Hydroconversion models	9
2.2.1	The base models	12
2.3	Parameter estimation	13
2.3.1	Identifiability	14
2.3.2	Estimability	15
2.3.3	Model reparameterization	16
2.4	Optimal experimental design	16
	Bibliography	18
3	Hydroconversion modeling	22
3.1	Introduction	22
3.2	Available Data	24
3.3	Proposed model	25
3.4	Model Identifiability & Estimability	28
3.4.1	Identifiability	28
3.4.2	Estimability	30
3.5	Parameter estimation	33
3.6	Optimal experimental design	39
3.7	Conclusion	44
3.8	Acknowledgements	45
	Bibliography	46
4	Conclusions and future work	49
4.1	Conclusions	49
4.2	Future work	50
	Bibliography	52
A	Maple codes	53

B	MATLAB codes	64
B.1	Parameter estimation	64
B.2	Optimal experimental design	80
B.3	Parameter back-calculations for different experimental designs	90

List of Figures

2.1	Reaction scheme used by Carbonell and Guirardello (1997). Numbers in arrows are the first-order rate constants relative to the rate constant of hard resid cracking to product gas oil and are from Mosby et al. (1986). i is the number to identify each lump. (Carbonell and Guirardello, 1997)	10
2.2	Reaction scheme used by Panariti et al. (2000)	11
3.1	Singular values comparison	34
3.2	Error comparisons for optimization problems with different g .	37
3.3	Model fit for $g = 3$ and confidence bounds for mean responses	38
3.4	Error comparison of parameter estimation for different designs	44

List of Tables

3.1	Properties of Belayim vacuum residue (Del Bianco et al., 1994)	24
3.2	Coke yield (wt%) and asphaltene conversion (%) of hydroconversion process at 450°C & 20 MPa with 1000 ppm Mo as MoN (Del Bianco et al., 1994)	24
3.3	Experimental data for P=20 MPa & T=450°C & catalyst=1000 ppm (Del Bianco et al., 1994).	28
3.4	Nominal values for parameters	33
3.5	Confidence intervals for pseudo-parameters	37
3.6	Parameter estimations for $g = 3$	39
3.7	Optimal configurations for different experimental designs	42
3.8	Parameter back-calculation for different designs	42
3.9	Properties of different designs	43

Chapter 1

Introduction

As the conventional crude oil reserves are declining and the heavy oils and bitumens reserves are large comparing to conventional crude oil, particular attention has been paid to upgrading processes of these heavy feedstocks, particularly, the conversion of vacuum residue (boiling over 524 °C) to distillates (Gray & McCaffrey, 2002). Upgrading processes can be divided into four categories (Schuetze & Hofmann, 1984):

1. Thermal processes
2. Catalytic hydroprocesses
3. Non-catalytic hydroprocesses
4. Catalytic cracking

Vacuum residues, containing cokable materials (referred to as Conradson Carbon Residue, CCR) and components with sulphur, nitrogen and metals, need to be upgraded so that the contents of these undesirable materials are reduced (Del Bianco et al., 1993). Two common upgrading processes for vacuum residues are coking and hydroconversion (Gray & McCaffrey, 2002). Coking is a thermal process, which has the following disadvantages: “low CCR reduction and poor heteroatoms removal and large volumes of high sulphur, high metals coke” (Del Bianco et al., 1993). Therefore, hydroconversion as a catalytic hydroprocess is of particular interest. During hydroconversion, the feed cracks in the presence of hydrogen and a suitable catalyst, and a portion of

the heteroatom components are removed (Carbonell & Guirardello, 1997; De Almeida & Guirardello, 2005). Also, the presence of hydrogen and catalyst suppresses coke formation considerably.

Coke as an undesirable byproduct of hydroconversion, causes fouling of the internals in a high-pressure reactor and catalyst deactivation. Process design, optimization and control are powerful tools for the prevention of coke formation. A key condition for design, optimization and control of the process is having a model, which describes the system kinetics accurately. A chemical process such as hydroconversion can be modeled based on physics first principles considering heat, mass, momentum transfer rates, reaction rates, etc.. Moreover, to obtain a computationally tractable model for oil refining processes, lumping methodology is often used (De Almeida & Guirardello, 2005; Rahmani et al., 2002; Wiehe, 1993) as the feed is a complex mixture containing many different types of molecules.

In this work, a model is proposed based on work by Wiehe (1993) and Rahmani et al. (2002). Before this model can be used, a couple of model characteristics have to be examined. The first characteristic is model Identifiability. A model is *structurally* identifiable if there is a unique set of parameters explaining the input-output relation (Jayasankar et al., 2009). Having the ideal (rich enough and noise-free) data set, this unique set of parameters can be estimated for a structurally identifiable model. If a model is structurally unidentifiable, it has to be re-structured and/or some of the parameters be removed from the model. The next characteristic to be checked after the model is proved to be identifiable is model estimability. A model is estimable, if for a set of real-world data or a specific experimental design, the parameters can be estimated accurately (Jayasankar et al., 2009). If different parameter values lead to indistinguishable predictions, the model is not estimable.

The model proposed in this work contains five states, two outputs and seven unknown parameters, and is described in a set of six ordinary differential equations. Based on the model structure, it was decided that the system represented by this model is bilinear. In the next step, the Identifiability of this model is examined, and the proposed model is proved to be identifiable

using the method by Vadjia (1985) for bilinear systems. Then, the model is checked for estimability using the experimental design by Del Bianco et al. (1994). The approach chosen to examine the model estimability involves performing Singular Value Decomposition (SVD) on the parameter covariance matrix. Using SVD results, the parameters are reparameterized to a new set of seven pseudo-parameters, which are the linear combinations of the former parameters. SVD also allows sorting these pseudo-parameters based on their significance in changing the output values. Considering this sorting, estimating only the three most important pseudo-parameters and fixing the other four at their nominal values results in a good model fit to the data by Del Bianco et al. (1994).

Considering that the other four pseudo-parameters can become more important in another experimental design, having only three pseudo-parameters estimated does not guarantee a model with good design capabilities. Therefore, a set of optimal experiments need to be designed using the model and the estimates of the three pseudo-parameters to make it possible to estimate all seven pseudo-parameters. This is accomplished in this work using the D-optimality criteria. It is shown that while one additional experiment does not complement the current data considerably, two additional experiments with the optimal designs proposed can lead to the estimation of all seven parameters and therefore pseudo-parameters.

1.1 Objectives and contributions

The stage-wise objectives of this work are as following:

1. Propose a model for Hydroconversion processing of vacuum residue.
2. Test the proposed model for Identifiability using a suitable Identifiability test.
3. Check the model for estimability using the SVD.
4. Use the estimability results to find the directions in the parameter space

along which the change in parameters affect the model predictions significantly.

5. Perform a linear reparameterization on the model and sort the new set of pseudo-parameters based on their significance in changing the model predictions.
6. Determine the number of pseudo-parameters that can be estimated accurately using the literature data.
7. Design a set of optimal experiments using the model and the estimates of the pseudo-parameters.
8. Decide which design is the best considering both the effort and the result.

The notable contributions of this work are as following:

1. Proposed a novel model for Hydroconversion processing of vacuum residue based on work by Wiehe and Rahmani et al..
2. Used a suitable Identifiability and proved that the model is identifiable.
3. Developed a method to check the model for estimability performing SVD.
4. Reparameterized the model and obtained a set of pseudo-parameters, which are the linear combinations of the model parameters.
5. Sorted the pseudo-parameters based on how much their change is significant in changing the output.
6. Fitted the model to the literature data estimating only three (out of seven) pseudo-parameters and fixing the rest at their nominal values.
7. Designed a set of two additional experiments to complement the current data to make it possible to estimate all the parameters.

1.2 Thesis outline

In this thesis, a model for Hydroconversion processing of vacuum residue is developed. **Chapter 2** contains the background information and literature review for the topics discussed in this work. These topics are: Hydroconversion, Hydroconversion modeling, the base models, Identifiability, estimability, model reparameterization and optimal experimental design. **Chapter 3**, submitted paper, contains the proposed model and evaluations done about the model. **Chapter 4**, contains the summary of this work results and the suggestions for the future work. **Appendix A** contains Maple codes used for performing identifiability test. **Appendix B** contains MATLAB codes used at every stage of work.

Bibliography

- [1] Carbonell MM, Guirardello R. In: Modelling of a slurry bubble column reactor applied to the hydroconversion of heavy oils. Proceedings of the 1997 3rd international conference on gas-liquid-solid reactor engineering, december 7, 1997 - december 10; 1997; Shonan, Jpn: Elsevier Science Ltd; 1997. p. 4179-85.
- [2] De Almeida RM, Guirardello R. Hydroconversion kinetics of marlim vacuum residue. *Catalysis Today*. 2005;109(1-4):104-11.
- [3] Del Bianco A, Panariti N, Carlo D, Beltrame PL, Carniti P. New development in deep hydroconversion of heavy oil residues with dispersed catalysts. 2. kinetic aspects of reaction. *Energy and Fuels*. 1994;8(3):593-7.
- [4] Del Bianco A, Panariti N, Di Carlo S, Elmouchnino J, Fixari B, Le Perchec P. Thermocatalytic hydroconversion of heavy petroleum cuts with dispersed catalyst. *Applied Catalysis A: General*. 1993;94(1):1-16.
- [5] Gray MR, McCaffrey WC. Role of chain reactions and olefin formation in cracking, hydroconversion, and coking of petroleum and bitumen fractions. *Energy and Fuels*. 2002;16(3):756-66.
- [6] Jayasankar BR, Ben-Zvi A, Huang B. Identifiability and estimability study for a dynamic solid oxide fuel cell model. *Computers and Chemical Engineering*. 2009;33(2):484-92.
- [7] Rahmani S, McCaffrey W, Gray MR. Kinetics of solvent interactions with asphaltenes during coke formation. *Energy and Fuels*. 2002;16(1):148-54.

- [8] Schuetze B, Hofmann H. How to upgrade heavy feeds. *Hydrocarbon Process.* 1984;63(2):75-82.
- [9] Surisetty K. Non-linear reparameterization of complex models with applications to a microalgal heterotrophic fed-batch bioreactor [dissertation]. 2009.
- [10] Vajda S. In: *Structural identifiability of linear, bilinear, polynomial and rational systems. Volume 6: Biomedical applications, water resources, environment, energy systems, development, social effects, SWIIS; Education; Budapest, Hung: IFAC by Pergamon Press; 1985. p. 717-22.*
- [11] Wiehe IA. In: *Phase separation kinetic model for coke formation. Symposium on resid upgrading presented at the 205th national meeting of the american chemical society, march 28, 1993 - april 2; 1993; Denver, CO, USA: Published by ACS; 1993. p. 428-33.*

Chapter 2

Background

2.1 Hydroconversion process

Hydroconversion is one of the upgrading processes for heavy oils and petroleum residues (Carbonell & Guirardello, 1997). When maximum diesel oil is required, hydroconversion of vacuum residue is preferred (De Ameida & Guirardello, 2005). Hydroconversion is cracking and hydrogenation of the feedstock in the presence of hydrogen and a catalyst (Bearden & Aldridge, 1979; Carbonell & Guirardello, 1997). During this process, a portion of compounds with high molecular weight are converted to lower boiling hydrocarbon products, and a portion of heteroatoms are removed (Bearden & Aldridge, 1979; Carbonell & Guirardello, 1997). In other words, two reactions happen during hydroconversion process: 1. Thermal cracking, 2. Catalytic hydrogenation and removal of contaminants. (Carbonell & Guirardello, 1997). The catalysts have no role in cracking reactions; the cracking reactions are thermal and independent of catalyst concentration and hydrogen pressure in hydroconversion (Panariti et al., 2000). The roles of catalysts are favoring hydrogenation reactions and suppressing coke formation (Panariti et al., 2000). However, hydroconversion term is sometimes used more broadly and also for processes with bifunctional catalysts in which catalytic cracking also occurs (Chavarria-Hernandez et al., 2009; De Ameida & Guirardello, 2005).

The operating conditions for hydroconversion process are severe pressure (7–25 MPa) (Carbonell & Guirardello, 1997) and temperature (400–500°C). The reactor types for hydroconversion are: fixed bed, moving bed, ebullated bed

and slurry reactor (Rana et al., 2007). In slurry hydroconversion the selected catalyst is dispersed in the feedstock (Rana et al., 2007). As dispersed catalysts are capable of controlling coke formation, slurry processes are good alternatives for conventional hydrocracking processes (Panariti et al. 2000). Although there is a large amount of data regarding the use of dispersed catalysts (Del Bianco et al., 1993), not much information is available about the kinetics of the process (Panariti et al., 2000).

2.2 Hydroconversion models

In order to commercialize a process, the process has to be modeled for design and scale up purposes. Moreover, the process model can be used in later on procedures including process optimization and control, which result in the better performance of the process plant and obtaining more of the desired products. A few models exist for hydroconversion processing of heavy hydrocarbons in the literature, and some of them are reviewed in this section.

Carbonell and Guirardello (1997) used a computational fluid dynamics approach to model the slurry bubble column reactor used in heavy oils hydroconversion. They believed that the models, which consider fluid dynamics of the process, are more accurate and can enhance the design and scale up of slurry bubble column reactor. Based on experimental evidence by Mosby et al. (1986), they assumed that thermal cracking and removal of contaminants are two independent reactions going on in the hydroconversion reactor. Therefore, They carried out their simulations in two steps without considering coke formation and hydrogenation reactions: 1. Determination of the fluid dynamics, 2. Simulation of the thermal cracking reaction. They also assumed that thermal cracking reactions did not change the fluid dynamics because the liquid phase was highly backmixed and the rate of gas adsorption was low.

The kinetic model that was used in Carbonell and Guirardello (1997) to simulate the thermal cracking reactions had been developed by Mobsy et al. (1986). In this model, the feed is divided into different lumps based on the boiling point range, and each lump is regarded as a single chemical component

separately taking part in the reaction. The reaction scheme for this model can be seen in Figure 2.1.

The model by Carbonell and Guirardello (1997) was in good agreement with experimental data. However, more experimental data is needed to confirm this agreement.

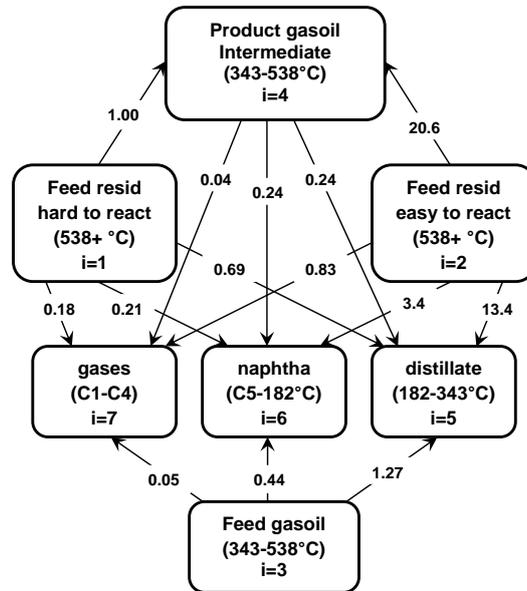


Figure 2.1: Reaction scheme used by Carbonell and Guirardello (1997). Numbers in arrows are the first-order rate constants relative to the rate constant of hard resid cracking to product gas oil and are from Mosby et al. (1986). i is the number to identify each lump. (Carbonell and Guirardello, 1997)

Panariti et al. (2000) carried out hydroconversion of Belayim residue in the presence of dispersed molybdenum added as molybdenum naphthenate. The process was operated at temperature of 400-460°C, pressure of 8-16 MPa with 0-5000 ppm Mo, and the effect of different operating conditions and catalyst loading was studied.

They then proposed a reaction scheme (Figure 2.2) for hydroconversion processing based on the experimental results they obtained. Again, lumping methodology was implemented, and four pseudo-components were introduced: 1. Vacuum residue (VR), 2. Vacuum residue that has accepted hydrogen (VRH), 3. Distillate, 4. Coke. Based on the reaction scheme, the vacuum

residue lump could go through two competing reactions: 1. Direct conversion to distillate and coke 2. Catalytic hydrogenation.

According to their results, the catalyst concentration and hydrogen pressure did not affect the production rate of distillate significantly as it is thermally controlled, and What they affected was the amount of coke formation.

Using Panariti et al.'s model, the performance of hydroconversion process can be optimized over different operating conditions and catalyst concentrations.

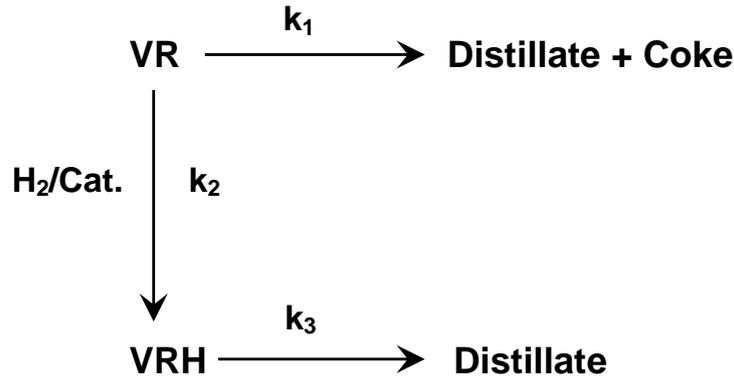


Figure 2.2: Reaction scheme used by Panariti et al. (2000)

Del Bianco et al. (1994) carried out hydroconversion process for vacuum residue of Belayim crude in the presence of molybdenum naphthenate as hydrogenation catalyst at temperature of 410-450°C, pressure of 20 MPa with 200-5000 ppm Mo as MoN.

They then proposed a kinetic model for vacuum residue conversion. Their modeling assumptions were: 1. Hydrogen pressure is nearly constant during the reaction, and therefore, the reaction order with respect to hydrogen pressure is zero, 2. The hydrogen transfer to gas-liquid interface is assumed to be very fast comparing to hydrogen consumption in the liquid phase due to the reaction 3. Coke formation pathway is neglected due to the small amount of coke being formed 4. The step of catalyst transformation and activation is considered very fast comparing to vacuum residue conversion. 5. Vacuum

residue conversion is thermally controlled.

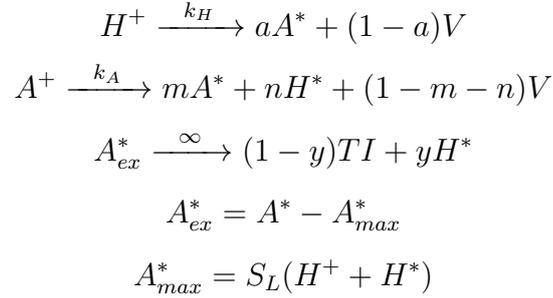
Finally, the reaction order with respect to vacuum residue concentration was computed as two, and their model could predict the vacuum residue conversion at different temperatures and pressure.

2.2.1 The base models

There are two models on which the modeling efforts in this work are based. Each of these models suggests some ideas about how to make the essential assumptions for the hydroconversion processing model.

Wiehe's model

Wiehe (1993) suggested a kinetic model for thermolysis of vacuum residue by lumping the vacuum residue according to its solubility and volatility characteristics. In this model, coke is assumed to form because heavy hydrocarbons phase separate when they exceed the solubility limit of the light hydrocarbons (the arrows indicate that a reaction is happening):

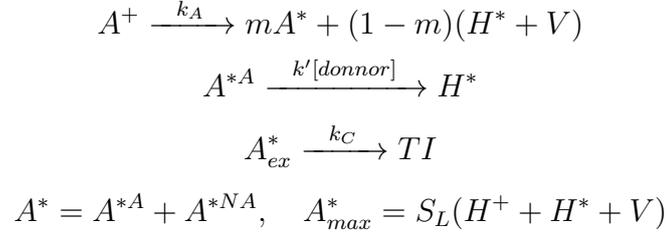


where A^+ is reacting asphaltenes; A^* , asphaltene cores; H^+ , reacting non-volatile heptane-solubles; H^* , produced nonvolatile heptane-solubles; V , volatiles; A_{max}^* , maximum asphaltene cores that can be held in solution; A_{ex}^* , excess asphaltene cores which cannot be held in solution; TI , toluene-insoluble coke; S_L , solubility limit (wt%/wt%); a , m , n and y , stoichiometric coefficients; k_H and k_A , first-order reaction rate constants (min^{-1}).

From Wiehe's model, the assumptions about how coke is formed and how the system is going to be lumped are made.

Rahmani et al.'s model

Rahmani et al. (2002) developed a model for thermolysis of asphaltenes in presence of hydrogen donors based on the principles of Wiehe's model. The model for the case of a closed reactor is as follows (the arrows indicate that a reaction is happening):



The assumptions, which Rahmani et al. (2002 based their model on, are:

1. Asphaltene cores do not form coke instantly as they phase separate. Therefore, a finite first-order rate constant, k_C , is used.
2. Asphaltene cores have a limited capacity for accepting hydrogen. Thus, they are divided into two types: cores that can accept enough hydrogen to change their solubility characteristics, A^{*A} , and cores that cannot, A^{*NA} .
3. The cores that accept sufficient hydrogen from a donor convert into heptane-soluble material, and the apparent first-order rate constant for hydrogen transfer is the product of the rate constant and the donor concentration.

From Rahmani et al.'s model, one can learn how to insert hydrogen into Wiehe's model, which is needed for hydroconversion processing model as hydrogen is present in the system. The changes made to these base models to make them applicable to hydroconversion processing are presented in Chapter 3.

2.3 Parameter estimation

To analyze performances of the physical and chemical systems under different conditions, one has to model these systems. Typically, Physical or chemical

systems are modeled mathematically using first principles, and the model parameters are estimated using the experimental data (Jayasankar et al., 2009). The resulting models then are used for design, scale up, control and optimization purposes (Jayasankar et al., 2009).

Parameter estimation is a crucial part of the modelling process. Without estimating the model parameters, the system model would be of little use, because it cannot predict the system behaviour under different conditions. However, parameter estimation is not a straightforward process at most of the times. First, the model has to be checked for identifiability. A model is structurally identifiable if all the model parameters can be determined uniquely having the ideal data set. Second, the model has to be tested for estimability, which means the model parameters can be estimated accurately using the current experimental design and data set in hand. Finally, if the model was both identifiable and estimable, the parameters can be estimated without any modifications made to the model. However, that is not the case most of the times, and model parameters can be unidentifiable, or even identifiable but inestimable.

In the following sections, identifiability and estimability are defined. Moreover, methods that are available in the literature to check for these two characteristics are explained briefly. Finally, approaches available to solve unidentifiability and inestimability are touched.

2.3.1 Identifiability

A model is identifiable if and only if a unique set of parameters can be determined for the model using the ideal experimental data (Jayasankar et al., 2009). If there is more than one parameter set for the model that can explain the relationship between the input and the system response, then the model is not identifiable (Jayasankar et al., 2009).

Some of the methods available to check for identifiability of models containing non-linear ordinary differential equations are Taylor series, generating series, local state isomorphism and differential algebraic approaches which are explained in details with examples in Walter and Pronzato (1996). Taylor

series method and local state isomorphism are compared in Chappell et al. (1990). Another method is linearization-based approach by Ben-Zvi et al. (2006). Methods particular for testing the identifiability of linear, bilinear, polynomial and rational systems can be found in Vajda (1985).

To solve the model unidentifiability, the model can be re-structured and/or the parameter whose change does not affect the model response can be fixed at its nominal value or removed from the model (Surisetty, 2009).

2.3.2 Estimability

A model is estimable if the model parameters can be estimated accurately from a data set or a specific experimental design (Jayasankar et al., 2009). When a specific experimental design cannot provide enough excitation so that all the parameters can be estimated, the model is inestimable for that experimental design. It is possible for identifiable models to be inestimable (Surisetty, 2009).

The model parameters are regarded as inestimable when they do not have statistically significant effect on the model predictions (Ben-Zvi, 2008; Yao et al., 2003). One method to test a model for estimability is sensitivity analysis during which a sensitivity coefficient matrix is formed and computed. If the matrix is of full column rank, then the model is estimable (Yao et al., 2003). Yao et al. proposed a method to determine which parameters are estimable and which are not for models with large number of parameters. This method was also used by Jayasankar et al. (2009). For models with a few numbers of parameters and outputs, the sensitivity coefficients are plotted with time. If any significant correlation can be detected in the plots, the model is inestimable (Beck & Arnold, 1977). Another method to test the estimability is through computing eigenvalues and eigenvectors of the Fisher Information Matrix (Vajda et al., 1989). The number of small eigenvalues corresponds to the number of parameter combinations which are not estimable.

When a system is inestimable, the parameter estimates obtained for this system are prone to be inaccurate (Ben-Zvi, 2008). With inaccurate parameter estimates, the system model cannot be used for scale up, design, control and optimization (Ben-Zvi, 2008). There two solutions for the inestimability

problem: 1. Obtaining more data, 2. Modifying the system model (Ben-Zvi, 2008). To obtain more data, more experiments have to be conducted, which may increase the cost or be impossible for the system under study (Ben-Zvi, 2008). Therefore, the second solution is usually chosen, and the model gets modified in order to reduce the number of parameters and improve the parameter estimates (Ben-Zvi, 2008).

2.3.3 Model reparameterization

There are several ways that one can modify and reduce the system model. Chakravarti et al. (2001) fixed the parameters that do not have a significant effect on the model predictions at a nominal value. Matos et al. (2001) used parameter correlation information to fix the parameters at their nominal values sequentially. Yao et al. (2003) used a sensitivity-based approach to determine an estimable subset of parameters. Preisig (2007) proposed a Bayes theorem-based approach in which different parameter subsets are determined at different times. Correa et al. (2005) also proposed a similar Monte-Carlo-based approach to measure estimability. Khare et al. (2002) used an algorithm in which parameter values are manually changed until having a satisfactory fit.

Methods listed above are all based on identifying an inestimable subset of parameters. However, the parameter combinations are often inestimable not the individual parameters. Ben-Zvi (2008) proposed a transformation-based approach for reparameterization of unidentifiable or inestimable systems. In this approach the parameter space is partitioned into an estimable part and an inestimable part.

2.4 Optimal experimental design

If a system is identifiable but not estimable, and it is possible to conduct more experiments, then there is a set of additional experiments that can complement the existing data and make it possible for all the parameters to be estimable or reduce the inestimability problem. To reduce the cost and effort associated

with experimental design, one has to design an optimal set of experiments. One can formulate the optimal experimental design problem as a maximization or minimization problem of a certain functional of the Fisher information matrix (FIM) with respect to the experimental conditions (Balsa-Canto et al., 2008). With this formulation, the design can consist of several sequential or simultaneous experiments with the initial conditions, sampling times, duration of the experiments and time-dependent input as degrees of freedom (Balsa-Canto et al., 2008). Fisher information matrix where having the least squares regression is defined as (Marsili-Libelli et al., 2003):

$$FIM = \sum_{i=1}^N (\partial y_i / \partial p)^T V_i^{-2} (\partial y_i / \partial p)$$

where p is the parameter vector, y_i , the output vector at time instant i , $(\partial y_i / \partial p)$, the sensitivity matrix at time instant i , and V_i is a diagonal matrix containing weighting factors.

There are several optimality criteria that can be chosen as FIM-based cost functional including D-, A-, E-, G-, and V-optimal (Balsa-Canto et al., 2008; Mannarswamy et al., 2009). However, the most popular ones are D- and E-Optimalities, which are defined as (Balsa-Canto et al., 2008):

1. D-optimality: Maximization of determinant of FIM, D-optimality criterion focuses on minimizing the volume of the confidence region in the parameter space and ensures collecting the maximum amount of information provided by the experiments, but does not indicate how the information is distributed among the parameters.
2. E-optimality: Maximization of minimum eigenvalue of FIM, E-optimality is about minimizing the maximum error which leads to reaching a compromise between the amount of collected information and the way it is distributed among the parameters.

Bibliography

- [1] Balsa-Canto E, Alonso AA, Banga JR. Computational procedures for optimal experimental design in biological systems. *IET Systems Biology*. 2008;2(4):163-72.
- [2] Bearden JR, Aldridge CL; Hydroconversion of heavy hydrocarbons. patent 4134825. 1979.
- [3] Beck JV, Arnold KJ. Parameter estimation in engineering and science. Toronto, Canada: John Wiley & Sons; 1977.
- [4] Ben-Zvi A, McLellan PJ, McAuley KB. Identifiability of non-linear differential algebraic systems via a linearization approach. *Canadian Journal of Chemical Engineering*. 2006;84(5):590-6.
- [5] Ben-Zvi A. Reparameterization of inestimable systems with applications to chemical and biochemical reactor systems. *AICHE Journal*. 2008;54(5):1270-81.
- [6] Carbonell MM, Guirardello R. In: Modelling of a slurry bubble column reactor applied to the hydroconversion of heavy oils. Proceedings of the 1997 3rd international conference on gas-liquid-solid reactor engineering, december 7, 1997 - december 10; 1997; Shonan, Jpn: Elsevier Science Ltd; 1997. p. 4179-85.
- [7] Chakravarti S, Ray WH, Zhang SX. Kinetic study of olefin polymerization with a supported metallocene catalyst. IV. comparison of bridged and unbridged catalyst in gas phase. *Journal of Applied Polymer Science*. 2001;81(6):1451-9.

- [8] Chappell MJ, Godfrey KR, Vajda S. Global identifiability of the parameters of nonlinear systems with specified inputs. A comparison of methods. *Mathematical Biosciences*. 1990;102(1):41.
- [9] Chavarria-Hernandez J, Ramirez J. Modeling ideal and nonideal hydrocracking of paraffins using the single-event lumped parameter hybrid (SELPH) model. *Industrial and Engineering Chemistry Research*. 2009;48(3):1203-7.
- [10] Correa JM, Farret FA, Popov VA, Simoes MG. Sensitivity analysis of the modeling parameters used in simulation of proton exchange membrane fuel cells. *IEEE Transactions on Energy Conversion*. 2005 03;20(1):211-18.
- [11] De Almeida RM, Guirardello R. Hydroconversion kinetics of marlim vacuum residue. *Catalysis Today*. 2005;109(1-4):104-11.
- [12] Del Bianco A, Panariti N, Carlo D, Beltrame PL, Carniti P. New development in deep hydroconversion of heavy oil residues with dispersed catalysts. 2. kinetic aspects of reaction. *Energy and Fuels*. 1994;8(3):593-7.
- [13] Jayasankar BR, Ben-Zvi A, Huang B. Identifiability and estimability study for a dynamic solid oxide fuel cell model. *Computers and Chemical Engineering*. 2009;33(2):484-92.
- [14] Khare NP, Seavey KC, Liu YA, Ramanathan S, Lingard S, Chen C. Steady-state and dynamic modeling of commercial slurry high-density polyethylene (HDPE) processes. *Industrial and Engineering Chemistry Research*. 2002;41(23):5601-18.
- [15] Mannarswamy A, Munson-McGee SH, Steiner R, Andersen PK. D-optimal experimental designs for freundlich and langmuir adsorption isotherms. *Chemometrics and Intelligent Laboratory Systems*. 2009;97(2):146-51.

- [16] Marsili-Libelli S, Guerrizio S, Checchi N. Confidence regions of estimated parameters for ecological systems. *Ecological Modelling*. 2003;165(2-3):127-46.
- [17] Matos V, Mattos Neto AG, Carlos Pinto J. Method for quantitative evaluation of kinetic constants in olefin polymerizations. I. kinetic study of a conventional ziegler-natta catalyst used for propylene polymerizations. *Journal of Applied Polymer Science*. 2001;79(11):2076-108.
- [18] Mosby JF, Buttke RD, Cox JA, Nikolaides C. Process characterization of expanded-bed reactors in series. *Chemical Engineering Science*. 1986;41(4):989-95.
- [19] Panariti N, Del Bianco A, Del Piero G, Marchionna M, Carniti P. Petroleum residue upgrading with dispersed catalysts. part 2. effect of operating conditions. *Applied Catalysis A: General*. 2000;204(2):215-22.
- [20] Preisig HA. In: Incremental nonlinear parameter estimation in dynamical systems. International the IFAC symposium on dynamics and control of process systems; ; 2007. p. 297-302.
- [21] Rahmani S, McCaffrey W, Gray MR. Kinetics of solvent interactions with asphaltenes during coke formation. *Energy and Fuels*. 2002;16(1):148-54.
- [22] Rana MS, Samano V, Ancheyta J, Diaz JAI. A review of recent advances on process technologies for upgrading of heavy oils and residua. *Fuel*. 2007;86(9):1216-31.
- [23] Surisetty K. Non-linear reparameterization of complex models with applications to a microalgal heterotrophic fed-batch bioreactor [dissertation]. 2009.
- [24] Vajda S, Rabitz H, Walter E, Lecourtier Y. Qualitative and quantitative identifiability analysis of nonlinear chemical kinetic models. *Chemical Engineering Communications*. 1989;83(1):191-219.

- [25] Vajda S. In: Structural identifiability of linear, bilinear, polynomial and rational systems. Volume 6: Biomedical applications, water resources, environment, energy systems, development, social effects, SWIIS; Education.; Budapest, Hung: IFAC by Pergamon Press; 1985. p. 717-22.
- [26] Walter E, Pronzato L. On the identifiability and distinguishability of nonlinear parametric models. *Mathematics and Computers in Simulation*. 1996;42(2-3):125-34.
- [27] Wiehe IA. In: Phase separation kinetic model for coke formation. Symposium on resid upgrading presented at the 205th national meeting of the american chemical society, march 28, 1993 - april 2; 1993; Denver, CO, USA: Published by ACS; 1993. p. 428-33.
- [28] Yao KZ, Shaw BM, Kou B, McAuley KB, Bacon DW. Modeling ethylene/butene copolymerization with multi-site catalysts: Parameter estimability and experimental design. *Polymer Reaction Engineering*. 2003;11(3):563-88.

Chapter 3

Hydroconversion modeling¹

3.1 Introduction

Hydroconversion is an upgrading process for heavy oils and petroleum residues. During hydroconversion, the heavy oil is cracked in presence of hydrogen and a catalyst to produce lighter oil fractions with a lower content of heteroatom compounds (metals, nitrogen, sulphur, and oxygen) and cokable materials (Carbonell & Guirardello, 1997; De Almeida & Guirardello, 2005). Hydroconversion processes can be categorized based on reactor type: fixed bed, moving bed, ebullated bed, and slurry (Rana et al., 2007). Typically, the reactor is operated at high pressure (7-25 MPa (Carbonell & Guirardello, 1997)) and temperature (400-500 °C) during the hydroconversion process. In the case of slurry hydroconversion, catalyst is fed into the reactor in a finely dispersed form along with the feedstock (Rana et al., 2007).

One key undesirable byproduct of the hydroconversion process is coke formation as it causes fouling of the internals in high-pressure reactors leading to premature shutdowns and safety problems. In addition, the formation and deposition of coke has an inhibitory effect on catalysts. Careful selection of reactor conditions may ameliorate the issue of coke formation. An accurate model describing the system kinetics is useful for suppressing coke formation while maintaining good yields of desirable oil fractions.

In order for a model to be useful for design, scale up, control, and optimiza-

¹A version of this chapter has been submitted for publication to Computers and Chemical Engineering Journal.

tion, a subset of the model parameters must typically be estimated. Often, some of the model parameters have little or no effect on model predictions and can be fixed at a nominal value. If a parameter has no effect on the model predictions regardless of the quality or quantity of the data collected then the model is not *structurally* identifiable. Several methods for testing the structural identifiability of model parameters have been proposed in the literature (Ben-Zvi et al., 2004; Vajda, 1985). If a parameter is not identifiable then it should be removed from a model or otherwise set to a nominal value. In practice it is possible that some model parameters are structurally identifiable, but have statistically insignificant effect on model predictions (Ben-Zvi, 2008; Yao et al., 2003). This situation corresponds to lack of estimability. Several methods exist for testing the estimability of model parameters in Ordinary Differential Equation (ODE) models (Jayasankar et al., 2009; Yao et al., 2003). Estimability has also been labeled *practical identifiability* in the literature (Vanrolleghem et al., 1995).

In this work, a model for hydroconversion processing of vacuum residue based on the previous work by Wiehe (1993) and Rahmani et al. (2002) is proposed. This model is tested for identifiability in order to ensure that it is structurally identifiable. Also, the estimability of model parameters is assessed using experimental data. It is shown that not all parameters are necessary for obtaining model predictions which match the experimental data. The results from the estimability analysis is validated using data for the hydroconversion processing of Belayim vacuum residue in the presence of molybdenum naphthenate (MoN) by Del Bianco et al. (1994). Also in this work, a set of optimal experimental designs are generated in order to make it possible for all the model parameters to be accurately estimated. In the case of the hydroconversion model, two additional experiments provide a good balance between accuracy in parameter estimates and experimental effort (i.e., cost).

3.2 Available Data

In this work, the estimability (or quality of parameter estimates) for a novel hydroconversion model will be assessed. The data used for assessing estimability will be from set of previously published data for hydroconversion processing of Belayim vacuum residue produced by Del Bianco et al. (1994). In Table 3.1, the properties of Belayim vacuum residue are listed. Table 3.2 lists the coke yields and asphaltene conversions for hydroconversion process at different sampling times. The experimental conditions used by Del Bianco et al. (1994) are pressure of 20 MPa, temperature of 450°C and catalyst concentration of 1000 ppm Mo as MoN. It should be noted that, in this data set, asphaltene is defined as pentane-insoluble, and coke as tetrahydrofuran-insoluble (THF-insoluble).

Table 3.1: Properties of Belayim vacuum residue (Del Bianco et al., 1994)

Density 15.4, g/ml	1.028
Viscosity at 100°C, cSt	5230
Pour point, °C	63
RCC, wt%	20.8
C5 asphaltenes, wt%	26.2
C7 asphaltenes, wt%	18.6
C, wt%	85.03
H, wt%	10.01
N, wt%	0.57
S, wt%	4.2
Ni, ppm	130
V, ppm	179

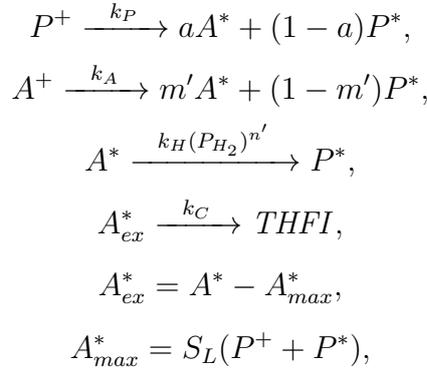
Table 3.2: Coke yield (wt%) and asphaltene conversion (%) of hydroconversion process at 450°C & 20 MPa with 1000 ppm Mo as MoN (Del Bianco et al., 1994)

Time (min)	Coke	Asphaltene Conversion
15	1	47.4
30	1.1	61.2
60	1.3	63.9
120	1.7	78.8
240	1.4	84.3

3.3 Proposed model

Heavy oil feedstocks are a complex mixture of many different types of molecules. In order to construct a computationally tractable model, lumping methodology has often been chosen to model petroleum refining processes (De Almeida & Guirardello, 2005; Rahmani et al., 2002; Wiehe, 1993). The model proposed in this work is based on previous work by Wiehe (1993) and Rahmani et al. (2002). In Wiehe (1993) a lumped-parameter model (i.e., an ODE system) is presented for the conversion of hydrocarbons via thermal cracking which includes induction period for coke formation. In Rahmani et al. (2002) the model proposed by Wiehe (1993) is modified to include the effect of the presence of hydrogen donors on the kinetics of thermal cracking. As hydroconversion process consists of cracking and hydrogenation reactions, the model by Rahmani et al. (2002) is used as a basis for the model developed in this work. However, it is modified to reflect the effect of catalyst presence and hydrogen pressure, and to fit the available data.

The model proposed in this work is given by the following set of reactions



where P^+ is reacting nonvolatile pentane-solubles; P^* , produced nonvolatile pentane-solubles plus volatiles; A^+ , reacting pentane-insoluble asphaltenes; A^* , asphaltene cores; A_{max}^* , maximum asphaltene cores that can be held in solution; A_{ex}^* , excess asphaltene cores which cannot be held in solution; $THFI$, THF-insoluble coke; S_L , solubility limit (wt%/wt%); a and m' , stoichiometric coefficients; k_P , k_A , and k_C , first-order reaction rate constants (min^{-1}); $k_H(P_{H_2})^{n'}$, apparent first-order rate constant for hydrogenation (min^{-1}). The assumptions for the model are:

1. The volatiles stay mainly in the liquid phase in a high-pressure closed reactor (Rahmani et al., 2002). Therefore, produced volatiles and light hydrocarbons are lumped together as P^* .
2. Coke is assumed to form because asphaltene cores phase separate when they exceed the solubility limit of the light hydrocarbons (Wiehe, 1993).
3. Asphaltene cores do not form coke instantly as they phase separate. Therefore, a finite first-order rate constant, k_C , is used (Rahmani et al., 2002).
4. Asphaltene cores that accept sufficient hydrogen convert into light hydrocarbons (Rahmani et al., 2002).
5. No limit is considered for the capability of asphaltene cores to accept hydrogen due to catalyst presence.
6. It can be assumed that hydrogenation rate constant is only correlated with $(P_{H_2})^{n'}$ while having the same catalyst and catalyst concentration. n' is shown to be 1 until a specific pressure, and zero afterwards (Qader et al., 1972). In this work, it is assumed that n' is 1 for 8-20 MPa.

To clarify Assumption 6, the mechanism of heterogeneous catalytic processes has to be explained. Based on Qader & Hill (1969), the heterogeneous catalytic processes can be explained in the following five steps: “1. diffusion of the reactant molecules from the bulk phase to the catalyst surface, 2. adsorption of the reactants on the catalyst surface, 3. surface reactions of the adsorbed molecules to form products, 4. desorption of the products, 5. diffusion of the desorbed products from the catalyst surface to the bulk phase”. Based on the activation enthalpies that Qader & Hill (1969) obtained for hydrocracking of low temperature tar, they concluded that the chemical reactions control the hydrogenation rate, and steps 1, 2, 4 and 5 are not limiting. In this work, it is assumed that the limiting reaction rate is the hydrogenation of the hydrocarbons on the catalyst surface, and the reactants concentrations on the catalyst surface are the same as their concentrations in the bulk phase.

Making these assumptions, hydrogenation reaction rate of the asphaltene cores (A^*) mostly depends on the hydrogen concentration (which is linearly correlated with hydrogen partial pressure (P_{H_2}) according to Henry's law (Trejo & Ancheyta, 2005)), A^* concentration and catalyst activity. Therefore, as it was mentioned in Assumption 6, the hydrogenation rate constant is correlated with $(P_{H_2})^{n'}$ while having the same catalyst activity. Hydrogenation reactions would also occur within the P^* and P^+ fractions, but these components would not change their solubility behavior.

The proposed model can be described using a set of non-linear coupled ordinary differential equations (ODEs) given by

$$\begin{aligned}
\frac{d[P^+]}{dt} &= -k_P[P^+], \\
\frac{d[P^*]}{dt} &= (1-a)k_P[P^+] + (1-m')k_A[A^+] + k_H P_{H_2}[A^*], \\
\frac{d[A^+]}{dt} &= -k_A[A^+], \\
\frac{d[A^*]}{dt} &= ak_P[P^+] + m'k_A[A^+] - k_H P_{H_2}[A^*] - k_C[A_{ex}^*].f([A_{ex}^*]), \\
\frac{d[THFI]}{dt} &= k_C[A_{ex}^*].f([A_{ex}^*]),
\end{aligned} \tag{3.1}$$

where

$$\begin{aligned}
[A_{ex}^*] &= [A^*] - S_L([P^+] + [P^*]), \\
f([A_{ex}^*]) &= \begin{cases} 0 & \text{if } [A_{ex}^*] \leq 0 \text{ ; during coke-induction period} \\ 1 & \text{if } [A_{ex}^*] > 0 \text{ ; after coke-induction period} \end{cases} .
\end{aligned}$$

The variables that can be measured experimentally for System 3.1 are the weight percentages of asphaltenes ($[A] = [A^+] + [A^*]$) and THF-insoluble coke ($[THFI]$). These measurable variables are available in Table 3.3 for each sampling time using Equation 3.2 and the data in Table 3.1 and 3.2. It has to be noted that according to the model equations $\frac{d}{dt}(P^+ + P^* + A^+ + A^* + THFI) = 0$. Therefore, the total content of pentane-solubles at each sampling time can be computed from the following equation: $[P] = [P^+] + [P^*] = constant - [A] - [THFI]$ (the constant is equal to 100 as the concentrations are reported in weight percent), which implies that there are only two independent

measurements at each sampling time ($[A]$ and $[THFI]$). As will be shown later, the relatively small number of measurements implies that only a small subset of the process parameters can be accurately estimated.

$$[A] = (1 - \text{Asphaltene conversion}(\%)/100) \times [A]_0. \quad (3.2)$$

Data in Table 3.3 will be used for the estimation of parameters in the system model described by Equation set 3.1.

Table 3.3: Experimental data for P=20 MPa & T=450°C & catalyst=1000 ppm (Del Bianco et al., 1994).

time (min)	Asphaltenes (wt%),[A]	THF-insoluble coke (wt%),[THFI]
0	26.2	0.0
15	13.8	1.0
30	10.2	1.1
60	9.5	1.3
120	5.6	1.7
240	4.1	1.4

3.4 Model Identifiability & Estimability

3.4.1 Identifiability

If some of the model parameters have no effect on the model predictions, the model is not structurally identifiable, and those parameters can be fixed at a nominal value or be removed from the model. In this section, the proposed model (i.e., System 3.1) is checked for identifiability to see if all the parameters have an effect on the model predictions. This test for identifiability can be accomplished in a computationally efficient manner, because System 3.1 is bilinear and can therefore be written as

$$\begin{cases} \dot{x} = A(p)x + B(p)u \\ y = Cx \end{cases},$$

where

$$x = \begin{bmatrix} [P^+] \\ [P^*] \\ [A^+] \\ [A^*] \\ [THFI] \end{bmatrix}, \quad y = \begin{bmatrix} [A] \\ [THFI] \end{bmatrix}, \quad u = P_{H_2}, p = \begin{bmatrix} k_P \\ k_A \\ k_H \\ k_C \\ m' \\ a \\ S_L \end{bmatrix},$$

$$A = \begin{bmatrix} -k_P & 0 & 0 & 0 & 0 \\ (1-a)k_P & 0 & (1-m')k_A & 0 & 0 \\ 0 & 0 & -k_A & 0 & 0 \\ ak_P + S_L k_C f & S_L k_C f & m' k_A & -k_C f & 0 \\ -S_L k_C f & -S_L k_C f & 0 & k_C f & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & k_H & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -k_H & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

$$C = \begin{bmatrix} 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

The identifiability of bilinear systems can be tested in a computationally efficient manner using the approach proposed by Vajda (1985). Under this approach, a sequence of matrices $\{Q_i\}_{i=0,1,\dots,k'}$ are computed as follows.

$$Q_0 = C, \quad Q_{k'} = \begin{bmatrix} Q_{k'-1}A \\ Q_{k'-1}B \end{bmatrix}.$$

For every value of $k' = 1, 2, \dots, 2n - 1$, with n the number of states, one can compute a vector $\tilde{Z}_{k'} = Q_{k'}x_0$. The vectors $\tilde{Z}_{k'}$ can be used to construct a matrix \tilde{S} given by

$$\tilde{S} = \begin{bmatrix} (\tilde{Z}_0)_1 & (\tilde{Z}_0)_2 & \cdots & (\tilde{Z}_0)_m \\ (\tilde{Z}_1)_1 & (\tilde{Z}_1)_2 & \cdots & (\tilde{Z}_1)_m \\ \vdots & & & \vdots \\ (\tilde{Z}_{2n-1})_1 & (\tilde{Z}_{2n-1})_2 & \cdots & (\tilde{Z}_{2n-1})_m \end{bmatrix},$$

where m is the number of system parameters and

$$(\tilde{Z}_i)_j = \frac{\partial \tilde{Z}_i}{\partial p_j}.$$

Using the method proposed by Vajda (1985), if the \tilde{S} matrix for the proposed model has a full column rank (i.e., $\text{rank}(\tilde{S}) = m$) then the system is

identifiable. In the case of System 3.1 the number of states and parameters are $n = 5$ and $m = 7$, respectively, and x_0 is a vector containing the initial concentrations of the states ([73.8; 0; 26.2; 0; 0]). The column rank of \tilde{S} was equal to m and therefore, System 3.1 is structurally identifiable and all the model parameters have a unique (although possibly small) effect on model predictions. While the identifiability analysis presented in this section relies on the fact that the system model is bilinear, there are identifiability tests which apply to more general non-linear systems (Walter & Pronzato, 1996).

3.4.2 Estimability

If a model is identifiable then every parameter can be estimated using some set of ideal (i.e., sufficiently rich providing enough excitation and noise-free) data. However, identifiability does not guarantee that the model parameters can be estimated from a set of real-world data. A model is said to be estimable if all the model parameters have a statistically significant effect on the model predictions (Ben-Zvi, 2008; Yao et al., 2003). It is possible for an structurally identifiable model parameter to be inestimable. In this section, System 3.1 is tested for estimability. It is shown that, while System 3.1 is identifiable, it is not estimable for the experimental design by Del Bianco et al. (1994).

The set of data used for parameter estimation can be generically represented using a matrix Y given by

$$Y(p) = [y_1|_{t_1} \cdots y_l|_{t_1} \quad y_1|_{t_2} \cdots y_l|_{t_2} \quad \cdots \quad y_1|_{t_h} \cdots y_l|_{t_h}]^T,$$

where l is the number of outputs at each sampling time, and $t_1, t_2, \dots, t_h \in \mathbb{R}^+$ are the time instants. In order to assess the effect of parameter values on model predictions, one can express the model predictions as a Taylor series with respect to the parameters about the nominal parameter values (or, equivalently, about the current best estimate). The Taylor series of Y with respect to p

about p_0 can be written as

$$Y(p) = Y(p_0) + \left. \frac{\partial Y}{\partial p} \right|_{p_0} \cdot (p - p_0) + \frac{1}{2} (p - p_0)^T \cdot \left. \frac{\partial^2 Y}{\partial p^2} \right|_{p_0} \cdot (p - p_0) + \dots \quad (3.3)$$

If $(p - p_0)$ is sufficiently small and $\left. \frac{\partial y}{\partial p} \right|_{p_0}$ exists, the right-hand side of Equation 3.3 can be approximated by

$$Y(p) \cong Y(p_0) + Z|_{p_0} \cdot (p - p_0),$$

in which $p - p_0$ is a vector describing a deviation from the nominal parameter value, and Z is given by

$$Z = \frac{\partial Y}{\partial p} = \begin{bmatrix} \left. \left(\frac{\partial y_1}{\partial p_1} \right) \right|_{t_1} & \dots & \left. \left(\frac{\partial y_1}{\partial p_m} \right) \right|_{t_1} \\ \vdots & & \vdots \\ \left. \left(\frac{\partial y_l}{\partial p_1} \right) \right|_{t_1} & \dots & \left. \left(\frac{\partial y_l}{\partial p_m} \right) \right|_{t_1} \\ \vdots & & \vdots \\ \left. \left(\frac{\partial y_1}{\partial p_1} \right) \right|_{t_h} & \dots & \left. \left(\frac{\partial y_1}{\partial p_m} \right) \right|_{t_h} \\ \vdots & & \vdots \\ \left. \left(\frac{\partial y_l}{\partial p_1} \right) \right|_{t_h} & \dots & \left. \left(\frac{\partial y_l}{\partial p_m} \right) \right|_{t_h} \end{bmatrix}, \quad (3.4)$$

where l is the number of outputs at each sampling time, m is the number of parameters, and $t_1, t_2, \dots, t_h \in \mathbb{R}^+$ are the time instants. In order to determine which directors in the parameter space will have the greatest effect on the vector of observations, Y , Singular Value Decomposition (SVD) is performed on Z at p_0 .

$$Z_{lh \times m} = U_{lh \times lh} S_{lh \times m} V_{m \times m}^T. \quad (3.5)$$

U is a unitary matrix consisting of eigenvectors $(u_1, u_2, \dots, u_{lh})$ of ZZ^T as columns. V is also a unitary matrix consisting of eigenvectors (v_1, v_2, \dots, v_m) of $Z^T Z$ as columns. S is a diagonal matrix with non-negative singular values

on diagonal, which are sorted in descending order (Golub & Reinsch, 1970).

$$S = \begin{bmatrix} s_1 & 0 & \cdots & 0 \\ 0 & s_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & s_m \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix},$$

where s_i is the singular value corresponding to vector v_i ($i \in \{1, 2, \dots, m\}$). Now, using SVD decomposition of matrix Z , the Taylor expansion in Equation 3.3 can be written as

$$Y(p) = Y(p_0) + USV^T(p - p_0) \quad (3.6)$$

$$= Y(p_0) + U \begin{bmatrix} s_1 v_1^T(p - p_0) \\ \vdots \\ s_m v_m^T(p - p_0) \end{bmatrix}.$$

U as a unitary matrix preserves the vector length. With s_1 being the largest singular value, $v_1^T(p - p_0)$ has the largest impact on the value of $Y(p)$. Similarly, s_m is the smallest singular value and $v_m^T(p - p_0)$ has the smallest effect on the value of $Y(p)$.

Each of the quantities $v_1^T(p - p_0)$ to $v_m^T(p - p_0)$ represent linear combinations of process parameters. The values of s_1 to s_m allow each of these linear combinations to be ranked in terms of their effect on the value of $Y(p)$. For example, $v_1^T(p - p_0)$ is deemed to have the largest effect on the model prediction $Y(p)$, while $v_m^T(p - p_0)$ is deemed to have the smallest. It should be noted, however, that these conclusions only hold about the nominal parameter value p_0 .

This ranking of parameter combinations can be further quantified by examining the value of s_i for $i = 1, 2, \dots, m$. Specifically, if $s_j/s_i \rightarrow 0$, then for some $i, j \in \{1, 2, \dots, m\}$ the effect of $v_j^T(p - p_0)$ may be numerically negligible.

For System 3.1 (using the nominal parameter values in Table 3.4) the matrix Z was chosen so that model prediction times matched the literature data listed in Table 3.3. Also, the experimental design chosen was the one

by Del Bianco et al. (1994). Note that the derivatives of the measurements with respect to the parameters at the initial time was assumed to be zero (as the initial conditions are known and fixed), and the derivatives of the measurements with respect to S_L was zero until such time as $f([A_{ex}^*])$ was equal to one. Singular Value Decomposition (SVD) was performed on Z , and the values for s_1 to s_7 were computed and plotted in Figure 3.1 as a column graph showing the relative singular values (i.e., $\{s_1/s_1, s_2/s_1, \dots, s_7/s_1\}$). As can be seen from Figure 3.1, some singular values are much larger than the others (e.g. $s_1/s_1 = 1.0000$ and $s_2/s_1 = 0.0164$ comparing to $s_7/s_1 = 0.0000$), which means not all the linear combinations of parameters affect the model predictions significantly, and therefore, not all the parameters are estimable using the experimental design by Del Bianco et al. (1994). In the next section the issue of how many (and which) parameter combinations can be estimated using the data set by Del Bianco et al. (1994) is addressed.

Table 3.4: Nominal values for parameters

Parameter	Value reported in literature	Nominal value	Unit	Reference
k_P	0.013 at T=400 °C and atmospheric pressure	0.08	min^{-1}	Wiehe (1993)
k_A	0.015 ± 0.03 for Athabasca asphaltenes	0.18	min^{-1}	Rahmani et al. (2002)
k_H	$k'[donor]/P_{H_2}$ $= (0.23 \pm 0.09) \times 0.75/20$ $= 0.0087 \pm 0.0037$	0.0024	$min^{-1}.MPa^{-1}$	Rahmani et al. (2002)
k_C	0.028 ± 0.008 for Athabasca asphaltenes	0.02	min^{-1}	Rahmani et al. (2002)
m'	$m + n = 0.54 \pm 0.03$	0.55	—	Rahmani et al. (2002)
a	0.221	0.20	—	Wiehe (1993)
S_L	0.49 for full residue	0.15	wt%/wt%	Wiehe (1993)

3.5 Parameter estimation

The estimability analysis showed that not all the parameters could be estimated for Del Bianco et al.'s experimental design. However, the experimental data provided by Del Bianco et al. (1994) and listed in Table 3.3 can be used to estimate some parameters or combinations of parameters. The parameter combinations $v_i^T(p - p_0)$ with $i = 1, 2, \dots, 7$, computed and ranked in Section

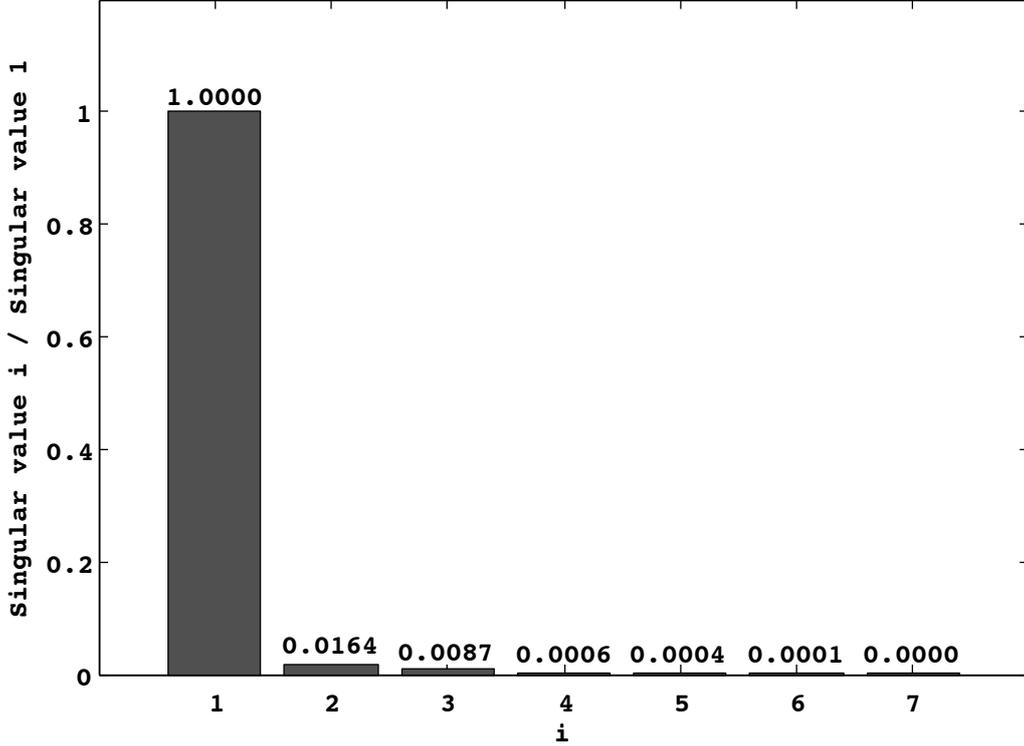


Figure 3.1: Singular values comparison

3.4.2, can be viewed as pseudo-parameters using the invertible matrix V defined in Equation 3.5. Specifically, let k denote the set of pseudo-parameters defined by

$$k = V^T p,$$

so that $k_i = v_i^T p$ and $p = V k$. Using this transformation Equation 3.6 can be written in terms of k as

$$Y(p) = Y(V k_0) + U \begin{bmatrix} s_1(k_1 - k_{0,1}) \\ \vdots \\ s_m(k_m - k_{0,m}) \end{bmatrix}$$

with $k_0 = [k_{0,1}, k_{0,2}, \dots, k_{0,m}] = V^T p_0$. As U is a unitary matrix, it preserves the vector length. Therefore, k_1 is the pseudo-parameter (i.e., linear combination of p) with the most significant effect on the model predictions, because s_1 is the largest singular value, and k_m is the least important pseudo-parameter, because s_m is the smallest singular value.

The question of how many pseudo-parameters are to be estimated can now be addressed by comparing the effect of each pseudo-parameter on the value of the objective function which defines the parameter estimation problem. Specifically, let the objective function for parameter estimation problem be defined as

$$\hat{p} = \arg \min_{p_1, \dots, p_m} (Y_{exp} - Y(p))^T W (Y_{exp} - Y(p)), \quad (3.7)$$

subject to

$$0 \leq p_i \leq 1 \quad (i \in \{1, 2, \dots, m\}),$$

where Y_{exp} consists of all the experimental outputs, W is a strictly positive weighting matrix, and m is the number of parameters. The lower limits for all the parameters were chosen as 0, because none of them can have a negative value. The upper limits for k_p , k_A , k_H , k_c , and S_L (p_1, p_2, p_3, p_4, p_7) were chosen as 1, because of the order of magnitudes of their values reported in literature (Table 3.4). The upper limits for m' and a (p_5, p_6) were chosen as 1, because they encode fraction values and cannot be larger than 1.

The objective function in Equation 3.7 can be written in terms of the pseudo-parameters as follows.

$$\hat{k} = \arg \min_{k_1, \dots, k_g} (Y_{exp} - Y(k))^T W (Y_{exp} - Y(k)),$$

subject to

$$0 \leq \tilde{v}_i k \leq 1 \quad (i \in \{1, 2, \dots, m\}),$$

where \tilde{v}_i is the i^{th} row of V , and g , with $0 \leq g \leq m$, is a positive integer that encodes the number of pseudo-parameters to be estimated. As the pseudo-parameters are sorted based on their importance, one can estimate the first g pseudo-parameters and fix the rest at their nominal values. To determine the number of pseudo-parameters necessary to obtain a good fit to literature data (Del Bianco et al., 1994) one can compare the objective function at each value of g . For System 3.1, with data from Table 3.3, the optimization problem

was solved, in turn, with g values of one to seven. The weighting matrix was chosen as

$$W_{lh \times lh} = \text{diag}(w_1, w_2, \dots, w_l, w_1, w_2, \dots, w_l, \dots, \dots, w_l),$$

where

$$w_i = \frac{r_i}{\|y_{i,exp}\|} \quad (i = 1, 2).$$

w_i was chosen to normalize the weight assigned to each output. The value of r_i is used to make sure that the problem is numerically tractable (in case $\|y_{i,exp}\|$ is very large for some of the outputs). In this work, r_i was chosen as $lh = 12$ for $i \in \{1, 2\}$.

The MATLAB optimization function *patternsearch* was first used to solve the estimation problem, and the nominal values in Table 3.4 were chosen as initial guesses for each parameter. Thereafter, the optimization answer was chosen as an initial guess for MATLAB *fmincon* function, and a final optimum was obtained after using *fmincon* function once or more. Due to the fact that the Taylor series expansion in Equation 3.3 is valid only about some nominal parameter value, the mapping V was recomputed after a specific number of optimizer iterations in order to reflect the updated parameter estimates. The minimum of the objective function obtained for the different number of pseudo-parameters (i.e. $g \in \{1, 2, \dots, 7\}$) is shown in Figure 3.2. As can be seen from Figure 3.2, a value of $g = 3$ provides the best tradeoff between having a low objective function value (hence a good fit of data) and model parsimony. The fit of the model using only three pseudo-paramters is compared with experimental data (from Table 3.3) in Figure 3.3, and the pseudo-parameter estimates along with their confidence regions are listed in Table 3.5. Estimates for original system parameters based on the transformation $V : k \mapsto p$ are shown in Table 3.6.

As shown in Figure 3.3, it is possible to obtain good fit of experimental data using only three pseudo-parameters. In addition to obtaining point-estimates it is also possible to obtain approximate confidence regions for these three pseudo-parameters as well as for the predicted response. The approximate

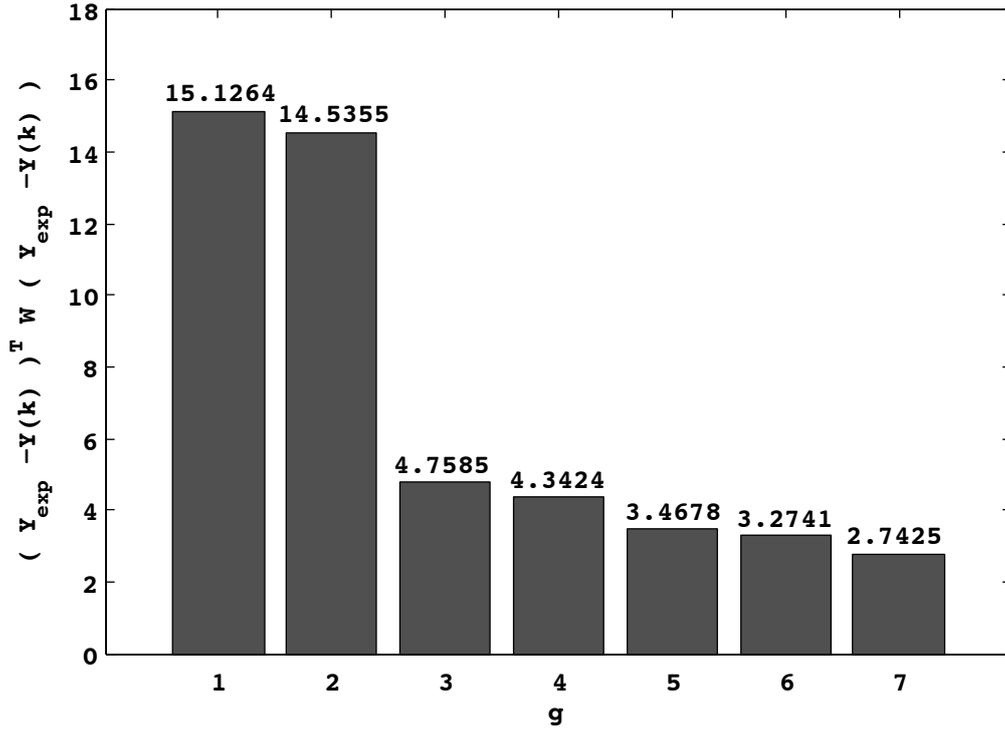


Figure 3.2: Error comparisons for optimization problems with different g

confidence interval for each pseudo-parameter is (Marsili-Libelli et al., 2003)

$$\delta_i = \pm t_{N-n_k}^{1-(\alpha/2)} \sqrt{C_{ii}},$$

where N is the number of experimental data points, n_k is the number of

Table 3.5: Confidence intervals for pseudo-parameters

i	k_i	δ_i
1	-0.0006	± 0.0004
2	-0.0276	± 0.0117
3	0.0744	± 0.0102
4	0.0982	N/A
5	-0.5330	N/A
6	0.2304	N/A
7	-0.1608	N/A

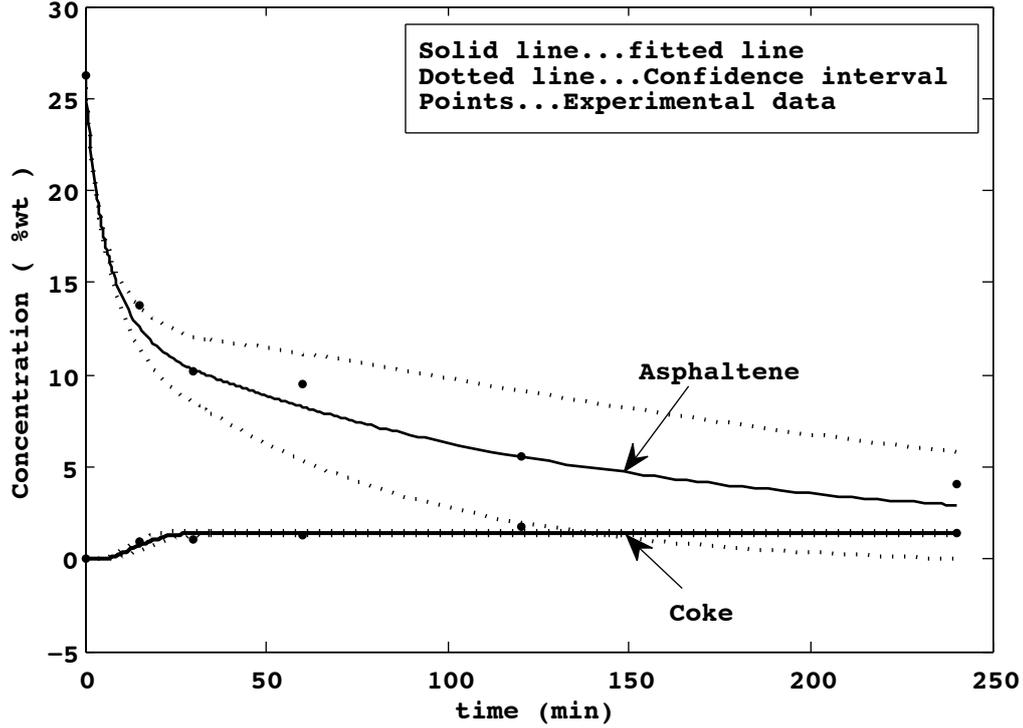


Figure 3.3: Model fit for $g = 3$ and confidence bounds for mean responses

pseudo-parameters, α is the confidence level, and

$$C(\hat{k}) = \frac{2}{N - n_k} E(\hat{k}) H(\hat{k})^{-1},$$

$$H(\hat{k}) = \left. \frac{\partial^2 E(k)}{\partial k \partial k^T} \right|_{\hat{k}},$$

$$E(\hat{k}) = \sum_{i=1}^N (y_i - \hat{y}_i)^T w_i (y_i - \hat{y}_i).$$

In the case where only three pseudo-parameters are estimated, the Hessian matrix was calculated with respect to k_1, k_2, k_3 . The results for pseudo-parameters approximate confidence intervals are shown in Table 3.5.

Based on the formula suggested by Seber & Wild (1989) for confidence intervals of model predictions, the approximate confidence intervals for the mean responses were obtained by using the following equation

Table 3.6: Parameter estimations for $g = 3$

NO.	Parameter	Estimate for data by Del Bianco et al. (1994)
1	k_p	0.0034
2	k_A	0.2278
3	k_H	0.0006
4	k_C	0.0480
5	m'	0.5306
6	a	0.1730
7	S_L	0.1149

$$\epsilon = \pm t_{N-n_k}^{1-(\alpha/2)} \left[\left(\frac{\partial y}{\partial [k_1, \dots, k_3]} \right) C(\hat{k}) \left(\frac{\partial y}{\partial [k_1, \dots, k_3]} \right)^T \right]^{1/2}.$$

The approximate 95% confidence limits for the predicted response are shown in Figure 3.3. It should be noted that the confidence intervals computed in this work are only approximate due to the fact that the parameter estimation problem is non-linear.

3.6 Optimal experimental design

As shown in Section 3.4.2 the measured data obtained by Del Bianco et al. (1994) can be reproduced using only three independent parameters. However, according to the identifiability analysis *every* parameter has some unique effect on the model predictions. This implies that there exists an alternative experiment or operating condition which could provide measured data that cannot be fitted using only three pseudo-parameters. More importantly, a process designed based on estimates obtained using only the data in Del Bianco et al. (1994) and operated under a different set of conditions may not behave as predicted and may therefore not be able to meet design specifications.

In this section, methodology for the computation of an optimal experimental design for the estimation of *all* system parameters is proposed. A feature of the proposed approach is that the new experiments are designed to complement existing data in the sense that new experiments are chosen so that the *combined* data set is optimal. The optimality criteria used in this work is D-optimality.

For any single experiment, one can define a scaled sensitivity matrix, Z_{scaled} , which is a matrix containing the scaled sensitivity coefficients of outputs with respect to parameters at different time instants.

$$Z_{\text{scaled}} = \begin{bmatrix} \left. \begin{array}{cc} \frac{hp_1}{\|y_1\|} & \frac{\partial y_1}{\partial p_1} \\ \frac{hp_2}{\|y_2\|} & \frac{\partial y_2}{\partial p_1} \end{array} \right|_{t_1} & \left. \begin{array}{cc} \frac{hp_2}{\|y_1\|} & \frac{\partial y_1}{\partial p_2} \\ \frac{hp_2}{\|y_2\|} & \frac{\partial y_2}{\partial p_2} \end{array} \right|_{t_1} & \cdots & \left. \begin{array}{cc} \frac{hp_7}{\|y_1\|} & \frac{\partial y_1}{\partial p_7} \\ \frac{hp_7}{\|y_2\|} & \frac{\partial y_2}{\partial p_7} \end{array} \right|_{t_1} \\ \left. \begin{array}{cc} \frac{hp_1}{\|y_1\|} & \frac{\partial y_1}{\partial p_1} \\ \frac{hp_2}{\|y_2\|} & \frac{\partial y_2}{\partial p_1} \end{array} \right|_{t_2} & \left. \begin{array}{cc} \frac{hp_2}{\|y_1\|} & \frac{\partial y_1}{\partial p_2} \\ \frac{hp_2}{\|y_2\|} & \frac{\partial y_2}{\partial p_2} \end{array} \right|_{t_2} & \cdots & \left. \begin{array}{cc} \frac{hp_7}{\|y_1\|} & \frac{\partial y_1}{\partial p_7} \\ \frac{hp_7}{\|y_2\|} & \frac{\partial y_2}{\partial p_7} \end{array} \right|_{t_2} \\ \vdots & \vdots & \vdots & \vdots \\ \left. \begin{array}{cc} \frac{hp_1}{\|y_1\|} & \frac{\partial y_1}{\partial p_1} \\ \frac{hp_2}{\|y_2\|} & \frac{\partial y_2}{\partial p_1} \end{array} \right|_{t_h} & \left. \begin{array}{cc} \frac{hp_2}{\|y_1\|} & \frac{\partial y_1}{\partial p_2} \\ \frac{hp_2}{\|y_2\|} & \frac{\partial y_2}{\partial p_2} \end{array} \right|_{t_h} & \cdots & \left. \begin{array}{cc} \frac{hp_7}{\|y_1\|} & \frac{\partial y_1}{\partial p_7} \\ \frac{hp_7}{\|y_2\|} & \frac{\partial y_2}{\partial p_7} \end{array} \right|_{t_h} \end{bmatrix},$$

where h is the number of samples, and $\|y_i\|$ ($i \in \{1, 2\}$) is

$$\|y_i\| = \sqrt{y_i^2|_{t_1} + y_i^2|_{t_2} + \cdots + y_i^2|_{t_h}}.$$

If $\|y_i\| = 0$ for an experiment, it means that y_i is zero at any time instant. For this situation, rows of Z_{scaled} containing y_i are put as all zero.

The notation of Z_{scaled} can be extended to a design with multiple experiments. For a q -experiment design, one may define a matrix \bar{Z} which contains the scaled sensitivity matrix from q experiments.

$$\bar{Z} = \begin{bmatrix} Z_{\text{scaled},0} \\ Z_{\text{scaled},1} \\ \vdots \\ Z_{\text{scaled},q} \end{bmatrix}, \quad (3.8)$$

where $Z_{\text{scaled},j}$ ($j \in \{1, 2, \dots, q\}$) is Z_{scaled} computed for j th design, and $Z_{\text{scaled},0}$ is Z_{scaled} computed for already available data.

The optimality criteria used in this work is D-optimality which implies that for an optimal design $|\det(\bar{Z}^T \bar{Z})|$ is maximized. For the system under consideration in this work there are two optimization variables: a) initial concentrations, b) hydrogen partial pressure. One can also optimize over sampling

time but in this work, it is fixed at a small practical value, 10 min. The existing constraints for the optimization variables are:

1. Initial concentrations ($[A]_0, [THFI]_0$): Maximum asphaltene initial concentration ($[A]_0$) was assumed to be less than 32 wt% ($[A]_0 = 32.7 \text{ wt\%}$ belongs to Athabasca bitumen vacuum bottoms (Rahimi et al., 1999)). Initial coke content ($[THFI]_0$) was fixed at zero.
2. Hydrogen partial pressure (P_{H_2}): the hydrogen pressure was kept constant during each experiment, and the range for it was chosen as 8-20 MPa according to pressures for similar experiments done by Del Bianco et al. (1994) and Panariti et al. (2000).

Now, the experimental design problem can be defined as

$$[[A]_0 \quad P_{H_2}]^* = \arg \max_{[A]_0, P_{H_2}} |\det(\bar{Z}^T \bar{Z})|, \quad (3.9)$$

subject to

$$0 < [A]_0 < 32,$$

$$8 \leq P_{H_2} \leq 20,$$

where the * notation in the left hand side of Equation 3.9 denote optimal value for the optimization variables. Solving Equation 3.9 for different number of experiments (i.e., the value of q in Equation 3.8) allows the trade off between experimental effort and quality of model to be quantified. For case of System 3.1, enforcing the constraints and using the parameter estimates in Table 3.6, the optimal configurations for one, two and three *new* experiments are listed in Table 3.7.

In order to choose the appropriate number of additional experiments, the parameter estimates obtained from the different experimental designs listed in Table 3.7 must be compared. This was done by generating a data set (via simulation) using the parameter values from Table 3.6 for each of the experimental designs in Table 3.7. Next, parameters in Table 3.6 were back-calculated using the combined data set containing the simulated data and the

Table 3.7: Optimal configurations for different experimental designs

Design	Experiment Number	[P], wt%	[A], wt%	[THFI], wt%	T_s , min	P_{H_2} , MPa
Current	1	73.8	26.2	0	Variant ^a	20
One-experiment	1	73.5	26.5	0	10	8
	2	68.0	32.0	0	10	19.9
Two-experiment	1	68.0	32.0	0	10	19.9
	2	68.4	31.6	0	10	8
	3	100.0	0.0	0	10	20

^a Current data is available at sampling times of 15, 30, 60, 120 and 240 min

data in Table 3.3. The objective function in Equation 3.7 was used with W chosen as an Identity matrix. MATLAB *patternsearch* and *fmincon* functions were used for optimization. The results for parameter back-calculations as well as the percent error for each parameter are listed in Table 3.8.

Table 3.8: Parameter back-calculation for different designs

Parameter		k_P	k_A	k	k_H	m'	a	S_L
Nominal value		0.0034	0.2278	0.0006	0.0480	0.5306	0.1730	0.1149
Current	Estimated value	0.0004	0.1410	0.0005	0.8721	0.5229	1.0000	0.1209
	%error	87.4	38.1	12.8	1716.5	1.5	477.9	5.2
One-experiment	Estimated value	0.0032	0.2044	0.0004	0.0494	0.5332	0.1273	0.1150
	% error	5.1	10.3	20.3	3.0	0.5	26.4	0.1
Two-experiment	Estimated value	0.0033	0.2143	0.0006	0.0497	0.5297	0.1708	0.1151
	% error	0.4	5.9	1.8	3.5	0.2	1.3	0.2
Three-experiment	Estimated value	0.0034	0.2156	0.0006	0.0505	0.5297	0.1692	0.1156
	%error	1.6	5.4	1.6	5.2	0.2	2.2	0.6

The different experimental designs listed in Table 3.7 were compared using several benchmarks. Firstly, the percent error for each parameter estimate was compared. As shown in Figure 3.4 there is a marked decrease in the percent estimation error for the parameters with the two (additional) experimental design as opposed to the single (additional) experimental design. However, the three design do not provide a large improvement in the quality of the parameter estimates over the two experiment design.

To get a quantitative assessment of the different experimental design, Three numerical benchmarks were chosen. The benchmarks used are the total area

in the approximate confidence ellipsoids for the parameter estimates (A_t), $|\det(\bar{Z}^T \bar{Z})|$, and the norm of error for the parameters (E_t) for each experimental design. The values for each benchmark and design are listed in Table 3.9. The confidence ellipsoids used to compute A_t were obtained based on the method proposed in Marsili-Libelli et al. (2003), and the areas under ellipsoids for each pair of parameters were estimated using normalized axis (i.e., where the nominal parameter value is equal to unity). The values of A_t for the case where zero and one additional experiments were performed is listed as ∞ , which means that the confidence intervals were in hyperboloidal shape. This hyperbolic shape of the confidence intervals supports the proposition that the parameters are inestimable using only the data provided in Del Bianco et al. (1994) and further suggests that the parameters are likely to be inestimable even in the case where one additional experiment is performed. This, in turn, implies that the experimental design with two additional runs is the smallest design (in terms of number of experiments) for which one can expect reliable parameter estimates.

Table 3.9: Properties of different designs

Design	Total area of confidence ellipsoids (A_t)	$ \det(\bar{Z}^T \bar{Z}) $	$E_t = \sqrt{\sum_{i=1}^7 (p_i - \hat{p}_i)^2}$
Current	∞	9.1369E-17	1.1707
One-experiment	∞	0.0002	0.0515
Two-experiment	0.0929	0.0497	0.0139
Three-experiment	0.0461	1.4465	0.0131

As expected, the value of A_t decreases as the number of experimental runs increases from two-experiment to three-experiment design. The value of E_t also decreases from current to three-experiment design. It should be noted that some of the individual estimation errors do not improve (Figure 3.4). For example, going from two-experiment design to three-experiment design, percent error for k_P increases.

Overall, the two experiment design is the smallest design which gives reliable parameter estimates and furthermore, increasing the number of additional

experiments from two to three does not result in a large improvement in parameter estimates or their confidence regions. As a result, the two-additional experiments design is the best one for System 3.1 given existing data (Table 3.3) and the current parameter estimates (Table 3.6).

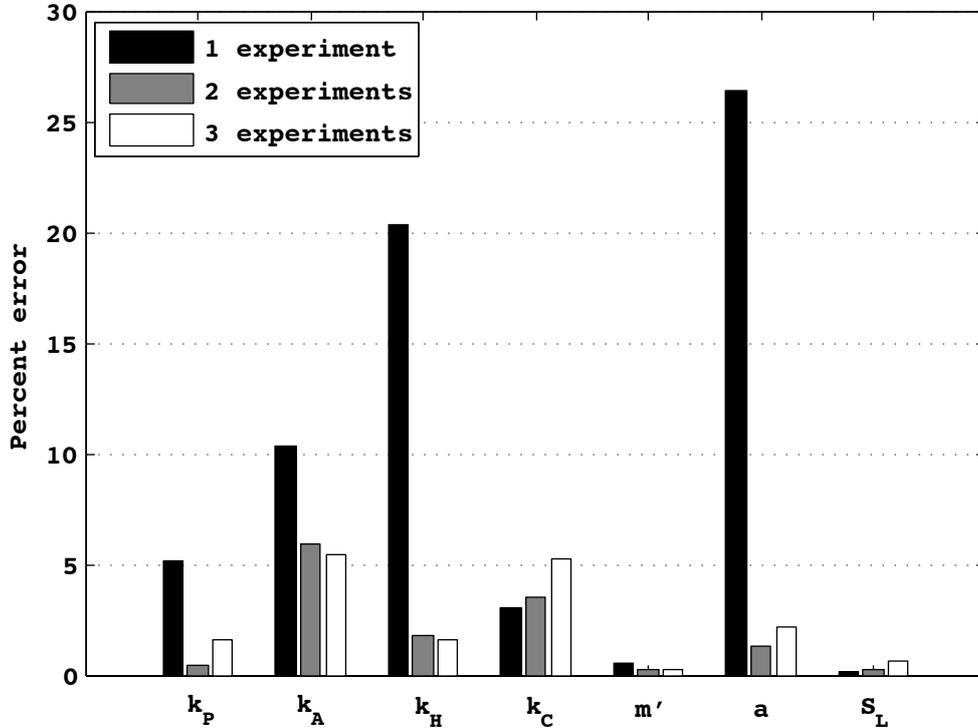


Figure 3.4: Error comparison of parameter estimation for different designs

3.7 Conclusion

A lumped kinetic model for hydroconversion processing of vacuum residue is proposed. The model is shown to be identifiable. However, subsequent sensitivity analysis revealed that not all model parameters can be estimated from literature data. It is shown that the model fit depends critically on only three pseudo-parameters. The ability to obtain good predictions using only three pseudo-parameters was validated using literature data. Confidence regions for the pseudo-parameters and model responses were also computed. A methodology for the design of new experiments which complement existing data is

presented. Several sets of optimal experiments were designed and evaluated. It is shown that a design with two additional experiments provides an ideal balance between the accuracy of parameter estimates and experimental effort.

The proposed sensitivity analysis and re-parameterization approach is quite general and can be applied to any system for which the dependence of the outputs on the parameters is differentiable or can be approximated using a finite difference. As a result, the proposed method could be applied to a system operated non-isothermally, as well as to some model formulations where additional parameters are added to characterize differences in feed or in catalyst activity. While the identifiability analysis presented in this work relied on the fact that the system model is bilinear, there are identifiability tests which apply to more general non-linear systems.

3.8 Acknowledgements

Financial support from UOP is gratefully acknowledged.

Bibliography

- [1] Ben-Zvi A, McAuley K, McLellan J. Identifiability study of a liquid-liquid phase-transfer catalyzed reaction system. *AICHE Journal*. 2004;50(10):2493-501.
- [2] Ben-Zvi, A. Reparameterization of inestimable systems with applications to chemical and biochemical reactor systems. *AICHE Journal*. 2008;54(5):1270-1281.
- [3] Carbonell MM, Guirardello R. In: Modelling of a slurry bubble column reactor applied to the hydroconversion of heavy oils. Proceedings of the 1997 3rd international conference on gas-liquid-solid reactor engineering, december 7, 1997 - december 10; 1997; Shonan, Jpn: Elsevier Science Ltd; 1997. p. 4179-85.
- [4] De Almeida RM, Guirardello R. Hydroconversion kinetics of marlim vacuum residue. *Catalysis Today*. 2005;109(1-4):104-11.
- [5] Del Bianco A, Panariti N, Carlo D, Beltrame PL, Carniti P. New development in deep hydroconversion of heavy oil residues with dispersed catalysts. 2. kinetic aspects of reaction. *Energy and Fuels*. 1994;8(3):593-7.
- [6] Golub G, Reinsch C. Singular value decomposition and least square solutions. *Numerische Mathematik*. 1970;14(5):403-20.
- [7] Jayasankar BR, Ben-Zvi A, Huang B. Identifiability and estimability study for a dynamic solid oxide fuel cell model. *Computers and Chemical Engineering*. 2009;33(2):484-92.

- [8] Marsili-Libelli S, Guerrizio S, Checchi N. Confidence regions of estimated parameters for ecological systems. *Ecological Modelling*. 2003;165(2-3):127-46.
- [9] Panariti N, Del Bianco A, Del Piero G, Marchionna M, Carniti P. Petroleum residue upgrading with dispersed catalysts. part 2. effect of operating conditions. *Applied Catalysis A: General*. 2000;204(2):215-22.
- [10] Qader SA, Hill GR. Catalytic hydrocracking. mechanism of hydrocracking of low temperature coal tar. *Industrial and Engineering Chemistry Process Design and Development*. 1969;8(4):456-61.
- [11] Qader SA, Wisler WH, Hill GR. Rate data of tar hydrocracking. *Fuel*. 1972;51(1):54-6.
- [12] Rahimi P, Gentzis T, Kubo J, Fairbridge C, Khulbe C. Coking propensity of athabasca bitumen vacuum bottoms in the presence of H-donors—formation and dissolution of mesophase from a hydrotreated petroleum stream (H-donor). *Fuel Processing Technology*. 1999 7;60(2):157-70.
- [13] Rahmani S, McCaffrey W, Gray MR. Kinetics of solvent interactions with asphaltenes during coke formation. *Energy and Fuels*. 2002;16(1):148-54.
- [14] Rana MS, Samano V, Ancheyta J, Diaz JAI. A review of recent advances on process technologies for upgrading of heavy oils and residua. *Fuel*. 2007;86(9):1216-31.
- [15] Seber GAF, Wild CJ. *Nonlinear regression*. USA: Wiley; 1989. p. 191-3.
- [16] Trejo F, Ancheyta J. Kinetics of asphaltenes conversion during hydrotreating of maya crude. *Catalysis Today*. 2005;109(1-4):99-103.
- [17] Vajda S. In: *Structural identifiability of linear, bilinear, polynomial and rational systems*. Volume 6: Biomedical applications, water resources, environment, energy systems, development, social effects, SWIIS; Education; Budapest, Hung: IFAC by Pergamon Press; 1985. p. 717-22.

- [18] Vanrolleghem PA, Van Daele M, Dochain D. Practical identifiability of a biokinetic model of activated sludge respiration. *Water Research*. 1995;29(11):2561-61.
- [19] Walter E, Pronzato L. On the identifiability and distinguishability of non-linear parametric models. *Mathematics and Computers in Simulation*. 1996;42(2-3):125-34.
- [20] Wiehe IA. In: Phase separation kinetic model for coke formation. Symposium on resid upgrading presented at the 205th national meeting of the american chemical society, march 28, 1993 - april 2; 1993; Denver, CO, USA: Published by ACS; 1993. p. 428-33.
- [21] Yao KZ, Shaw BM, Kou B, McAuley KB, Bacon DW. Modeling ethylene/butene copolymerization with multi-site catalysts: Parameter estimability and experimental design. *Polymer Reaction Engineering*. 2003;11(3):563-88.

Chapter 4

Conclusions and future work

4.1 Conclusions

Nowadays, converting vacuum residue (boiling point above 524 °C) into distillates and improving its economic value is of particular interest (Gray & McCaffrey, 2002). In order to produce distillates from and upgrade heavy feedstocks, some of their properties such as H/C ratio have to be improved (Del Bianco et al., 1993). H/C ratio can be increased by either carbon rejection such as in coking or hydrogen addition such as in hydroconversion (Del Bianco et al., 1993). Carbon rejection processes have several disadvantages (Del Bianco et al., 1993) : “low distillate conversion with poor quality (low CCR reduction and poor heteroatoms removal and large volumes of high sulphur, high metals coke). Therefore, hydrogen addition processes like hydroconversion have attracted a lot of attention.

A model for hydroconversion processing of vacuum residue is proposed. This model contains five states, two outputs and seven parameters, and is expressed in five ordinary differential equations (ODEs). The proposed model is developed using lumping methodology and first-principle conservation equations. The components participating in the reaction are lumped according to their solubility characteristics. After model proposal, it is checked for identifiability using the method by Vajda (1985) for bilinear systems, and proved to be identifiable. Then, it is checked for estimability performing the singular value decomposition (SVD) on the parameter covariance matrix. The result was that not all the model parameters are estimable using the experimental

design by Del Binaco et al..

The proposed model is reparameterized using the results of SVD of the parameter covariance matrix, and seven pseudo-parameters are obtained, which are the linear combinations of the former parameters. The advantage of the new set of parameters is that they are sorted based on their significance in changing the output predictions. The model then is fitted to the data by Del Bianco et al. (1994) estimating only the three most important pseudo-parameters and fixing the rest at their nominal values, which confirms that not all the parameters are estimable for the data by Del Bianco et al..

The fact that only three pseudo-parameters are estimated will concern one about the design capabilities of the model as the other four pseudo-parameters can become more important in a different experiment. Therefore, a set of additional experiments has to be designed to collect enough data for estimating all seven parameters. This is accomplished using the D-optimality criteria in this work, and different sets of experiments are designed and compared. The designs being compared are one, two and three additional experiments. Comparing the effort and the different properties for these designs such as norm of the error, two-experiment design is chosen as the best experimental design in this work.

4.2 Future work

To better understand the dynamics of hydroconversion, and improve the quality of its products, suppress coke formation, and reduce the catalyst deactivation, the following items are suggested for future work:

1. Obtain additional measurements running the optimal experiments designed in this work.
2. Check the model estimability for the data obtained, and try to estimate all the parameters.
3. The model should be expanded in order to include the effect of changes in temperature, catalyst concentration and type.

4. The estimability of the expanded model should be examined.
5. New experiments should be designed for estimation of all (new and old) parameters of the expanded model.
6. A model-based predictive control should be designed so that the maximum desired products (distillates) are produced and coke formation is minimized.

Bibliography

- [1] Del Bianco A, Panariti N, Carlo D, Beltrame PL, Carniti P. New development in deep hydroconversion of heavy oil residues with dispersed catalysts. 2. kinetic aspects of reaction. *Energy and Fuels*. 1994;8(3):593-7.
- [2] Del Bianco A, Panariti N, Di Carlo S, Elmouchnino J, Fixari B, Le Perchec P. Thermocatalytic hydroconversion of heavy petroleum cuts with dispersed catalyst. *Applied Catalysis A: General*. 1993;94(1):1-16.
- [3] Gray MR, McCaffrey WC. Role of chain reactions and olefin formation in cracking, hydroconversion, and coking of petroleum and bitumen fractions. *Energy and Fuels*. 2002;16(3):756-66.
- [4] Vajda S. In: *Structural identifiability of linear, bilinear, polynomial and rational systems*. Volume 6: Biomedical applications, water resources, environment, energy systems, development, social effects, SWIIS; Education.; Budapest, Hung: IFAC by Pergamon Press; 1985. p. 717-22.

Appendix A

Maple codes

Note 1: the entries that are numbered are maple outputs, and the ones that are not numbered are inserted by the user.

Note 2: Pp, Ps, Ap, As are P^+, P^*, A^+, A^* respectively. k_1, k_2, k_3, k_4, m and s are k_P, k_A, k_H, k_C, m' and S_L respectively.

$$\begin{aligned}
& \left[\begin{array}{l}
> A := \begin{bmatrix} -k_1 & 0 & 0 & 0 & 0 \\ (1-a) \cdot k_1 & 0 & (1-m) \cdot k_2 & 0 & 0 \\ 0 & 0 & -k_2 & 0 & 0 \\ a \cdot k_1 + s \cdot k_4 \cdot f & s \cdot k_4 \cdot f & m \cdot k_2 & -k_4 \cdot f & 0 \\ -s \cdot k_4 \cdot f & -s \cdot k_4 \cdot f & 0 & k_4 \cdot f & 0 \end{bmatrix} \\
> x := \begin{bmatrix} Pp \\ Ps \\ Ap \\ As \\ THFI \end{bmatrix} \\
> B := \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & k_3 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -k_3 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \\
> C := \begin{bmatrix} 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \\
> dx := A \cdot x + Hy \cdot B \cdot x \\
dx := \begin{bmatrix} -k_1 Pp \\ (1-a) k_1 Pp + (1-m) k_2 Ap + Hy k_3 As \\ -k_2 Ap \\ (a k_1 + s k_4 f) Pp + s k_4 f Ps + m k_2 Ap - k_4 f As - Hy k_3 As \\ -s k_4 f Pp - s k_4 f Ps + k_4 f As \end{bmatrix} \tag{1} \\
> y := C \cdot x \\
y := \begin{bmatrix} Ap + As \\ THFI \end{bmatrix} \tag{2}
\end{array} \right.$$

```
> p := [ k1
         k2
         k3
         k4
         m
         a
         s ]
```

```
> Q0 := C
```

$$Q_0 := \begin{bmatrix} 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

(3)

```
> for l from 1 to 9 do Q_l := (Q_{l-1}, A; Q_{l-1}, B) end do
```

$$Q_1 := \begin{bmatrix} a k_1 + s k_4 f & s k_4 f & -k_2 + m k_2 & -k_4 f & 0 \\ -s k_4 f & -s k_4 f & 0 & k_4 f & 0 \\ 0 & 0 & 0 & -k_3 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$Q_2 := [[-(a k_1 + s k_4 f) k_1 + s k_4 f (1 - a) k_1 - k_4 f (a k_1 + s k_4 f), -k_4^2 f^2 s, s k_4 f (1 - m) k_2 - (-k_2 + m k_2) k_2 - k_4 f m k_2, k_4^2 f^2, 0],$$

$$[s k_4 f k_1 - s k_4 f (1 - a) k_1 + k_4 f (a k_1 + s k_4 f), k_4^2 f^2 s, -s k_4 f (1 - m) k_2 + k_4 f m k_2, -k_4^2 f^2, 0],$$

$$[-k_3 (a k_1 + s k_4 f), -k_3 s k_4 f, -k_3 m k_2, k_3 k_4 f, 0],$$

$$[0, 0, 0, 0, 0],$$

$$[0, 0, 0, k_3 s k_4 f + k_3 k_4 f, 0],$$

$$[0, 0, 0, -k_3 s k_4 f - k_3 k_4 f, 0],$$

$$[0, 0, 0, k_3^2, 0],$$

$$[0, 0, 0, 0, 0]]$$

$$Q_3 := \begin{bmatrix} 16 \times 5 \text{ Matrix} \\ \text{Data Type: anything} \\ \text{Storage: rectangular} \\ \text{Order: Fortran_order} \end{bmatrix}$$

$$Q_4 := \begin{bmatrix} 32 \times 5 \text{ Matrix} \\ \text{Data Type: anything} \\ \text{Storage: rectangular} \\ \text{Order: Fortran_order} \end{bmatrix}$$

$$Q_5 := \begin{bmatrix} 64 \times 5 \text{ Matrix} \\ \text{Data Type: anything} \\ \text{Storage: rectangular} \\ \text{Order: Fortran_order} \end{bmatrix}$$

$$Q_6 := \begin{bmatrix} 128 \times 5 \text{ Matrix} \\ \text{Data Type: anything} \\ \text{Storage: rectangular} \\ \text{Order: Fortran_order} \end{bmatrix}$$

$$Q_7 := \begin{bmatrix} 256 \times 5 \text{ Matrix} \\ \text{Data Type: anything} \\ \text{Storage: rectangular} \\ \text{Order: Fortran_order} \end{bmatrix}$$

$$Q_8 := \begin{bmatrix} 512 \times 5 \text{ Matrix} \\ \text{Data Type: anything} \\ \text{Storage: rectangular} \\ \text{Order: Fortran_order} \end{bmatrix}$$

$$Q_9 := \begin{bmatrix} 1024 \times 5 \text{ Matrix} \\ \text{Data Type: anything} \\ \text{Storage: rectangular} \\ \text{Order: Fortran_order} \end{bmatrix}$$

(4)

$$\text{> } x0 := \begin{bmatrix} 73.8 \\ 0 \\ 26.2 \\ 0 \\ 0 \end{bmatrix}$$

$$x0 := \begin{bmatrix} 73.8 \\ 0 \\ 26.2 \\ 0 \\ 0 \end{bmatrix}$$

(5)

> for l from 0 to 9 do Z_l := Q_lx0 end do

$$Z_0 := \begin{bmatrix} 26.20000000000000 \\ 0. \end{bmatrix}$$

$$Z_1 := \begin{bmatrix} 73.8 a k_1 + 73.8 s k_4 f - 26.2 k_2 + 26.2 m k_2 \\ -73.8 s k_4 f \\ 0. \\ 0. \end{bmatrix}$$

$$Z_2 := [[-73.8 (a k_1 + s k_4 f) k_1 + 73.8 s k_4 f (1 - a) k_1 - 73.8 k_4 f (a k_1 + s k_4 f) + 26.2 s k_4 f (1 - m) k_2 - 26.2 (-k_2 + m k_2) k_2 - 26.2 k_4 f m k_2], \\ [73.8 s k_4 f k_1 - 73.8 s k_4 f (1 - a) k_1 + 73.8 k_4 f (a k_1 + s k_4 f) - 26.2 s k_4 f (1 - m) k_2 + 26.2 k_4 f m k_2], \\ [-73.8 k_3 (a k_1 + s k_4 f) - 26.2 k_3 m k_2], \\ [0.], \\ [0.], \\ [0.], \\ [0.], \\ [0.]]$$

$$Z_3 := \begin{bmatrix} 1 \dots 16 \text{ Vector}_{column} \\ \text{Data Type: anything} \\ \text{Storage: rectangular} \\ \text{Order: Fortran_order} \end{bmatrix}$$

$$Z_4 := \begin{bmatrix} 1 \dots 32 \text{ Vector}_{column} \\ \text{Data Type: anything} \\ \text{Storage: rectangular} \\ \text{Order: Fortran_order} \end{bmatrix}$$

$$\begin{aligned}
Z_5 &:= \left[\begin{array}{l} 1 \dots 64 \text{ Vector}_{\text{column}} \\ \text{Data Type: anything} \\ \text{Storage: rectangular} \\ \text{Order: Fortran_order} \end{array} \right] \\
Z_6 &:= \left[\begin{array}{l} 1 \dots 128 \text{ Vector}_{\text{column}} \\ \text{Data Type: anything} \\ \text{Storage: rectangular} \\ \text{Order: Fortran_order} \end{array} \right] \\
Z_7 &:= \left[\begin{array}{l} 1 \dots 256 \text{ Vector}_{\text{column}} \\ \text{Data Type: anything} \\ \text{Storage: rectangular} \\ \text{Order: Fortran_order} \end{array} \right] \\
Z_8 &:= \left[\begin{array}{l} 1 \dots 512 \text{ Vector}_{\text{column}} \\ \text{Data Type: anything} \\ \text{Storage: rectangular} \\ \text{Order: Fortran_order} \end{array} \right] \\
Z_9 &:= \left[\begin{array}{l} 1 \dots 1024 \text{ Vector}_{\text{column}} \\ \text{Data Type: anything} \\ \text{Storage: rectangular} \\ \text{Order: Fortran_order} \end{array} \right] \tag{6}
\end{aligned}$$

> *with(MTM)*
[*ElementwiseAnd, ElementwiseNot, ElementwiseOr, Map, Minus, Mod, Zip, abs, acos, acosh, acot, acoth, acsc, acsch, array_dims, asec, asech, asin, asinh, atan, atanh, besseli, besselj, bessell, bessely, ccode, ceil, char, coeffs, collect, colspace, compose, conj, cos, cosh, cosint, cot, coth, csc, csch, ctranspose, det, diag, diff, digits, dirac, disp, double, dsolve, eig, end, eq, erf, exp, expand, expm, ezcontour, ezcontourf, ezmesh, ezmeshc, ezplot, ezplot3, ezpolar, ezsurf, ezsurf, factor, findsym, finverse, fix, floor, fortran, fourier, frac, \gamma, gcd, ge, gt, heaviside, horner, horzcat, hypergeom, ifourier, ilaplace, imag, int, int16, int32, int64, int8, inv, isreal, iztrans, jacobian, jordan, lambertw, laplace, latex, lcm, ldivide, le, limit, log, log10, log2, lt, mfun,*

(7)

mldivide, mpower, mrdivide, mtimes, ne, null, numden, numel, plus, poly, poly2sym, power, pretty, procread, prod, quorem, rank, rdivide, real, round, rref, sec, sech, simple, simplify, sin, single, sinh, sinint, size, solve, sort, sqrt, struct, subs, subsasgn, subsref, sum, svd, sym2poly, symsum, tan, tanh, taylor, times, transpose, tril, triu, uint16, uint32, uint64, uint8, vertcat, vpa, ζ, ztrans]

> **for** *l* **from** 0 **to** 9 **do** $Sen_l := jacobian(Z_p, p)$ **end do**

$$Sen_0 := \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$Sen_1 := \begin{bmatrix} 73.8 a & -26.2 + 26.2 m & 0 & 73.8 s f & 26.2 k_2 & 73.8 k_1 & 73.8 k_4 f \\ 0 & 0 & 0 & -73.8 s f & 0 & 0 & -73.8 k_4 f \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$Sen_2 := [[-147.6 a k_1 - 73.8 s k_4 f + 73.8 s k_4 f (1 - a) - 73.8 k_4 f a, 26.2 s k_4 f (1 - m) - 26.2 (-1 + m) k_2 + 26.2 k_2 - 26.2 m k_2 - 26.2 k_4 f m, 0, -73.8 s f k_1 + 73.8 s f (1 - a) k_1 - 73.8 f (a k_1 + s k_4 f) - 73.8 k_4 f^2 s + 26.2 s f (1 - m) k_2 - 26.2 f m k_2, -26.2 s k_4 f k_2 - 26.2 k_2^2 - 26.2 k_4 f k_2, -73.8 k_1^2 - 73.8 s k_4 f k_1 - 73.8 k_4 f k_1, -73.8 k_4 f k_1 + 73.8 k_4 f (1 - a) k_1 - 73.8 k_4^2 f^2 + 26.2 k_4 f (1 - m) k_2],$$

$$[73.8 s k_4 f - 73.8 s k_4 f (1 - a) + 73.8 k_4 f a, -26.2 s k_4 f (1 - m) + 26.2 k_4 f m, 0, 73.8 s f k_1 - 73.8 s f (1 - a) k_1 + 73.8 f (a k_1 + s k_4 f) + 73.8 k_4 f^2 s - 26.2 s f (1 - m) k_2 + 26.2 f m k_2, 26.2 s k_4 f k_2 + 26.2 k_4 f k_2, 73.8 s k_4 f k_1 + 73.8 k_4 f k_1, 73.8 k_4 f k_1 - 73.8 k_4 f (1 - a) k_1 + 73.8 k_4^2 f^2 - 26.2 k_4 f (1 - m) k_2],$$

$$[-73.8 k_3 a, -26.2 k_3 m, -73.8 a k_1 - 73.8 s k_4 f - 26.2 m k_2, -73.8 k_3 s f, -26.2 k_3 k_2, -73.8 k_3 k_1, -73.8 k_3 k_4 f],$$

$$[0, 0, 0, 0, 0, 0, 0],$$

$$[0, 0, 0, 0, 0, 0, 0],$$

$$[0, 0, 0, 0, 0, 0, 0],$$

$$[0, 0, 0, 0, 0, 0, 0],$$

[0, 0, 0, 0, 0, 0, 0]]

$Sen_3 := \begin{bmatrix} 16 \times 7 \text{ Matrix} \\ \text{Data Type: anything} \\ \text{Storage: rectangular} \\ \text{Order: Fortran_order} \end{bmatrix}$

$Sen_4 := \begin{bmatrix} 32 \times 7 \text{ Matrix} \\ \text{Data Type: anything} \\ \text{Storage: rectangular} \\ \text{Order: Fortran_order} \end{bmatrix}$

$Sen_5 := \begin{bmatrix} 64 \times 7 \text{ Matrix} \\ \text{Data Type: anything} \\ \text{Storage: rectangular} \\ \text{Order: Fortran_order} \end{bmatrix}$

$Sen_6 := \begin{bmatrix} 128 \times 7 \text{ Matrix} \\ \text{Data Type: anything} \\ \text{Storage: rectangular} \\ \text{Order: Fortran_order} \end{bmatrix}$

$Sen_7 := \begin{bmatrix} 256 \times 7 \text{ Matrix} \\ \text{Data Type: anything} \\ \text{Storage: rectangular} \\ \text{Order: Fortran_order} \end{bmatrix}$

$Sen_8 := \begin{bmatrix} 512 \times 7 \text{ Matrix} \\ \text{Data Type: anything} \\ \text{Storage: rectangular} \\ \text{Order: Fortran_order} \end{bmatrix}$

$Sen_9 := \begin{bmatrix} 1024 \times 7 \text{ Matrix} \\ \text{Data Type: anything} \\ \text{Storage: rectangular} \\ \text{Order: Fortran_order} \end{bmatrix}$

(8)

> with(LinearAlgebra)

[&x, Add, Adjoint, BackwardSubstitute, BandMatrix, Basis, BezoutMatrix,

(9)

BidiagonalForm, BilinearForm, CARE, CharacteristicMatrix, CharacteristicPolynomial, Column, ColumnDimension, ColumnOperation, ColumnSpace, CompanionMatrix, ConditionNumber, ConstantMatrix, ConstantVector, Copy, CreatePermutation, CrossProduct, DARE, DeleteColumn, DeleteRow, Determinant, Diagonal, DiagonalMatrix, Dimension, Dimensions, DotProduct, EigenConditionNumbers, Eigenvalues, Eigenvectors, Equal, ForwardSubstitute, FrobeniusForm, GaussianElimination, GenerateEquations, GenerateMatrix, Generic, GetResultDataType, GetResultShape, GivensRotationMatrix, GramSchmidt, HankelMatrix, HermiteForm, HermitianTranspose, HessenbergForm, HilbertMatrix, HouseholderMatrix, IdentityMatrix, IntersectionBasis, IsDefinite, IsOrthogonal, IsSimilar, IsUnitary, JordanBlockMatrix, JordanForm, KroneckerProduct, LA_Main, LUdecomposition, LeastSquares, LinearSolve, LyapunovSolve, Map, Map2, MatrixAdd, MatrixExponential, MatrixFunction, MatrixInverse, MatrixMatrixMultiply, MatrixNorm, MatrixPower, MatrixScalarMultiply, MatrixVectorMultiply, MinimalPolynomial, Minor, Modular, Multiply, NoUserValue, Norm, Normalize, NullSpace, OuterProductMatrix, Permanent, Pivot, PopovForm, QRdecomposition, RandomMatrix, RandomVector, Rank, RationalCanonicalForm, ReducedRowEchelonForm, Row, RowDimension, RowOperation, RowSpace, ScalarMatrix, ScalarMultiply, ScalarVector, SchurForm, SingularValues, SmithForm, StronglyConnectedBlocks, SubMatrix, SubVector, SumBasis, SylvesterMatrix, SylvesterSolve, ToeplitzMatrix, Trace, Transpose, TridiagonalForm, UnitVector, VandermondeMatrix, VectorAdd, VectorAngle, VectorMatrixMultiply, VectorNorm, VectorScalarMultiply, ZeroMatrix, ZeroVector, Zip]

$$\text{> Rank}(Sen_1) \qquad \qquad \qquad 2 \qquad \qquad \qquad (10)$$

$$\text{> } R1 := \text{RowSpace}(Sen_1)$$

$$R1 := \left[\left[1, \frac{0.01355013550 (-26.2 + 26.2 m)}{a}, 0, 0, \frac{0.3550135501 k_2}{a}, \right. \right. \qquad (11)$$

$$\left. \left. \frac{1.000000000 k_1}{a}, 0, \right] \left[0 \ 0 \ 0 \ 1 \ 0 \ 0 \ \frac{1.000000000 k_4}{s} \right] \right]$$

$$\text{> with}(linalg) \qquad \qquad \qquad (12)$$

[BlockDiagonal, GramSchmidt, JordanBlock, LUdecomp, QRdecomp, Wronskian, addcol, addrow, adj, adjoint, angle, augment, backsub, band, basis, bezout, blockmatrix, charmat, charpoly, cholesky, col, coldim, colspace, colspan,

companion, concat, cond, copyinto, crossprod, curl, definite, delcols, delrows, det, diag, diverge, dotprod, eigenvals, eigenvalues, eigenvectors, eigenvects, entermatrix, equal, exponential, extend, ffgausselim, fibonacci, forwardsub, frobenius, gausselim, gaussjord, geneqns, genmatrix, grad, hadamard, hermite, hessian, hilbert, htranspose, ihermite, indexfunc, innerprod, intbasis, inverse, ismith, issimilar, iszero, jacobian, jordan, kernel, laplacian, leastsqrs, linsolve, matadd, matrix, minor, minpoly, mulcol, mulrow, multiply, norm, normalize, nullspace, orthog, permanent, pivot, potential, randmatrix, randvector, rank, ratform, row, rowdim, rowspace, rowspan, rref, scalarmul, singularvals, smith, stackmatrix, submatrix, subvector, sumbasis, swapcol, swaprow, sylvester, toeplitz, trace, transpose, vandermonde, vecpotent, vectdim, vector, wronskian]

$$\begin{aligned}
 &> Rmat1 := stackmatrix(RI[1], RI[2]) \\
 Rmat1 := &\left[\left[1, \frac{0.01355013550 (-26.2 + 26.2 m)}{a}, 0, 0, \frac{0.3550135501 k_2}{a}, \right. \right. \\
 &\left. \left. \frac{1.000000000 k_1}{a}, 0. \right], \right. \\
 &\left. \left[0, 0, 0, 1, 0, 0, \frac{1.000000000 k_4}{s} \right] \right] \quad (13)
 \end{aligned}$$

$$\begin{aligned}
 &> MI := \langle Rmat1, Sen_2 \rangle \\
 MI := &\left[\left[1, \frac{0.01355013550 (-26.2 + 26.2 m)}{a}, 0, 0, \frac{0.3550135501 k_2}{a}, \right. \right. \\
 &\left. \left. \frac{1.000000000 k_1}{a}, 0. \right], \right. \\
 &\left[0, 0, 0, 1, 0, 0, \frac{1.000000000 k_4}{s} \right], \\
 &\left[-147.6 a k_1 - 73.8 s k_4 f + 73.8 s k_4 f (1 - a) - 73.8 k_4 f a, 26.2 s k_4 f (1 - m) \right. \\
 &- 26.2 (-1 + m) k_2 + 26.2 k_2 - 26.2 m k_2 - 26.2 k_4 f m, 0, -73.8 s f k_1 \\
 &+ 73.8 s f (1 - a) k_1 - 73.8 f (a k_1 + s k_4 f) - 73.8 k_4 f^2 s + 26.2 s f (1 - m) k_2 \\
 &- 26.2 f m k_2, -26.2 s k_4 f k_2 - 26.2 k_2^2 - 26.2 k_4 f k_2, -73.8 k_1^2 - 73.8 s k_4 f k_1 \\
 &- 73.8 k_4 f k_1, -73.8 k_4 f k_1 + 73.8 k_4 f (1 - a) k_1 - 73.8 k_4^2 f^2 + 26.2 k_4 f (1 \\
 &- m) k_2 \left. \right], \\
 &\left[73.8 s k_4 f - 73.8 s k_4 f (1 - a) + 73.8 k_4 f a, -26.2 s k_4 f (1 - m) \right. \\
 &+ 26.2 k_4 f m, 0, 73.8 s f k_1 - 73.8 s f (1 - a) k_1 + 73.8 f (a k_1 + s k_4 f) \\
 &+ 73.8 k_4 f^2 s - 26.2 s f (1 - m) k_2 + 26.2 f m k_2, 26.2 s k_4 f k_2 + 26.2 k_4 f k_2,
 \end{aligned} \quad (14)$$

```

73.8 s k4 f k1 + 73.8 k4 f k1, 73.8 k4 f k1 - 73.8 k4 f (1 - a) k1 + 73.8 k42 f2
- 26.2 k4 f (1 - m) k2],
[-73.8 k3 a, -26.2 k3 m, -73.8 a k1 - 73.8 s k4 f - 26.2 m k2, -73.8 k3 s f,
-26.2 k3 k2, -73.8 k3 k1, -73.8 k3 k4 f],
[0, 0, 0, 0, 0, 0, 0],
[0, 0, 0, 0, 0, 0, 0],
[0, 0, 0, 0, 0, 0, 0],
[0, 0, 0, 0, 0, 0, 0],
[0, 0, 0, 0, 0, 0, 0]]
> Rank(M1)
5 (15)
> R2 := RowSpace(M1)
> Rmat2 := stackmatrix(R2[1], R2[2], R2[3], R2[4], R2[5])
> M2 := <Rmat2, Sen3[1], Sen3[2]>
> Rank(M2)
7 (16)
> R3 := RowSpace(M2)
R3 := [[ 1 0 0 0 0 0. 0. ], [ 0 1 0 0 0 0. 0. ], [ 0 0 1 0 0 0. 0. ],
[ 0 0 0 1 0 0. 0. ], [ 0 0 0 0 1 0 0 ], [ 0 0 0 0 0 1 0 ],
[ 0 0 0 0 0 0. 1 ]]
>

```

Appendix B

MATLAB codes

B.1 Parameter estimation

1. ODEs

```
function dx=dbmodrecurws(t,x,u,p) %ODEs
% x(1):Pentane-soluble, reactant
% x(2):Pentane-soluble, product
% x(3):Asphaltenes, reactant
% x(4):Asphaltenes, product
% x(5):THF-insolubles
% p(1)=kP
% p(2)=kA
% p(3)=kH
% p(4)=kC
% p(5)=m'
% p(6)=a
% p(7)=SL
% u=PH2
dx=zeros(5,1);

if x(4)>(p(7)*(x(1)+x(2)))
    f=1;
else
    f=0;
end;

dx(1)=-p(1)*x(1);
dx(2)=(1-p(6))*p(1)*x(1)+(1-p(5))*p(2)*x(3)+p(3)*u*x(4);
dx(3)=-p(2)*x(3);
dx(4)=p(6)*p(1)*x(1)+p(5)*p(2)*x(3)-p(3)*u*x(4)-p(4)*(x(4)-
p(7)*(x(1)+x(2)))*f;
dx(5)=p(4)*(x(4)-p(7)*(x(1)+x(2)))*f;
```

2. ODEs including sensitivities

```

function dx=dbmodrecur(t,x,u,p) %ODEs including sensitivities
% x(1):Pentane-soluble, reactant
% x(2):Pentane-soluble, product
% x(3):Asphaltenes, reactant
% x(4):Asphaltenes, product
% x(5):THF-insolubles
% p(1)=kP
% p(2)=kA
% p(3)=kH
% p(4)=kC
% p(5)=m'
% p(6)=a
% p(7)=SL
% u=PH2
dx=zeros(40,1);

if x(4)>(p(7)*(x(1)+x(2)))
    f=1;
else
    f=0;
end;

dx(1)=-p(1)*x(1);
dx(2)=(1-p(6))*p(1)*x(1)+(1-p(5))*p(2)*x(3)+p(3)*u*x(4);
dx(3)=-p(2)*x(3);
dx(4)=p(6)*p(1)*x(1)+p(5)*p(2)*x(3)-p(3)*u*x(4)-p(4)*(x(4)-
p(7)*(x(1)+x(2)))*f;
dx(5)=p(4)*(x(4)-p(7)*(x(1)+x(2)))*f;

c = [-p(1), 0, 0, 0, 0; (1 - p(6)) * p(1), 0, (1 - p(5)) * p(2), u
* p(3), 0; 0, 0, -p(2), 0, 0; p(6) * p(1) + p(7) * p(4) * f, p(7)
* p(4) * f, p(5) * p(2), -p(4) * f - u * p(3), 0; -p(7) * p(4) *
f, -p(7) * p(4) * f, 0, p(4) * f, 0;];

b = [-x(1), 0, 0, 0, 0, 0, 0; (1 - p(6)) * x(1), (1 - p(5)) *
x(3), u * x(4), 0, -p(2) * x(3), -p(1) * x(1), 0; 0, -x(3), 0, 0,
0, 0, 0; p(6) * x(1), p(5) * x(3), -u * x(4), p(7) * f * x(1) +
p(7) * f * x(2) - f * x(4), p(2) * x(3), p(1) * x(1), p(4) * f *
x(1) + p(4) * f * x(2); 0, 0, 0, -p(7) * f * x(1) - p(7) * f *
x(2) + f * x(4), 0, 0, -p(4) * f * x(1) - p(4) * f * x(2)];

r=zeros(5,7);
for d=1:7
    for e=1:5
        r(e,d)=x(d*5+e);
    end;
end;

s=b+c*r;
for d=1:7
    for e=1:5
        dx(d*5+e)=s(e,d);
    end;
end;

```

3. Sensitivity test

```
clear all

p=[0.08    0.18    0.0024    0.02    0.55    0.20    0.15];
u=20;
x0=[73.8,0,26.2,0,0,zeros(1,35)];
[T,X]=ode45(@(t,x)dbmodrecur(t,x,u,p),[0,15,30,60,120,240],x0);

Y=zeros(size(X,1),16);
for i=1:8
    Y(:,2*(i-1)+1)=X(:,5*(i-1)+3)+X(:,5*(i-1)+4);
    Y(:,2*i)=X(:,5*i);
end;

Z=zeros(2*size(Y,1),7);

for i=1:size(Y,1)
    for j=1:2
        for k=1:7
            Z((i-1)*2+j,k)=Y(i,k*2+j);
        end;
    end;
end;

[U,S,V]=svd(Z);

Sd=diag(S);
for i=1:7
    s(i)=Sd(i)/Sd(1);
end;

bar(s) %Resulting in bar figure of relative singular values
```

4. Singular Value Decomposition (SVD)

```
function V=svdcal(p)
```

```
u=20;
x0=[73.8,0,26.2,0,0,zeros(1,35)];
[T,X]=ode45(@(t,x)dbmodrecur(t,x,u,p),[0,200],x0);

Y=zeros(size(X,1),16);
for i=1:8
    Y(:,2*(i-1)+1)=X(:,5*(i-1)+3)+X(:,5*(i-1)+4);
    Y(:,2*i)=X(:,5*i);
end;

Z=zeros(2*size(Y,1),7);

for i=1:size(Y,1)
    for j=1:2
        for k=1:7
            Z((i-1)*2+j,k)=Y(i,k*2+j);
        end;
    end;
end;

[U,S,V]=svd(Z);
```

5. Objective function for parameter estimation stage

```
function J=dbrecurlinrep(c0,t,u,k,V,k1)
global s

kp=[k,k1]';
p=V*kp;
[m,n]=size(c0);

y0=c0(1,:);
x0=zeros(1,5);
x0(1,3)=y0(1,1);
x0(1,5)=y0(1,2);
x0(1,1)=100-y0(1,1)-y0(1,2);
Y(1,:)=y0;

for i=1:m-1
    [T,X]=ode45(@(t,x)dbmodrecurws(t,x,u,p),[t(i,1) t(i+1,1)],x0);
    x0=X(end,:);
    Y(i+1,1)=X(end,3)+X(end,4);
    Y(i+1,2)=X(end,5);
    clear X T
end;
W=zeros(1,2);
W(1,1)=12/norm(c0(:,1));
W(1,2)=12/norm(c0(:,2));
J=0;
s=s+1
for l=1:m
    for o=1:n
        J=J+(W(1,o)*(Y(1,o)-c0(1,o))).^2;
    end;
end;
J
```

```

6. Parameter estimation with optimizing over only one pseudo-
parameter


---


clear all
clc
global s
s=0;
u=20;
%load p1.mat
load Y.mat
load T.mat

p=[0.08      0.18      0.0024  0.02      0.55      0.20      0.15];%initial
guess
V=svdcal(p);
vp=p*V;%transforming p
k0=vp(1,1);%the pseudo-parameters that are being held constant
k1=vp(1,2:7);% the initial guess for pseudo-parameters that are
being optimized
exitflag=0;%the exit error for patternsearch which has to be
"minimum TolX has reached"

A(1:7,:)= -V(:,1);
b(1:7,1)=V(:,7)*k1(1,6)+V(:,6)*k1(1,5)+V(:,5)*k1(1,4)+V(:,4)*k1(1,
3)+V(:,3)*k1(1,2)+V(:,2)*k1(1,1);% the transformed left-hand side
of constraint for parameters (0<p)
A(8:14,:)=V(:,1);
b(8:14,1)=ones(7,1)-
(V(:,7)*k1(1,6)+V(:,6)*k1(1,5)+V(:,5)*k1(1,4)+V(:,4)*k1(1,3)+V(:,3
)*k1(1,2)+V(:,2)*k1(1,1));% the transformed right-hand side of
constraint for parameters (p<1)

options1=psoptimset('TolFun',0.000001,'TolX',0.000001,'MaxIter',
25, 'MaxFunEval', 30000);

while exitflag==0

[K,fval,exitflag]=fmincon(@(k)dbrecurlinrep(Y,T,u,k,V,k1),k0,A,b,[
],[],[
],[],[],options1);
vp=[K,k1];% the answer of optimization+ constant pseudo-
paramaters
p=vp*V';%new set of parameter estimates
V=svdcal(p);% new V based on new p
kf=vp;
vp=p*V;% new set of pseudo-parameter estimates based on new V
k0=vp(1,1);% new initial guess based on new p and V
k1=vp(1,2:7);% new values of constant pseudo-parameters
%Upadating constraints
A(1:7,:)= -V(:,1);

b(1:7,1)=V(:,7)*k1(1,6)+V(:,6)*k1(1,5)+V(:,5)*k1(1,4)+V(:,4)*k1(1,
3)+V(:,3)*k1(1,2)+V(:,2)*k1(1,1);
A(8:14,:)=V(:,1);
b(8:14,1)=ones(7,1)-
(V(:,7)*k1(1,6)+V(:,6)*k1(1,5)+V(:,5)*k1(1,4)+

```

```
        V(:,4)*k1(1,3)+V(:,3)*k1(1,2)+V(:,2)*k1(1,1));  
end;  
  
save('p1','p');  
run RecursiveLinReparaSimulator  
clear all  
clc
```

7. Parameter estimation with optimizing over two pseudo-parameters

```

global s
s=0;
u=20;
load Y.mat
load T.mat
%load p2.mat
p=[0.08    0.18    0.0024  0.02    0.55    0.20    0.15];%initial
guess
V=svdcal(p);
vp=p*V;%transforming p
k0=vp(1,1:2);%the pseudo-parameters that are being held constant
k1=vp(1,3:7);% the initial guess for pseudo-parameters that are
being optimized
exitflag=0;%the exit error for patternsearch which has to be
"minimum TolX has reached"

A(1:7,:)= -V(:,1:2);
b(1:7,1)=V(:,7)*k1(1,5)+V(:,6)*k1(1,4)+V(:,5)*k1(1,3)+V(:,4)*k1(1,
2)+V(:,3)*k1(1,1);% the transformed left-hand side of constraint
for parameters (0<p)
A(8:14,:)=V(:,1:2);
b(8:14,1)=ones(7,1)-
(V(:,7)*k1(1,5)+V(:,6)*k1(1,4)+V(:,5)*k1(1,3)+V(:,4)*k1(1,2)+V(:,3
)*k1(1,1));% the transformed right-hand side of constraint for
parameters (p<1)

options1=psoptimset('TolFun',0.000001,'TolX',0.000001,'MaxIter',
25, 'MaxFunEval', 30000);

while exitflag==0

[K,fval,exitflag]=fmincon(@(k)dbrecurlinrep(Y,T,u,k,V,k1),k0,A,b,[
],[],
[],[],[],options1);
vp=[K,k1];% the answer of optimization+ constant pseudo-
paramaters
p=vp*V';%new set of parameter estimates
Vn=svdcal(p);% new V based on new p
kf=vp;
V=Vn;% new V based on new p
vp=p*V;% new set of pseudo-parameter estimates based on new V
k0=vp(1,1:2);% new initial guess based on new p and V
k1=vp(1,3:7);% new values of constant pseudo-parameters
%Updating constraints
A(1:7,:)= -V(:,1:2);
b(1:7,1)=V(:,7)*k1(1,5)+V(:,6)*k1(1,4)+V(:,5)*k1(1,3)+
V(:,4)*k1(1,2)+V(:,3)*k1(1,1);
A(8:14,:)=V(:,1:2);
b(8:14,1)=ones(7,1)-
(V(:,7)*k1(1,5)+V(:,6)*k1(1,4)+V(:,5)*k1(1,3)+
V(:,4)*k1(1,2)+V(:,3)*k1(1,1));
end;

save('p2','p');
run RecursiveLinReparaSimulator

```

8. Parameter estimation with optimizing over three pseudo-parameters

```

clear all
clc
global s
s=0;
u=20;
%load p32.mat
load Y.mat
load T.mat

p=[0.08    0.18    0.0024  0.02    0.55    0.20    0.15];%initial
guess
V=svdcal(p);
vp=p*V;%transforming p
k0=vp(1,1:3);%the pseudo-parameters that are being held constant
k1=vp(1,4:7);% the initial guess for pseudo-parameters that are
being optimized
exitflag=0;%the exit error for patternsearch which has to be
"minimum TolX has reached"

A(1:7,:)= -V(:,1:3);
b(1:7,1)=V(:,7)*k1(1,4)+V(:,6)*k1(1,3)+V(:,5)*k1(1,2)+V(:,4)*k1(1,
1);% the transformed left-hand side of constraint for parameters
(0<p)
A(8:14,:)=V(:,1:3);
b(8:14,1)=ones(7,1)-
(V(:,7)*k1(1,4)+V(:,6)*k1(1,3)+V(:,5)*k1(1,2)+V(:,4)*k1(1,1));%
the transformed right-hand side of constraint for parameters (p<1)

options1=psoptimset('TolFun',0.000001,'TolX',0.000001,'MaxIter',
25, 'MaxFunEval', 30000);

while exitflag==0

[K,fval,exitflag]=fmincon(@(k)dbrecurlinrep(Y,T,u,k,V,k1),k0,A,b,[
],[],
[],[],[],options1);
vp=[K,k1];% the answer of optimization+ constant pseudo-
paramaters
p=vp*V';%new set of parameter estimates
V_op=V;
k_op=vp;
V=svdcal(p);% new V based on new p
vp=p*V;% new set of pseudo-parameter estimates based on new V
k0=vp(1,1:3);% new initial guess based on new p and V
k1=vp(1,4:7);% new values of constant pseudo-parameters
%Updating constraints
A(1:7,:)= -V(:,1:3);

b(1:7,1)=V(:,7)*k1(1,4)+V(:,6)*k1(1,3)+V(:,5)*k1(1,2)+V(:,4)*k1(1,
1);
A(8:14,:)=V(:,1:3);
b(8:14,1)=ones(7,1)-
(V(:,7)*k1(1,4)+V(:,6)*k1(1,3)+V(:,5)*k1(1,2)+
V(:,4)*k1(1,1));

```

```
end;  
save('p32','p');  
save('V_op32','V_op')  
save('k_op32','k_op')  
save('/home/sshams/model2v/Jan14_2measVarib/K_CI/p32','p');  
save('/home/sshams/model2v/Jan14_2measVarib/K_CI/V_op32','V_op');  
save('/home/sshams/model2v/Jan14_2measVarib/K_CI/k_op32','k_op');  
run RecursiveLinReparaSimulator
```

```

9. Parameter estimation with optimizing over four pseudo-
parameters


---


clear all
clc
global s
s=0;
u=20;
%load p4.mat
load Y.mat
load T.mat

p=[0.08      0.18      0.0024  0.02      0.55      0.20      0.15];%initial
guess
V=svdcal(p);
vp=p*V;%transforming p
k0=vp(1,1:4);%the pseudo-parameters that are being held constant
k1=vp(1,5:7);% the initial guess for pseudo-parameters that are
being optimized
exitflag=0;%the exit error for patternsearch which has to be
"minimum TolX has reached"

A(1:7,:)= -V(:,1:4);
b(1:7,1)=V(:,7)*k1(1,3)+V(:,6)*k1(1,2)+V(:,5)*k1(1,1);% the
transformed left-hand side of constraint for parameters (0<p)
A(8:14,:)=V(:,1:4);
b(8:14,1)=ones(7,1)-
(V(:,7)*k1(1,3)+V(:,6)*k1(1,2)+V(:,5)*k1(1,1));% the transformed
right-hand side of constraint for parameters (p<1)

options1=psoptimset('TolFun',0.000001,'TolX',0.000001,'MaxIter',20
,'MaxFunEval',30000);

while exitflag==0

[K,fval,exitflag]=patternsearch(@(k)dbrecurlinrep(Y,T,u,k,V,k1),k0
,A,b,
[],[],[],[],[],options1);
vp=[K,k1];% the answer of optimization+ constant pseudo-
paramaters
p=vp*V';%new set of parameter estimates
kf=vp;
V=svdcal(p);% new V based on new p
vp=p*V;% new set of pseudo-parameter estimates based on new V
k0=vp(1,1:4);% new initial guess based on new p and V
k1=vp(1,5:7);% new values of constant pseudo-parameters
%Updating constraints
A(1:7,:)= -V(:,1:4);
b(1:7,1)=V(:,7)*k1(1,3)+V(:,6)*k1(1,2)+V(:,5)*k1(1,1);
A(8:14,:)=V(:,1:4);
b(8:14,1)=ones(7,1)-
(V(:,7)*k1(1,3)+V(:,6)*k1(1,2)+V(:,5)*k1(1,1));
save('p4','p');
end;

run RecursiveLinReparaSimulator

```

```

10. Parameter estimation with optimizing over five pseudo-
parameters


---


clear all
clc
global s
s=0;
u=20;
%load p5.mat
load Y.mat
load T.mat

p=[0.08    0.18    0.0024  0.02    0.55    0.20    0.15];%initial
guess
V=svdcal(p);
vp=p*V;%transforming p
k0=vp(1,1:5);%the pseudo-parameters that are being held constant
k1=vp(1,6:7);% the initial guess for pseudo-parameters that are
being optimized
exitflag=0;%the exit error for patternsearch which has to be
"minimum TolX has reached"

A(1:7,:)= -V(:,1:5);
b(1:7,1)=V(:,7)*k1(1,2)+V(:,6)*k1(1,1);% the transformed left-hand
side of constraint for parameters (0<p)
A(8:14,:)=V(:,1:5);
b(8:14,1)=ones(7,1)-(V(:,7)*k1(1,2)+V(:,6)*k1(1,1));% the
transformed right-hand side of constraint for parameters (p<1)

options1=psoptimset('TolFun',0.000001,'TolX',0.000001,'MaxIter',
33, 'MaxFunEval', 30000);
pashe=0;
while exitflag==0

[K,fval,exitflag]=patternsearch(@(k)dbrecurlinrep(Y,T,u,k,V,k1),k0
,A,b,
[],[],[],[],[],options1);
vp=[K,k1];% the answer of optimization+ constant pseudo-
paramaters
p=vp*V';%new set of parameter estimates
V=svdcal(p);% new V based on new p
kf=vp;
vp=p*V;% new set of pseudo-parameter estimates based on new V
k0=vp(1,1:5);% new initial guess based on new p and V
k1=vp(1,6:7);% new values of constant pseudo-parameters
%Updating Constraints
A(1:7,:)= -V(:,1:5);
b(1:7,1)=V(:,7)*k1(1,2)+V(:,6)*k1(1,1);
A(8:14,:)=V(:,1:5);
b(8:14,1)=ones(7,1)-(V(:,7)*k1(1,2)+V(:,6)*k1(1,1));
pashe=pashe+1;
end;

save('p5','p');
run RecursiveLinReparaSimulator

```

```

11. Parameter estimation with optimizing over six pseudo-
parameters


---


clear all
clc
global s
s=0;
u=20;
%load p6.mat
load Y.mat
load T.mat

p=[0.08    0.18    0.0024  0.02    0.55    0.20    0.15];%initial
guess
V=svdcal(p);
vp=p*V;%transforming p
k0=vp(1,1:6);%the pseudo-parameters that are being held constant
k1=vp(1,7);% the initial guess for pseudo-parameters that are
being optimized
exitflag=0;%the exit error for patternsearch which has to be
"minimum TolX has reached"

A(1:7,:)= -V(:,1:6);
b(1:7,1)=V(:,7)*k1(1,1);% the transformed left-hand side of
constraint for parameters (0<p)
A(8:14,:)=V(:,1:6);
b(8:14,1)=ones(7,1)-(V(:,7)*k1(1,1));% the transformed right-hand
side of constraint for parameters (p<1)

options1=psoptimset('TolFun',0.000001,'TolX',0.000001,'MaxIter',
20, 'MaxFunEval', 30000);
pashe=0;
while exitflag==0

[K,fval,exitflag]=fmincon(@(k)dbrecurlinrep(Y,T,u,k,V,k1),k0,A,b,
[],[],[],[],[],options1);
vp=[K,k1];% the answer of optimization+ constant pseudo-
paramaters
p=vp*V';%new set of parameter estimates
kf=vp;
V=svdcal(p);% new V based on new p
vp=p*V;% new set of pseudo-parameter estimates based on new V
k0=vp(1,1:6);% new initial guess based on new p and V
k1=vp(1,7);% new values of constant pseudo-parameters
%Updating constraints
A(1:7,:)= -V(:,1:6);
b(1:7,1)=V(:,7)*k1(1,1);
A(8:14,:)=V(:,1:6);
b(8:14,1)=ones(7,1)-(V(:,7)*k1(1,1));
pashe=pashe+1;
end;

save('p6','p');
run RecursiveLinReparaSimulator

```

12. Parameter estimation with optimizing over all the pseudo-parameters

```
clear all
clc
global s
s=0;
u=20;
%load p7.mat
load Y.mat
load T.mat

p=[0.08    0.18    0.0024    0.02    0.55    0.20
0.15];%initial guess
V=svdcal(p);
vp=p*V;%transforming p
k0=vp(1,1:7);%the pseudo-parameters that are being held constant
k1=[];
exitflag=0;%the exit error for patternsearch which has to be
"minimum TolX has reached"

A(1:7,:)= -V(:,1:7);
b(1:7,1)=zeros(7,1);% the transformed left-hand side of constraint
for parameters (0<p)
A(8:14,:)=V(:,1:7);
b(8:14,1)=ones(7,1);% the transformed right-hand side of
constraint for parameters (p<1)

options1=psoptimset('TolFun',0.000001,'TolX',0.000001,'MaxIter',
25, 'MaxFunEval', 30000);
pashe=0;
while exitflag==0

[K,fval,exitflag]=fmincon(@(k)dbrecurlinrep(Y,T,u,k,V,k1),k0,A,b,
[],[],[],[],[],options1);
vp=[K,k1];% the answer of optimization+ constant pseudo-
parameters
p=vp*V';%new set of parameter estimates
kf=vp;
V=svdcal(p);% new V based on new p
vp=p*V;% new set of pseudo-parameter estimates based on new V
k0=vp(1,1:7);% new initial guess based on new p and V
k1=[];% new values of constant pseudo-parameters
%Updating constraints
A(1:7,:)= -V(:,1:7);
b(1:7,1)=zeros(7,1);
A(8:14,:)=V(:,1:7);
b(8:14,1)=ones(7,1);
pashe=pashe+1;
save('p7','p');
end;

run RecursiveLinReparaSimulator
```

13. RecursiveLinReparaSimulator

```
clc
%simulates the system using estimated parameters, the actual
points are
%from Y.mat and T.mat

clear u y0 x0 X time Ys

u=20;
y0=Y(1,:);
x0=zeros(1,5);
x0(1,3)=y0(1,1);
x0(1,5)=y0(1,2);
x0(1,1)=100-y0(1,1)-y0(1,2);

[time,X]=ode45(@(t,x)dbmodrecurws(t,x,u,p),[0 250],x0);
Ys(:,1)=X(:,3)+X(:,4);
Ys(:,2)=X(:,5);
Ys=roundn(Ys,-1);

Ys=roundn(Ys,-4);

plot(time,Ys(1:end,1),'-g',time,Ys(1:end,2),'-
b',T,Y(1:end,1),'g',T,Y(1:end,2),'b','LineWidth',2,'MarkerSize',
20);
set(gca,'FontSize',16)
legend('A','THFI','A (exp)','THFI (exp)')
xlabel('time (min)','FontSize',18);
ylabel('%wt','FontSize',18);
```

B.2 Optimal experimental design

1. ODEs including sensitivities

```

function dx=dbmodrecur(t,x,u,p)
% x(1):Pentane-soluble, reactant
% x(2):Pentane-soluble, product
% x(3):Asphaltenes, reactant
% x(4):Asphaltenes, product
% x(5):THF-insolubles
% p(1)=kP
% p(2)=kA
% p(3)=kH
% p(4)=kC
% p(5)=m'
% p(6)=a
% p(7)=SL
% u=P_H2
dx=zeros(40,1);

if x(4)>(p(7)*(x(1)+x(2)))
    f=1;
else
    f=0;
end;

dx(1)=-p(1)*x(1);
dx(2)=(1-p(6))*p(1)*x(1)+(1-p(5))*p(2)*x(3)+p(3)*u*x(4);
dx(3)=-p(2)*x(3);
dx(4)=p(6)*p(1)*x(1)+p(5)*p(2)*x(3)-p(3)*u*x(4)-p(4)*(x(4)-
p(7)*(x(1)+x(2)))*f;
dx(5)=p(4)*(x(4)-p(7)*(x(1)+x(2)))*f;

c = [-p(1), 0, 0, 0, 0; (1 - p(6)) * p(1), 0, (1 - p(5)) * p(2), u
* p(3), 0; 0, 0, -p(2), 0, 0; p(6) * p(1) + p(7) * p(4) * f, p(7)
* p(4) * f, p(5) * p(2), -p(4) * f - u * p(3), 0; -p(7) * p(4) *
f, -p(7) * p(4) * f, 0, p(4) * f, 0;];

b = [-x(1), 0, 0, 0, 0, 0, 0; (1 - p(6)) * x(1), (1 - p(5)) *
x(3), u * x(4), 0, -p(2) * x(3), -p(1) * x(1), 0; 0, -x(3), 0, 0,
0, 0, 0; p(6) * x(1), p(5) * x(3), -u * x(4), p(7) * f * x(1) +
p(7) * f * x(2) - f * x(4), p(2) * x(3), p(1) * x(1), p(4) * f *
x(1) + p(4) * f * x(2); 0, 0, 0, -p(7) * f * x(1) - p(7) * f *
x(2) + f * x(4), 0, 0, -p(4) * f * x(1) - p(4) * f * x(2)];

r=zeros(5,7);
for d=1:7
    for e=1:5
        r(e,d)=x(d*5+e);
    end;
end;

s=b+c*r;
for d=1:7
    for e=1:5
        dx(d*5+e)=s(e,d);
    end;
end;

```

2. Building Z_{Scaled} for one-experiment design

```
function J=maxdet1(Ts,p)
global rs
u=Ts(2,1);
x0=[100-Ts(1,1),0,Ts(1,1),0,0,zeros(1,35)];

[T,X]=ode45(@(t,x)dbmodrecur(t,x,20,p),[0,15,30,60,120,240],[73.8,
0,26.2,0,0,zeros(1,35)]);
[T1,X1]=ode45(@(t,x)dbmodrecur(t,x,u,p),(0:10:250),x0);

Y=zeros(size(X,1),16);
for i=1:8
    Y(:,2*(i-1)+1)=X(:,5*(i-1)+3)+X(:,5*(i-1)+4);
    Y(:,2*i)=X(:,5*i);
end;
for i=1:16
    if norm(Y(:,i))>0
        Y(:,i)=Y(:,i)*size(Y,1)/norm(Y(:,i));
    end;
end;
Z=zeros(2*size(Y,1),7);

for i=1:size(Y,1)
    for j=1:2
        for k=1:7
            Z((i-1)*2+j,k)=p(1,k)*Y(i,k*2+j);
        end;
    end;
end;

Y1=zeros(size(X1,1),16);
for i=1:8
    Y1(:,2*(i-1)+1)=X1(:,5*(i-1)+3)+X1(:,5*(i-1)+4);
    Y1(:,2*i)=X1(:,5*i);
end;
for i=1:16
    if norm(Y1(:,i))>0
        Y1(:,i)=Y1(:,i)*size(Y1,1)/norm(Y1(:,i));
    end;
end;
Z1=zeros(2*size(Y1,1),7);

for i=1:size(Y1,1)
    for j=1:2
        for k=1:7
            Z1((i-1)*2+j,k)=p(1,k)*Y1(i,k*2+j);
        end;
    end;
end;
z=[Z;Z1];
rs=rs+1
J=-abs(det(z'*z))
```

3. Designing one additional optimal experiment

```
clear all
clc
global rs
rs=0;
load p32.mat

options=psoptimset('TolMesh',10^-8,'TolX',0.001);
[Ts,fval]=patternsearch(@(ts)maxdet1(ts,p),[32;15],[],[],[],[],[0;
8],[32;20],[]);
save('TsOneReal','Ts')
```

4. Building Z_{Scaled} for two-experiment design

```
function J=maxdetTwo(Ts,p)
global rs

u1=Ts(3,1);
u2=Ts(4,1);

%r=fix(250/Ts(3,1));

x01=[100-Ts(1,1),0,Ts(1,1),0,0,zeros(1,35)];
x02=[100-Ts(2,1),0,Ts(2,1),0,0,zeros(1,35)];

[T,X]=ode45(@(t,x)dbmodrecur(t,x,20,p),[0,15,30,60,120,240],[73.8,
0,26.2,0,0,zeros(1,35)]);
[T1,X1]=ode45(@(t,x)dbmodrecur(t,x,u1,p),(0:10:250),x01);
[T2,X2]=ode45(@(t,x)dbmodrecur(t,x,u2,p),(0:10:250),x02);

Y=zeros(size(X,1),16);
for i=1:8
    Y(:,2*(i-1)+1)=X(:,5*(i-1)+3)+X(:,5*(i-1)+4);
    Y(:,2*i)=X(:,5*i);
end;
for i=1:16
    if norm(Y(:,i))>0
        Y(:,i)=Y(:,i)*size(Y,1)/norm(Y(:,i));
    end;
end;

Y1=zeros(size(X1,1),16);
for i=1:8
    Y1(:,2*(i-1)+1)=X1(:,5*(i-1)+3)+X1(:,5*(i-1)+4);
    Y1(:,2*i)=X1(:,5*i);
end;

Y2=zeros(size(X2,1),16);
for i=1:8
    Y2(:,2*(i-1)+1)=X2(:,5*(i-1)+3)+X2(:,5*(i-1)+4);
    Y2(:,2*i)=X2(:,5*i);
end;

for i=1:16
    if norm(Y1(:,i))>0
        Y1(:,i)=Y1(:,i)*size(Y1,1)/norm(Y1(:,i));
    end;
end;

for i=1:16
    if norm(Y2(:,i))>0
        Y2(:,i)=Y2(:,i)*size(Y2,1)/norm(Y2(:,i));
    end;
end;
end;
```

```

Z=zeros(2*size(Y,1),7);

for i=1:size(Y,1)
    for j=1:2
        for k=1:7
            Z((i-1)*2+j,k)=p(1,k)*Y(i,k*2+j);
        end;
    end;
end;

Z1=zeros(2*size(Y1,1),7);

for i=1:size(Y1,1)
    for j=1:2
        for k=1:7
            Z1((i-1)*2+j,k)=p(1,k)*Y1(i,k*2+j);
        end;
    end;
end;
Z2=zeros(2*size(Y2,1),7);

for i=1:size(Y2,1)
    for j=1:2
        for k=1:7
            Z2((i-1)*2+j,k)=p(1,k)*Y2(i,k*2+j);
        end;
    end;
end;
z=[Z;Z1;Z2];

rs=rs+1
J=-abs(det(z'*z))

```

5. Designing two additional optimal experiments

```
clear all
clc
global rs
rs=0;
load p32.mat
options=psoptimset('TolMesh',10^-8,'TolX',0.1);
[Ts,fval]=patternsearch(@(ts)maxdetTwo(ts,p),[0;0;20;8],[],[],[],[
],[0;0;8;8],[32;32;20;20],[],options);
save('TsRealTwo','Ts')
```

6. Building Z_{scaled} for three-experiment design

```
function J=maxdetThree(Ts,p)
global rs

u1=Ts(4,1);
u2=Ts(5,1);
u3=Ts(6,1);

%r=fix(250/Ts(4,1));

x01=[100-Ts(1,1),0,Ts(1,1),0,0,zeros(1,35)];
x02=[100-Ts(2,1),0,Ts(2,1),0,0,zeros(1,35)];
x03=[100-Ts(3,1),0,Ts(3,1),0,0,zeros(1,35)];

[T,X]=ode45(@(t,x)dbmodrecur(t,x,20,p),[0,15,30,60,120,240],[73.8,
0,26.2,0,0,zeros(1,35)]);
[T1,X1]=ode45(@(t,x)dbmodrecur(t,x,u1,p),(0:10:250),x01);
[T2,X2]=ode45(@(t,x)dbmodrecur(t,x,u2,p),(0:10:250),x02);
[T3,X3]=ode45(@(t,x)dbmodrecur(t,x,u3,p),(0:10:250),x03);

Y=zeros(size(X,1),16);
for i=1:8
    Y(:,2*(i-1)+1)=X(:,5*(i-1)+3)+X(:,5*(i-1)+4);
    Y(:,2*i)=X(:,5*i);
end;

Y1=zeros(size(X1,1),16);
for i=1:8
    Y1(:,2*(i-1)+1)=X1(:,5*(i-1)+3)+X1(:,5*(i-1)+4);
    Y1(:,2*i)=X1(:,5*i);
end;
Y2=zeros(size(X2,1),16);
for i=1:8
    Y2(:,2*(i-1)+1)=X2(:,5*(i-1)+3)+X2(:,5*(i-1)+4);
    Y2(:,2*i)=X2(:,5*i);
end;
Y3=zeros(size(X3,1),16);
for i=1:8
    Y3(:,2*(i-1)+1)=X3(:,5*(i-1)+3)+X3(:,5*(i-1)+4);
    Y3(:,2*i)=X3(:,5*i);
end;

for i=1:16
    if norm(Y(:,i))>0
        Y(:,i)=Y(:,i)*size(Y,1)/norm(Y(:,i));
    end;
end;

for i=1:16
    if norm(Y1(:,i))>0
        Y1(:,i)=Y1(:,i)*size(Y1,1)/norm(Y1(:,i));
    end;
end;
end;
```

```

for i=1:16
    if norm(Y2(:,i))>0
        Y2(:,i)=Y2(:,i)*size(Y2,1)/norm(Y2(:,i));
    end;
end;

for i=1:16
    if norm(Y3(:,i))>0
        Y3(:,i)=Y3(:,i)*size(Y3,1)/norm(Y3(:,i));
    end;
end;

Z=zeros(2*size(Y,1),7);

for i=1:size(Y,1)
    for j=1:2
        for k=1:7
            Z((i-1)*2+j,k)=p(1,k)*Y(i,k*2+j);
        end;
    end;
end;

Z1=zeros(2*size(Y1,1),7);

for i=1:size(Y1,1)
    for j=1:2
        for k=1:7
            Z1((i-1)*2+j,k)=p(1,k)*Y1(i,k*2+j);
        end;
    end;
end;

Z2=zeros(2*size(Y2,1),7);
for i=1:size(Y2,1)
    for j=1:2
        for k=1:7
            Z2((i-1)*2+j,k)=p(1,k)*Y2(i,k*2+j);
        end;
    end;
end;

Z3=zeros(2*size(Y3,1),7);
for i=1:size(Y3,1)
    for j=1:2
        for k=1:7
            Z3((i-1)*2+j,k)=p(1,k)*Y3(i,k*2+j);
        end;
    end;
end;

z=[Z;Z1;Z2;Z3];

rs=rs+1
J=-abs(det(z'*z))

```

7. Designing three additional optimal experiments

```
clear all
clc
global rs
rs=0;
load p32.mat
options=psoptimset('TolMesh',10^-8,'TolX',0.0001);
[Ts,fval]=patternsearch(@(ts)maxdetThree(ts,p),[0;30;25;8;8;20],[],[],[],[],[0;0;0;8;8;8],[32;32;32;20;20;20],[],options);
save('TsRealThree','Ts')
```

B.3 Parameter back-calculations for different experimental designs

1. ODEs

```
function dx=dbmodrecurws(t,x,u,p)
% x(1):Pentane-soluble, reactant
% x(2):Pentane-soluble, product
% x(3):Asphaltenes, reactant
% x(4):Asphaltenes, product
% x(5):THF-insolubles
% p(1)=kP
% p(2)=kA
% p(3)=kH
% p(4)=kC
% p(5)=m'
% p(6)=a
% p(7)=SL
% u=P_H2

dx=zeros(5,1);

if x(4)>(p(7)*(x(1)+x(2)))
    f=1;
else
    f=0;
end;

dx(1)=-p(1)*x(1);
dx(2)=(1-p(6))*p(1)*x(1)+(1-p(5))*p(2)*x(3)+p(3)*u*x(4);
dx(3)=-p(2)*x(3);
dx(4)=p(6)*p(1)*x(1)+p(5)*p(2)*x(3)-p(3)*u*x(4)-p(4)*(x(4)-
p(7)*(x(1)+x(2)))*f;
dx(5)=p(4)*(x(4)-p(7)*(x(1)+x(2)))*f;
```

2. Parameter back-calculation objective function for current experimental design

```
function J=joptex0(T,Y,u1,p_op)
global s

ob=0;
[m,n]=size(Y);

y0=Y(1,:);
x0=zeros(1,5);
x0(1,3)=y0(1,1);
x0(1,5)=y0(1,2);
x0(1,1)=100-y0(1,1)-y0(1,2);

Yt(1,:)=y0;

for i=1:m-1
    [Tt,Xt]=ode45(@ (t,x) dbmodrecurws(t,x,u1,p_op), [T(i,1)
T(i+1,1)],x0);
    x0=Xt(end,:);
    Yt(i+1,1)=Xt(end,3)+Xt(end,4);
    Yt(i+1,2)=Xt(end,5);
    clear Xt Tt
end;

W(1,1)=1;
W(1,2)=1;

s=s+1
for l=1:m
    for o=1:n
        ob=ob+(W(1,o)*(Yt(l,o)-Y(1,o))).^2;
    end;
end;
J=ob
```

3. Parameter back-calculation for current experimental design

```
clear all

global s
s=0;
load pZeroReal.mat
load Y.mat
load T.mat

u1=20;

p0=[0.08    0.18    0.0024  0.02    0.55    0.20    0.15];
options=psoptimset('MaxIter',50000,'MaxfunEvals',600000,'TolMesh',
1e-8);

[P,fval]=patternsearch(@(p_op)joptex0(T,Y,u1,p_op),p0,[],[],[],[],
zeros(1,7),ones(1,7),[],options);
p0=P;

[Tt,Xt]=ode45(@(t,x)dbmodrecurws(t,x,u1,P),0:10:250,[73.8,0,26.2,0
,0]);
Yt(:,1)=Xt(:,3)+Xt(:,4);
Yt(:,2)=Xt(:,5);

plot(Tt,Yt(1:end,1),'-g',Tt,Yt(1:end,2),'-
b',T,Y(1:end,1),'g',T,Y(1:end,2),'b');

save('/home/sshams/model2v/Jan14_2measVarib/CI on experimental
design/CI for one experiment/pZeroReal','p0')
save('pZeroReal','p0')
```

4. Parameter back-calculation objective function for one-experiment design

```

function J=joptex(T,Y,Ye,u1,u2,p_op)
global s
[Tt,Xt]=ode45(@(t,x)dbmodrecurws(t,x,u1,p_op),0:10:250,[73.5,0,26.5,0,0]);

Yt(:,1)=Xt(:,3)+Xt(:,4);
Yt(:,2)=Xt(:,5);

W=zeros(1,2);
W(1,1)=1;
W(1,2)=1;

ob=0;
for i=1:size(Ye,1)
    for j=1:2
        ob=ob+(W(1,j)*(Yt(i,j)-Ye(i,j)))^2;
    end;
end;

clear Xt Yt Tt
[m,n]=size(Y);

y0=Y(1,:);
x0=zeros(1,5);
x0(1,3)=y0(1,1);
x0(1,5)=y0(1,2);
x0(1,1)=100-y0(1,1)-y0(1,2);

Yt(1,:)=y0;

for i=1:m-1
    [Tt,Xt]=ode45(@(t,x)dbmodrecurws(t,x,u2,p_op),[T(i,1)
T(i+1,1)],x0);
    x0=Xt(end,:);
    Yt(i+1,1)=Xt(end,3)+Xt(end,4);
    Yt(i+1,2)=Xt(end,5);
    clear Xt Tt
end;

W(1,1)=1;
W(1,2)=1;

s=s+1
for l=1:m
    for o=1:n
        ob=ob+(W(1,o)*(Yt(l,o)-Y(1,o)))^2;
    end;
end;
J=ob

```

5. Parameter back-calculation for one-experiment design

```
clc
clear all

global s
s=0;
%load pOneReal.mat
load p32.mat
load Y.mat
load T.mat
u1=8;
u2=20;
[Te,Xe]=ode45(@(t,x)dbmodrecurws(t,x,u1,p),0:10:250,[73.5,0,26.5,0,0]);

Ye(:,1)=Xe(:,3)+Xe(:,4);
Ye(:,2)=Xe(:,5);

p0=[0.08 0.18 0.0024 0.02 0.55 0.20 0.15];
options=psoptimset('MaxIter',50000,'MaxfunEvals',600000,'TolMesh',1e-8);

[P,fval]=patternsearch(@(p_op)joptex(T,Y,Ye,u1,u2,p_op),p0,[],[],[],[],[],zeros(1,7),ones(1,7),[],options);
p0=P;

[Tt,Xt]=ode45(@(t,x)dbmodrecurws(t,x,u1,P),0:10:250,[73.5,0,26.5,0,0]);
Yt(:,1)=Xt(:,3)+Xt(:,4);
Yt(:,2)=Xt(:,5);

plot(Tt,Yt(1:end,1),'-g',Tt,Yt(1:end,2),'-b',Te,Ye(1:end,1),'g',Te,Ye(1:end,2),'b');

save('/home/sshams/model2v/Jan14_2measVarib/CI on experimental design/CI for one experiment/pOneRealfmin','p0')
save('pOneRealfmin','p0')
```

6. Parameter back-calculation objective function for two-experiment design

```

function J=j2exp(T,Y,Ye1,Ye2,u1,u2,u3,p_op)
global s
[Tt1,Xt1]=ode45(@(t,x)dbmodrecurws(t,x,u1,p_op),0:10:250,[68,0,32,
0,0]);
W=zeros(1,2);
W(1,1)=1;
W(1,2)=1;

Yt1(:,1)=Xt1(:,3)+Xt1(:,4);
Yt1(:,2)=Xt1(:,5);

ob=0;
for i=1:size(Ye1,1)
    for j=1:2
        ob=ob+(W(1,j)*(Yt1(i,j)-Ye1(i,j)))^2;
    end;
end;

[Tt2,Xt2]=ode45(@(t,x)dbmodrecurws(t,x,u2,p_op),0:10:250,[73.5,0,2
6.5,0,0]);

Yt2(:,1)=Xt2(:,3)+Xt2(:,4);
Yt2(:,2)=Xt2(:,5);

for i=1:size(Ye2,1)
    for j=1:2
        ob=ob+(W(1,j)*(Yt2(i,j)-Ye2(i,j)))^2;
    end;
end;

[m,n]=size(Y);

y0=Y(1,:);
x0=zeros(1,5);
x0(1,3)=y0(1,1);
x0(1,5)=y0(1,2);
x0(1,1)=100-y0(1,1)-y0(1,2);

Yt(1,:)=y0;

for i=1:m-1
    [Tt,Xt]=ode45(@(t,x)dbmodrecurws(t,x,u3,p_op),[T(i,1)
T(i+1,1)],x0);
    x0=Xt(end,:);
    Yt(i+1,1)=Xt(end,3)+Xt(end,4);
    Yt(i+1,2)=Xt(end,5);
    clear Xt Tt
end;
for l=1:m
    for o=1:n
        ob=ob+(W(1,o)*(Yt(1,o)-Y(1,o)))^2;
    end;
end;

```

```
end;  
s=s+1  
J=ob  
clc
```

7. Parameter back-calculation for two-experiment design

```
clear all

global s
s=0;
%load pTwoReal.mat
load p32.mat
load Y.mat
load T.mat
u1=20;
u2=8;
u3=20;
p0=[0.08    0.18    0.048/20    0.02    0.55    0.20    0.15];
%p0=P;
[Te1,Xe1]=ode45(@(t,x)dbmodrecurws(t,x,u1,p),0:10:250,[68,0,32,0,0
1]);

Ye1(:,1)=Xe1(:,3)+Xe1(:,4);
Ye1(:,2)=Xe1(:,5);
[Te2,Xe2]=ode45(@(t,x)dbmodrecurws(t,x,u2,p),0:10:250,[73.5,0,26.5
,0,0]);

Ye2(:,1)=Xe2(:,3)+Xe2(:,4);
Ye2(:,2)=Xe2(:,5);

options=psoptimset('MaxIter',150000,'MaxfunEvals',200000);
[P,fval]=patternsearch(@(p_op)j2exp(T,Y,Ye1,Ye2,u1,u2,u3,p_op),p0,
[],[],[],[],zeros(1,7),ones(1,7),[],options);

[Tt1,Xt1]=ode45(@(t,x)dbmodrecurws(t,x,u1,P),0:10:250,[68,0,32,0,0
1]);
Yt1(:,1)=Xt1(:,3)+Xt1(:,4);
Yt1(:,2)=Xt1(:,5);

[Tt2,Xt2]=ode45(@(t,x)dbmodrecurws(t,x,u2,P),0:10:250,[73.5,0,26.5
,0,0]);
Yt2(:,1)=Xt2(:,3)+Xt2(:,4);
Yt2(:,2)=Xt2(:,5);

plot(Tt1,Yt1(1:end,1),'-g',Tt1,Yt1(1:end,2),'-
b',Te1,Ye1(1:end,1),'g',Te1,Ye1(1:end,2),'b');
figure; plot(Tt2,Yt2(1:end,1),'-g',Tt2,Yt2(1:end,2),'-
b',Te2,Ye2(1:end,1),'g',Te2,Ye2(1:end,2),'b');

save('pTwoReal1','P')
save('/home/sshams/model2v/Jan14_2measVarib/CI on experimental
design/CI for 2 experiments/pTwoReal1','P')
```

8. Parameter back-calculation objective function for three-experiment design

```
function J=j3exp(T,Y,Ye1,Ye2,Ye3,u1,u2,u3,u4,p_op)
global s
[Tt1,Xt1]=ode45(@(t,x)dbmodrecurws(t,x,u1,p_op),0:10:250,[68,0,32,
0,0]);

W=zeros(1,2);
W(1,1)=1;
W(1,2)=1;

Yt1(:,1)=Xt1(:,3)+Xt1(:,4);
Yt1(:,2)=Xt1(:,5);

ob=0;
for i=1:size(Ye1,1)
    for j=1:2
        ob=ob+(W(1,j)*(Yt1(i,j)-Ye1(i,j)))^2;
    end;
end;

[Tt2,Xt2]=ode45(@(t,x)dbmodrecurws(t,x,u2,p_op),0:10:250,[68.4,0,3
1.6,0,0]);

W(1,1)=1;
W(1,2)=1;

Yt2(:,1)=Xt2(:,3)+Xt2(:,4);
Yt2(:,2)=Xt2(:,5);

for i=1:size(Ye2,1)
    for j=1:2
        ob=ob+(W(1,j)*(Yt2(i,j)-Ye2(i,j)))^2;
    end;
end;

[Tt3,Xt3]=ode45(@(t,x)dbmodrecurws(t,x,u3,p_op),0:10:250,[100,0,0,
0,0]);

W(1,1)=1;
W(1,2)=1;

Yt3(:,1)=Xt3(:,3)+Xt3(:,4);
Yt3(:,2)=Xt3(:,5);

for i=1:size(Ye3,1)
    for j=1:2
        ob=ob+(W(1,j)*(Yt3(i,j)-Ye3(i,j)))^2;
    end;
end;

[m,n]=size(Y);

y0=Y(1,:);
```

```

x0=zeros(1,5);
x0(1,3)=y0(1,1);
x0(1,5)=y0(1,2);
x0(1,1)=100-y0(1,1)-y0(1,2);

Yt(1,:)=y0;

for i=1:m-1
    [Tt,Xt]=ode45(@ (t,x)dbmodrecurws(t,x,u4,p_op),[T(i,1)
T(i+1,1)],x0);
    x0=Xt(end,:);
    Yt(i+1,1)=Xt(end,3)+Xt(end,4);
    Yt(i+1,2)=Xt(end,5);
    clear Xt Tt
end;

W(1,1)=1;
W(1,2)=1;

for l=1:m
    for o=1:n
        ob=ob+(W(1,o)*(Yt(1,o)-Y(1,o))).^2;
    end;
end;
s=s+1
J=ob

```

9. Parameter back-calculation for three-experiment design

```
clc
clear all

global s
s=0;
%load pThreeRealfmin.mat
load p32.mat
load Y.mat
load T.mat
u1=19.9; u2=8; u3=20; u4=20;
p0=[0.08 0.18 0.048/20 0.02 0.55 0.20 0.15];
%p0=P;
[Te1,Xe1]=ode45(@(t,x)dbmodrecurws(t,x,u1,p),0:10:250,[68,0,32,0,0
]);

Ye1(:,1)=Xe1(:,3)+Xe1(:,4);
Ye1(:,2)=Xe1(:,5);

[Te2,Xe2]=ode45(@(t,x)dbmodrecurws(t,x,u2,p),0:10:250,[68.4,0,31.6
,0,0]);

Ye2(:,1)=Xe2(:,3)+Xe2(:,4);
Ye2(:,2)=Xe2(:,5);

[Te3,Xe3]=ode45(@(t,x)dbmodrecurws(t,x,u3,p),0:10:250,[100,0,0,0,0
]);

Ye3(:,1)=Xe3(:,3)+Xe3(:,4);
Ye3(:,2)=Xe3(:,5);

options=psoptimset('MaxIter',150000,'MaxfunEvals',200000,'TolMesh'
,1e-6);
[P,fval]=patternsearch(@(p_op)j3exp(T,Y,Ye1,Ye2,Ye3,u1,u2,u3,u4,p_
op),p0,[1,[1],[1],[1],[1],zeros(1,7),ones(1,7)],[1],options);

[Tt1,Xt1]=ode45(@(t,x)dbmodrecurws(t,x,u1,P),0:10:250,[68,0,32,0,0
]);
Yt1(:,1)=Xt1(:,3)+Xt1(:,4);
Yt1(:,2)=Xt1(:,5);

[Tt2,Xt2]=ode45(@(t,x)dbmodrecurws(t,x,u2,P),0:10:250,[68.4,0,31.6
,0,0]);
Yt2(:,1)=Xt2(:,3)+Xt2(:,4);
Yt2(:,2)=Xt2(:,5);

[Tt3,Xt3]=ode45(@(t,x)dbmodrecurws(t,x,u3,P),0:10:250,[100,0,0,0,0
]);
Yt3(:,1)=Xt3(:,3)+Xt3(:,4);
Yt3(:,2)=Xt3(:,5);

plot(Tt1,Yt1(1:end,1),'-g',Tt1,Yt1(1:end,2),'-
b',Te1,Ye1(1:end,1),'-g',Te1,Ye1(1:end,2),'-b');
```

```
figure; plot(Tt2,Yt2(1:end,1),'-g',Tt2,Yt2(1:end,2),'-
b',Te2,Ye2(1:end,1),'.g',Te2,Ye2(1:end,2),'.b');
figure; plot(Tt3,Yt3(1:end,1),'-g',Tt3,Yt3(1:end,2),'-
b',Te3,Ye3(1:end,1),'.g',Te3,Ye3(1:end,2),'.b');

save('pThreeRealfmin1','P')
save('/home/sshams/model2v/Jan14_2measVarib/CI on experimental
design/CI for 3 experiments/pThreeRealfmin1','P')
```