



Nonlinear Model Predictive Control for Oil Reservoirs

Ritschel, Tobias Kasper Skovborg

Publication date:
2019

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

[Link back to DTU Orbit](#)

Citation (APA):
Ritschel, T. K. S. (2019). *Nonlinear Model Predictive Control for Oil Reservoirs*. Technical University of Denmark. DTU Compute PHD-2018 Vol. 485

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Ph.D. Thesis

Nonlinear Model Predictive Control for Oil Reservoirs

Tobias Kasper Skovborg Ritschel

Department of Applied Mathematics and Computer Science
Technical University of Denmark
DK-2800 Kgs. Lyngby
Denmark

PHD-2018-485

TOBIAS KASPER SKOVBOG RITSCHER

Nonlinear Model Predictive Control for Oil Reservoirs

Ph.D. Thesis, Aug. 2018

Supervisors:

John Bagterp Jørgensen,

Associate Professor at the Department of Mathematics and Computer Science

Andrea Capolei,

Postdoctoral Researcher at the Department of Mathematics and Computer Science

Technical University of Denmark, Kgs. Lyngby - 2018

Nonlinear Model Predictive Control for Oil Reservoirs

This thesis was prepared by:

Tobias Kasper Skovborg Ritschel

Supervisors:

John Bagterp Jørgensen,

Associate Professor at the Department of Mathematics and Computer Science

Andrea Capolei,

Postdoctoral Researcher at the Department of Mathematics and Computer Science

Department of Applied Mathematics and Computer Science

Technical University of Denmark

Richard Petersens Plads, Building 324

DK-2800 Kgs. Lyngby

Denmark

Tel: +45 4525 3031

E-mail: compute@compute.dtu.dk

Project period: Aug. 2015 - Aug. 2018

Degree: Ph.D.

Field: Applied Mathematics

Remarks: This thesis is submitted in partial fulfillment of the requirements for receiving the Ph.D. degree in applied mathematics at the Department of Applied Mathematics and Computer Science, Technical University of Denmark.

Serial number: PHD-2018-485

ISSN: 0909-3192

Copyright: © Tobias Kasper Skovborg Ritschel 2018

Acknowledgments

I would like to thank my supervisor John Bagterp Jørgensen for his advice, teaching, and support throughout this project. Furthermore, I would like to thank my co-supervisor Andrea Capolei as well as Jozsef Gaspar, Mathias Sorgenfri Lorenz, and Sabine Fie Hansen for our collaborations. I would also like to thank Lasse Hjuler Christiansen and Steen Hørsholt as well as my other colleagues at the Scientific Computing section at DTU Compute, in CERE, and in the OPTION project for the many interesting discussions. Finally, I would like to thank Line Græse Fleron for providing feedback on the introduction and my friends and family for their support.

Preface

This thesis was prepared at the Department of Applied Mathematics and Computer Science (DTU Compute) and at the Center for Energy Resources Engineering (CERE), Technical University of Denmark (DTU), in partial fulfillment of the requirements for receiving the Ph.D. degree in applied mathematics.

The work presented in this thesis was carried out in the period from August 15th 2015 to August 14th 2018 under the supervision of Associate Professor John Bagterp Jørgensen and Postdoctoral Researcher Andrea Capolei.

This project is funded by Innovation Fund Denmark in the OPTION project (63-2013-3).

Tobias Kasper Skovborg Ritschel

Kgs. Lyngby, Aug. 2018

Summary (in english)

Oil remains the world's leading fuel, and it accounted for a third of the global energy consumption in 2016. Oil is mainly used as a source of energy, but it is also used for non-energy purposes, e.g. for road surfaces, lubricants, and in the chemical industry. Furthermore, the global demand for oil is expected to increase towards 2040 where it is predicted to account for 31% of the world's energy consumption.

The subject of this thesis is nonlinear model predictive control (NMPC) for closed-loop reservoir management (CLRM). The purpose of NMPC for oil reservoirs management is to compute a field-wide closed-loop feedback control strategy (i.e. an oil production strategy) that optimizes a long-term financial measure of the oil production process, e.g. the total oil recovery or the net present value over the reservoir life-time.

More specifically, this thesis is concerned with models and algorithms for NMPC of thermal and isothermal compositional oil recovery processes. Two main principles are used to model such processes: 1) mass and energy conservation, and 2) phase equilibrium. The conservation of energy relates to the first law of thermodynamics, and the phase equilibrium relates to the second law of thermodynamics (i.e. the entropy of a closed system in equilibrium is maximal). The phase equilibrium problem that is relevant to thermal reservoir flow models is the UV flash which is a direct statement of the second law of thermodynamics. For isothermal reservoir flow models, the relevant phase equilibrium problem is the VT flash. The condition of maximal entropy does not apply directly to isothermal systems because they are not closed. Instead, the Helmholtz energy is minimal for isothermal systems in equilibrium. In this work, we formulate the phase equilibrium problems as equality constrained optimization problems and the phase equilibrium conditions as the corresponding first order optimality (or Karush-Kuhn-Tucker) conditions. Consequently, the phase equilibrium conditions are a set of algebraic equations. The conservation equations are a set of coupled partial differential equations. We use the method of lines to solve these partial differential-algebraic equations, and we discretize the partial differential equations with a finite volume method. The result is a set of differential equations, and combined with the phase equilibrium conditions, the model equations are a set of differential-algebraic equations (DAEs) which are in a specific semi-explicit form.

In this work, we describe algorithms for 1) simulation, 2) state estimation, 3) dynamic optimization, and 4) NMPC of DAEs in this specific semi-explicit form. Numerical methods for simulation, i.e. for numerical solution of initial value problems (IVPs), are central to the state estimation algorithms and the dynamic optimization algorithm (and therefore also to the NMPC algorithm) considered in this thesis. We consider the numerical solution of both deterministic and stochastic IVPs that involve DAEs in the semi-explicit form. We present two approaches for the numerical solution of the deterministic IVPs: 1) a simultaneous approach and 2) a nested approach. Both approaches use Euler's implicit method. In the simultaneous approach, the differential equations

and the algebraic equations are solved simultaneously. In the nested approach, the solution of the algebraic equations is nested into the solution of the differential equations. We present one approach for the numerical solution of the stochastic IVPs. It is a simultaneous approach, and it uses a semi-implicit discretization scheme. We consider the extended Kalman filter (EKF), the unscented Kalman filter (UKF), a particle filter (PF), and the ensemble Kalman filter (EnKF) for state estimation of continuous-discrete DAE systems in the semi-explicit form. Furthermore, we describe an algorithm for gradient-based numerical solution of dynamic optimization problems that involve DAEs in the semi-explicit form. The algorithm uses 1) the single-shooting method and 2) the discrete adjoint method for the computation of gradients. Finally, we describe an NMPC algorithm which combines either of the four state estimation algorithms with the gradient-based dynamic optimization algorithm.

It is natural to model other dynamic phase equilibrium processes using DAEs in the specific semi-explicit form that we consider. Therefore, the above algorithms are relevant for dynamic phase equilibrium processes in general.

We implement and test the algorithms on a small-scale flash separation process. The model of this process consists of mass and energy conservation equations, and the relevant phase equilibrium problem is the UV flash. Therefore, this flash separation process is representative of the thermal and compositional reservoir flow process. For the flash separation process, we consider Matlab implementations of the state estimation algorithms, Matlab and C implementations of the dynamic optimization algorithm, and a mixed Matlab and C implementation of the NMPC algorithm. For the thermal and the isothermal compositional reservoir flow models, we consider a C/C++ implementation of the dynamic optimization algorithm.

In this work, we develop an open-source thermodynamic software library called ThermoLib which we use to evaluate the thermodynamic functions in the reservoir flow models and in the model of the flash separation process. ThermoLib is available at www.psetools.org. The thermodynamic model in ThermoLib is based on data and correlations from the DIPPR database and on cubic equations of state. ThermoLib provides Matlab and C routines for evaluating the enthalpy, entropy, and volume of 1) ideal gas mixtures, 2) ideal liquid mixtures, and 3) nonideal mixtures as functions of temperature, pressure, and mixture composition (in moles). All other thermodynamic functions can be computed from the enthalpy, entropy, and volume using fundamental thermodynamic relations. The main novelty of ThermoLib is that its routines also evaluate the first and second order derivatives of the thermodynamic functions with respect to the temperature, pressure, and composition (in moles). The expressions for these derivatives are derived analytically.

This thesis consists of a summary report and a collection of twelve research papers and two technical reports written in the period from August 2015 to August 2018: 1) one paper is published in Computers and Chemical Engineering, 2) one paper is submitted to Journal of Process Control, 3) nine papers are published in conference proceedings, and 4) one paper is in preparation to be submitted.

Summary (in danish)

For at undgå meningsforstyrrende oversættelser har vi i dette resumé bibeholdt nogle af de engelske betegnelser. Vi skriver disse i kursiv. Derudover betegner vi ved strømningsmodeller, modeller af strømningerne i et reservoir eller på engelsk 'reservoir flow models'.

Olie er fortsat verdens førende brændstof, og en tredjedel af verdens energiforbrug blev i 2016 dækket af olie. Olie bruges hovedsageligt som energikilde, men bliver også brugt til andre formål, for eksempel til vejbelægning, smøremidler, eller i den kemiske industri. Desuden forventes den globale efterspørgsel på olie at stige hen imod 2040, hvor det forventes at dække 31% af verdens energiforbrug.

Denne afhandling omhandler ikke-lineær model prædiktiv regulering (NMPC) til administration af olieindvinding i lukket sløjfe (CLRM). Formålet med NMPC til administration af olieindvinding er at beregne en *feedback*-baseret kontrol strategi for hele oliefeltet (altså en strategi for olieindvindingen) der optimerer et langsigtet økonomisk aspekt af olieindvindingsprocessen, for eksempel den totale mængde af produceret olie eller nutidsværdien beregnet over hele oliefeltets levetid.

Mere specifikt beskæftiger denne afhandling sig med modeller og algoritmer til NMPC af termiske og isoterme kompositionelle olieindvindingsprocesser. Der er to principper der bruges til at modellere sådanne processer: 1) masse- og energibevarelse, og 2) faseligevægt. Energibevarelse er relateret til termodynamikkens første hovedsætning, og faseligevægt er relateret til termodynamikkens anden hovedsætning (som siger at entropien i et lukket system i ligevægt er maksimal). Det faseligevægtsproblem der er relevant for termiske strømningsmodeller betegnes som et *UV-flash* hvilket er direkte baseret på termodynamikkens anden hovedsætning. For isoterme strømningsmodeller betegnes det relevante faseligevægtsproblem som et *VT-flash*. Betingelsen om maksimal entropi gør sig ikke direkte gældende for isoterme systemer fordi de ikke er lukkede. I stedet er Helmholtz energi minimal for isoterme systemer i ligevægt. I dette projekt formulerer vi faseligevægtsproblemerne som optimeringsproblemer og faseligevægtsbetingelserne som de tilhørende førsteordens optimalitetsbetingelser (eller Karush-Kuhn-Tucker betingelser). Det betyder at faseligevægtsbetingelserne er et sæt af algebraiske ligninger. Bevarelsesligningerne er et sæt af koblede partielle differentiaalligninger. Vi bruger en metode kaldet *method of lines* til at løse disse partielle differentialalgebraiske ligninger, og vi diskretiserer de partielle differentiaalligninger med en *finite volume* metode. Resultatet er et sæt af differentiaalligninger som, kombineret med faseligevægtsbetingelserne, udgør et sæt af differentialalgebraiske ligninger (DAEer) der er på en specifik semi-eksplicit form.

I dette projekt beskriver vi algoritmer til 1) simulering, 2) tilstandsestimering, 3) dynamisk optimering, og 4) NMPC af DAEer på denne specifikke semi-eksplicite form. Numeriske metoder til simulering, altså til numerisk løsning af begyndelsesværdiproblemer (IVPer), er centrale for tilstandsestimeringsalgoritmerne og den dynamiske optimeringsalgoritme (og derfor også for den NMPC-algoritme) som vi beskæftiger os med i denne afhandling. Vi beskæftiger os både med

numerisk løsning af deterministiske og stokastiske IVP'er som involverer DAE'er på den semi-eksplicitte form. Vi præsenterer to metoder til at løse de deterministiske IVP'er numerisk: 1) en simultan metode og 2) en *nested* metode. Begge metoder bruger Eulers implicitte metode. I den simultane metode løses differentially ligningerne og de algebraiske ligninger samtidigt. I den *nestede* metode *nestes* løsningen af de algebraiske ligninger ind i løsningen af differentially ligningerne. Vi præsenterer en enkelt metode til at løse de stokastiske IVP'er numerisk. Det er en simultan metode, og den bruger en semi-implicit diskretisering. Vi bruger det udvidede Kalman filter (EKF), det *unscented* Kalman filter (UKF), et partikelfilter (PF), og ensemble Kalman filteret (EnKF) til tilstandsestimering af kontinuert-diskrete DAE systemer på den semi-eksplicitte form. Derudover beskriver vi en algoritme til gradient-baseret numerisk løsning af dynamiske optimeringsproblemer der involverer DAE'er på den semi-eksplicitte form. Algoritmen bruger 1) *single-shooting* metoden og 2) den diskrete *adjoint* metode til beregning af gradienter. Som det sidste beskriver vi en NMPC-algoritme som kombinerer en af de fire tilstandsestimeringsalgoritmer med den gradient-baserede dynamiske optimeringsalgoritme.

Det er naturligt at modellere andre dynamiske faseligevægtsprocesser med DAE'er på den specifikke semi-eksplicitte form som vi beskæftiger os med. Derfor er de ovenstående algoritmer relevante for dynamiske faseligevægtsprocesser generelt.

Vi implementerer og tester algoritmerne på en små-skala *flash* separationsproces. Modellen af denne proces, består af masse- og energibevarelsesligninger, og det relevante faseligevægtsproblem er UV-*flash* problemet. Denne *flash* separationsproces er derfor repræsentativ for den termiske og kompositionelle strømningssmodel. Vi implementerer tilstandsestimeringsalgoritmerne i Matlab, den dynamiske optimeringsalgoritme i Matlab og C, og NMPC-algoritmen i både Matlab og C (altså en blandet implementering) for *flash* separationsprocessen. Vi implementerer den dynamiske optimeringsalgoritme i C/C++ for den termiske og den isothermiske kompositionelle strømningssmodel.

I dette projekt udvikler vi et *open-source* termodynamisk software bibliotek kaldet ThermoLib som vi bruger til at evaluere de termodynamiske funktioner der optræder i strømningssmodellerne og i modellen af *flash* separationsprocessen. ThermoLib er tilgængelig på www.psetools.org. Den termodynamiske model i ThermoLib er baseret på data og korrelationer fra DIPPR-databasen og på kubiske tilstandsligninger. ThermoLib indeholder Matlab og C rutiner til at evaluere entalpi, entropi, og volumen af 1) ideelle gasblandinger, 2) ideelle væskeblandinger, og 3) ikke-ideelle blandinger som funktioner af temperatur, tryk, og blandingskomposition (i mol). Alle andre termodynamiske funktioner kan beregnes fra entalpi, entropi, og volumen ved hjælp af fundamentale termodynamiske relationer. ThermoLibs primære bidrag er at dets rutiner også evaluerer de første- og andenordens afledede af de termodynamiske funktioner med hensyn til temperatur, tryk, og blandingskomposition (i mol). Udtrykkene for disse afledede er udledt analytisk.

Denne afhandling består af en sammenfattende rapport og en samling af tolv forskningsartikler og to tekniske rapporter skrevet i perioden august 2015 til august 2018: 1) en artikel er udgivet i Computers and Chemical Engineering, 2) en artikel er indsendt til Journal of Process Control, 3) ni artikler er udgivet som conferenceartikler, og 4) en artikel er endnu ikke indsendt.

Contents

	Page
Preface	ix
Summary (in english)	xi
Summary (in danish)	xiii
List of figures	xix
List of tables	xxi
List of abbreviations	xxiii
1 Introduction	1
1.1 Introduction and motivation	1
1.2 Literature review	6
1.3 Main objectives and contributions	9
1.4 Outline of the thesis	12
I Models	15
2 Thermodynamic model	17
2.1 Brief description	17
2.2 DIPPR correlations	17
2.3 Ideal gas mixtures	19
2.4 Ideal liquid mixtures	20
2.5 Nonideal mixtures	21
2.6 Summary	22
3 Flash separation	23
3.1 Brief description	23
3.2 Phase equilibrium	23
3.3 Mass and energy conservation	24
3.4 Semi-explicit differential-algebraic form	25
3.5 Summary	26
4 Reservoir flow	27
4.1 Brief description	27
4.2 Thermal and compositional reservoir flow model	28

4.3	Isothermal and compositional reservoir flow model	34
4.4	Summary	35
II	Algorithms	37
5	Simulation	39
5.1	Problem formulation	39
5.2	Numerical solution of deterministic initial value problems	40
5.3	Numerical solution of stochastic initial value problems	43
5.4	Summary	43
6	State estimation	45
6.1	Problem formulation	45
6.2	The extended Kalman filter	46
6.3	The unscented Kalman filter	49
6.4	The particle filter	52
6.5	The ensemble Kalman filter	54
6.6	Summary	55
7	Dynamic optimization	57
7.1	Problem formulation	57
7.2	The dynamic optimization algorithm	58
7.3	Summary	62
8	Nonlinear model predictive control	63
8.1	Problem formulation	63
8.2	The nonlinear model predictive control algorithm	64
8.3	Summary	65
III	Examples	67
9	Numerical examples – flash separation	69
9.1	State estimation	69
9.2	Dynamic optimization	71
9.3	Nonlinear model predictive control	80
9.4	Summary	82
10	Numerical examples – waterflooding	85
10.1	Dynamic optimization	85
10.2	Summary	87
IV	Conclusions	91
11	Conclusions	93
12	Suggestions for future work	97
	Bibliography	101

V	Appendix	111
A	First order optimality conditions	113
B	Description of workstation	115
C	Paper I	117
D	Paper II	125
E	Paper III	133
F	Paper IV	141
G	Paper V	149
H	Paper VI	167
I	Paper VII	177
J	Paper VIII	185
K	Paper IX	193
L	Paper X	203
M	Paper XI	217
N	Paper XII	231
O	Technical Report I	239
P	Technical Report II	267

List of figures

1.1	An illustration of offshore oil production facilities.	2
1.2	An illustration of the interaction between state estimation, dynamic optimization, and the process in NMPC.	3
1.3	An illustration of CLRM.	4
1.4	An overview of the papers written during this Ph.D. project.	14
3.1	A schematic of the flash separator.	24
4.1	An illustration of waterflooding.	28
4.2	An illustration of the fluid phases (water, oil, and gas) and the rock in a grid cell. . .	29
7.1	A sketch of the continuous states, $x(t)$, which are implicit functions of the parametrized manipulated inputs, $\{u_k\}_{k=0}^{N-1}$, in the single-shooting approach.	59
8.1	An illustration of the interaction between state estimation, dynamic optimization, and the process in NMPC (including variables).	65
9.1	The true values and the (filtered) estimates of the state variables (i.e. the internal energy and the total composition in moles), the temperature, and the pressure.	71
9.2	Estimates of the total mole fractions, the vapor-liquid mole fractions, and the vapor fraction corresponding to the (filtered) state estimates computed with the PF.	72
9.3	Tracking of temperature, pressure, and liquid volume setpoints using 1) an optimized strategy and 2) a reference strategy.	74
9.4	The total mole fractions and the vapor-liquid mole fractions of the mixture in the separator obtained with the optimized tracking strategy.	74
9.5	Flash separation of a mixture consisting of four hydrocarbons and hydrogen sulfide using 1) an optimized control strategy and 2) a reference strategy.	76
9.6	The total mole fractions and the vapor-liquid mole fractions of the mixture in the separator corresponding to the optimized control strategy.	77
9.7	Closed-loop simulation of economical NMPC for disturbance rejection in the flash separation process.	83
9.8	The estimated and the true feed temperature, and the manipulated inputs.	83
10.1	The permeability field and the locations of the injectors and producers.	86
10.2	The optimized injector and producer BHPs and the cumulative volume of the injected water, the produced oil, and the produced gas.	87

10.3 The oil and gas saturations, the pressure, and the temperature for the optimized thermal waterflooding strategy. 88

10.4 The oil and gas saturations and the pressure for the optimized isothermal waterflooding strategy. 89

List of tables

9.1	Values of 1) the parameters and inputs in the model of the flash separation process and 2) the parameters in the state estimation algorithms.	70
9.2	Accuracy of the EKF, the UKF, the PF, and the EnKF and average computation times of the measurement-updates and the time-updates in the algorithms.	71
9.3	Key characteristics of the tracking-type problem and the economical dynamic optimization problem.	77
9.4	Key performance indicators of a Matlab implementation of the dynamic optimization algorithm for the numerical solution of the tracking-type problem and the economical dynamic optimization problem.	78
9.5	The absolute computation times of 1) evaluating thermodynamic properties and their derivatives and 2) solving $Ax = b$ for x where A is dense.	79
9.6	The absolute computation times and the relative computation times of solving the tracking-type problem and the economical dynamic optimization problem using the dynamic optimization algorithm (with the simultaneous approach).	80
9.7	Values of 1) the parameters and the inputs in the model of the flash separation process and 2) the parameters in the NMPC algorithm.	82
10.1	Key characteristics of the two dynamic optimization problems and KPIs of the dynamic optimization algorithm.	89

List of abbreviations

BFGS	...	Broyden-Fletcher-Goldfarb-Shanno (algorithm)
BHP	...	Bottom-hole pressure
CAS	...	Cryogenic air separation
CLRM	...	Closed-loop reservoir management
DAE	...	Differential-algebraic equation
DUNE	...	Distributed and Unified Numerics Environment
EIA	...	(U.S.) Energy Information Administration
EKF	...	Extended Kalman filter
EnKF	...	Ensemble Kalman filter
EU	...	European Union
GCC	...	GNU Compiler Collection
GMRES	...	Generalized minimal residual (method)
GPRS	...	(Stanford's) General Purpose Reservoir Simulator
ILU	...	Incomplete LU (factorization)
IVP	...	Initial value problem
KPI	...	Key performance indicator
MKL	...	(Intel) Math Kernel Library
MPC	...	Model predictive control
MRST	...	Matlab Reservoir Simulation Toolbox
NLP	...	Nonlinear program
NMPC	...	Nonlinear model predictive control
OPEC	...	Organization of the Petroleum Exporting Countries
PF	...	Particle filter
SAGD	...	Steam-assisted gravity drainage
SLQP	...	Sequential linear-quadratic programming (method)
SPSA	...	Simultaneous perturbation stochastic approximation (method)
SQP	...	Sequential quadratic programming (method)
UKF	...	Unscented Kalman filter
ZOH	...	Zero-order-hold (parametrization)

Introduction

The subject of this thesis is nonlinear model predictive control (NMPC) for oil reservoirs management. The purpose of NMPC is to compute a closed-loop feedback control strategy for optimizing the economics of a dynamical process (or for setpoint tracking). In the case of oil recovery processes, the objective of NMPC is to optimize a long-term financial measure of the recovery process, e.g. the amount of oil produced or the net present value over the life-time of the reservoir. NMPC algorithms require a model of the dynamical process. In this work, we consider thermodynamically rigorous thermal (varying temperature) and isothermal (constant temperature) compositional models of oil reservoir flow. In particular, we focus on the formulation of the phase equilibrium problems involved in these models. We describe both NMPC and reservoir flow models in more detail in this chapter.

In Section 1.1, we provide a motivation for research on improving the economics of oil recovery processes as well as a brief introduction to 1) oil production, 2) NMPC, 3) reservoir flow models, and 4) phase equilibrium. In Section 1.2, we review the recent literature on subjects that are relevant to NMPC for oil reservoirs. Next, we describe the objectives of this thesis and the main contributions of this work in Section 1.3. Finally, we outline the structure of the thesis and briefly discuss the content of each chapter in Section 1.4.

1.1 Introduction and motivation

Motivation. Oil remains the world's leading fuel, and in 2016, it accounted for a third of the global energy consumption [23]. Oil is mainly used as a source of energy, but it is also used for other purposes. For instance, 14.5% of the EU oil consumption in 2015 was for non-energy purposes, e.g. bitumen for road surfaces, lubricants for the reduction of friction, and various usage in the chemical industry [53]. The U.S. Energy Information Administration (EIA) states that renewables and nuclear energy are the fastest growing sources of energy (whereas the use of coal is stagnant) [172]. Despite of this, EIA also projects that liquid fuels (mainly oil) will account for 31% of the world's energy consumption in 2040 whereas liquid fuels accounted for 33% in 2015. Similarly, projections by the Organization of the Petroleum Exporting Countries (OPEC) indicate that the global oil demand will increase from 95.4 million barrels per day in 2016 to 111.1 million barrels per day in 2040 [117].

In conclusion, it is expected that the global demand for oil will remain significant for several decades. This motivates research on methods that improve the economics and total recovery in oil production processes. Control software based on NMPC algorithms is a cost-efficient approach for reaching this goal because it is primarily used to better utilize existing recovery techniques (e.g. waterflooding or enhanced oil recovery techniques).

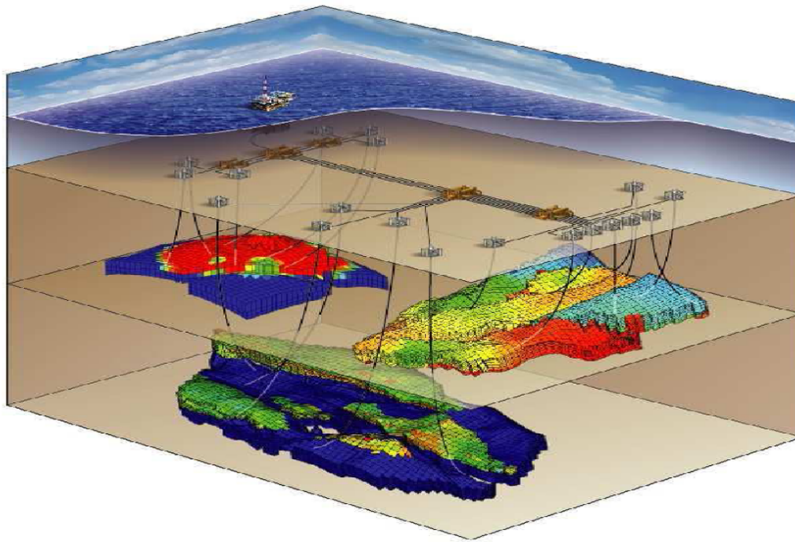


Fig. 1.1: An illustration of offshore oil production facilities. The colors illustrate the heterogeneity of the petrophysical properties of the reservoir rock (e.g. the porosity or the permeability). This figure originally appeared in [176].

Oil production. Oil is located in porous subsurface reservoir rock, and it is produced (i.e. brought to surface) using production wells that are drilled into the reservoir as illustrated in Fig. 1.1. Oil production involves up to three stages:

- a) Before production begins, the reservoir pressure is high. In the primary stage of recovery, this initial pressure is used to push the reservoir fluid up to the surface.
- b) During the primary stage, the reservoir pressure drops because the reservoir fluid is removed from the reservoir. In the secondary stage, injection wells are drilled into the reservoir and water is injected in order to maintain a high pressure.
- c) The tertiary stage of recovery involves enhanced oil recovery techniques, e.g. chemical, biological, or thermal injection. The purpose is to mobilize and recover the oil that remains after the secondary stage.

After the oil has been recovered, it is refined to consumer products (using distillation), e.g. 1) fuels (such as gasoline and kerosene), 2) bitumen for road surfaces and waterproofing products, and 3) chemicals used to make plastics and pharmaceuticals. Depending on the specific production scenario, it may be uneconomical to implement the secondary and tertiary stages. In this work, we are concerned with NMPC of the secondary stage of recovery. However, NMPC is equally applicable to all three stages.

There are several aspects of the reservoir flow dynamics that are important to take into account during production planning. The petrophysical properties of the rock, e.g. porosity and permeability, can vary throughout the reservoir. The porosity is the fraction of a given volume that is available for fluid, and the permeability is a measure of the reservoir rock's ability to transmit this fluid. The permeability depends on several aspects of the pores in the rock, e.g. on how well-connected they are. Both the porosity and the permeability have a significant effect on the fluid flow, and if the rock is very heterogeneous, there is a risk of trapping the oil in certain regions of the reservoir during the production. Furthermore, it can be important to account for changes in the oil and gas compositions since they determine the flow properties of the reservoir fluid, e.g. the viscosity (resistance to movement) and the relative permeability. Finally, it is not uncommon for a reservoir to be perforated by multiple injection and production wells. The interaction between these wells can be complex and may have a significant impact on the profitability of a given production strategy.

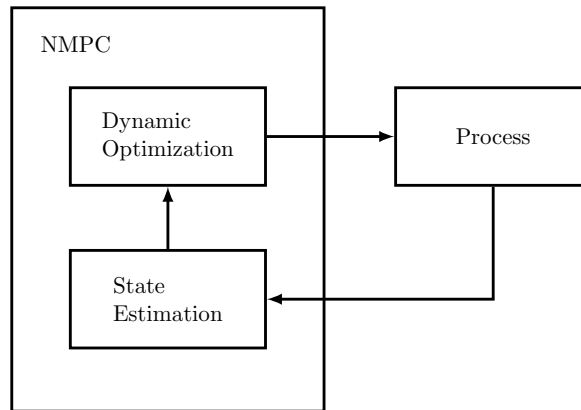


Fig. 1.2: An illustration of the interaction between state estimation, dynamic optimization, and the process in NMPC.

One important measure of the profitability of a production strategy is the recovery factor which is the percentage of the initial oil in the reservoir that is recovered. According to Jansen [77], the recovery factor is often between 10% and 50% after the secondary phase of recovery.

To conclude, it can be nontrivial and time-consuming to manually devise effective production strategies (e.g. strategies with high recovery factors), and there is a significant potential for improving the amount of recovered oil. The purpose of the NMPC algorithms that we consider in this work is to improve the effectiveness of the production strategies by using model-based optimization and data assimilation techniques (in which case the task of production planning will be, at least partially, automated).

Nonlinear model predictive control. As mentioned previously, the subject of this thesis is NMPC for oil reservoirs management. NMPC algorithms compute closed-loop feedback control strategies by using the moving horizon optimization principle, i.e. by repeatedly solving dynamic optimization problems for open-loop control strategies. Only the first part of these open-loop control strategies is implemented in practice before a new strategy is computed based on new process measurements. The objective of NMPC is to optimize the economics of a process (or to follow setpoints) while satisfying constraints, e.g. due to equipment limitations, safety concerns, or requirements related to supply and demand. Dynamic optimization problems are optimization problems that involve dynamical constraints, i.e. difference or differential equations. These dynamical constraints represent a model of the process, and they describe the temporal evolution of a set of state variables. The dynamical equations require an initial condition (for the state variables) in order to have a unique solution. Often, it is necessary to use state estimation (i.e. data assimilation) techniques to determine this initial condition because direct measurement can be 1) difficult or impossible, 2) insufficiently accurate, 3) prohibitively time-consuming, or 4) too expensive. The purpose of state estimation is to reconstruct the state variables of a process based on 1) noisy measurements and 2) a dynamical model of the process. Fig. 1.2 illustrates the interaction between state estimation, dynamic optimization, and the process in NMPC.

Oil reservoir management based on NMPC technology is an example of closed-loop reservoir management (CLRM) which has been proposed to improve the long-term economics of oil recovery processes [24, 75, 79, 80]. CLRM aims at increasing the oil recovery by combining data assimilation (e.g. history matching) with model-based optimization (also referred to as production optimization). Fig. 1.3 illustrates the concept of CLRM. In the red loop (data assimilation), iterative methods are used to update one or more models of the reservoir based on measurements. This corresponds to the state estimation in NMPC. In the blue loop (model-based optimization), iterative optimization algorithms are used to compute a long-term field-wide production strategy that optimizes a financial

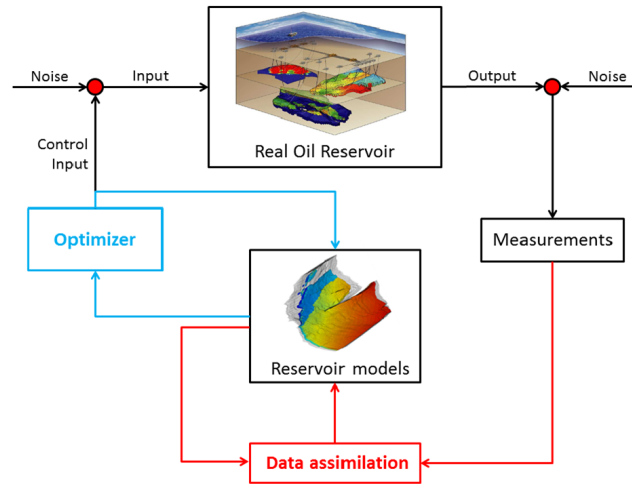


Fig. 1.3: An illustration of CLRM. This figure originally appeared in [30].

measure, e.g. total recovery or net present value. This corresponds to the dynamic optimization in NMPC. The noise on the input and the output from the reservoir represents the finite precision of the production equipment and the measurement devices (also referred to as sensors). In NMPC of oil recovery processes, the state estimation techniques are used to estimate both the current state of the reservoir as well as uncertain model parameters, e.g. the porosity and permeability.

In practice, the role of NMPC-based software for oil reservoirs management will be to assist reservoir engineers in planning the production of oil from the reservoirs. Currently, reservoir engineers manually plan the production using simulation software (based on reservoir flow models) to predict the amount of oil produced by a given production strategy. The NMPC-based software will, in part, automate this task by automatically incorporating recent data and computing an optimized production strategy (also using simulation software based on reservoir flow models). The reservoir engineer may adjust the optimized strategy based on experience and on knowledge about the reservoir that is not incorporated into the models.

Model predictive control (MPC) technology has been widely used in the downstream process industries [58, 97, 133–135] for controlling complex systems, e.g. reactors and distillation columns. Therefore, it is reasonable to believe that NMPC technology can be used for improving the economics of oil recovery processes as well.

Models of oil recovery processes. The predictive model (i.e. the reservoir flow model) in NMPC for oil reservoirs management is central to the effectiveness of the computed optimized production strategies. There exist several types of models of subsurface reservoir flow, and the four most common types are

- a) thermal and compositional models
- b) isothermal and compositional models
- c) black oil models
- d) two-phase flow models

Thermal and compositional models involve mass conservation equations for each of a number of chemical components as well as a combined energy conservation equation for the internal energy of the rock and the fluid.¹ Isothermal and compositional models also involve a mass conservation equation for each chemical component. However, the temperature is assumed to be constant

¹It is possible to formulate separate energy conservation equations for the fluid and the rock (referred to as local thermal nonequilibrium or LTNE models). However, that involves modeling of the energy transfer between the fluid and the rock which is difficult in the context of subsurface reservoir flow.

throughout the reservoir in isothermal models, and they do not involve an energy conservation equation. Both thermal and isothermal compositional models involve phase equilibrium between the fluid phases and the rock. The phase equilibrium conditions determine the temperature (in the case of thermal models), pressure, and phase compositions based on the conserved properties. Black oil models are special cases of isothermal and compositional models in which there are only three components: a water component, a pseudo-oil component, and a pseudo-gas component. The pseudo-gas component can dissolve into the oil phase, but the pseudo-oil component cannot vaporize into the gas phase.² Two-phase flow models are also special cases of isothermal and compositional models, and they involve a water component and a pseudo-oil component. The water and the oil phase are often assumed to be immiscible in such models.

The choice of model depends on the specific reservoir that is being modeled and the specific recovery method that is used. Compositional models are suitable for modeling miscible flooding (i.e. injection of miscible gases) which is one of the most commonly used enhanced oil recovery processes [121, 122]. Furthermore, the Development Geology Reference Manual [2, 113] provides the following discussion of the use of black oil models and compositional models. Black oil models are widely used in industry because they can be used to describe the primary stage of recovery, waterflooding, and certain types of gas injection for many oil and gas reservoirs. However, if the reservoir contains volatile oil or gas condensates, it is in general necessary to use compositional models. Thermal models are most often used to model thermal enhanced oil recovery processes such as steam injection. It is possible to formulate thermal black oil models. However, it is often necessary to use thermal and compositional models to describe thermal oil recovery processes.

There already exists a large body of research on NMPC and CLRM of two-phase flow models. Furthermore, black oil models (as well as two-phase flow models) are special cases of compositional models as mentioned above. Therefore, we are concerned with thermal and isothermal compositional models in this work.³

Phase equilibrium. A key element of the thermal and isothermal compositional models is the phase equilibrium conditions. In the numerical simulation of these models, the reservoir is discretized into a set of grid cells, and the phase equilibrium between the fluid phases and the rock is enforced in each grid cell.

In the reservoir simulation and optimization literature, the phase equilibrium conditions in thermal and isothermal compositional models are often formulated as the isofugacity condition.⁴ That condition is derived from the PT flash which is the phase equilibrium problem that is relevant to isothermal and isobaric (constant pressure) systems. It is called the PT or (PTn) flash because it involves constraints on the temperature, T , pressure, P , and total mixture composition, n .

Although it is common to use the isofugacity condition, the phase equilibrium problem that is relevant to thermal and compositional models is the UV (or UVn) flash. Its name indicates that it involves constraints on the internal energy, U , the volume, V , and the total mixture composition, n . It is also called the isoenergetic-isochoric (constant energy and volume) flash. The total composition and the internal energy are given by the mass and energy conservation equations, and the volume is the size of the grid cell. The solution to the UV flash is the equilibrium temperature, pressure, and phase compositions that maximize the entropy subject to the constraints on the internal energy, volume, and total composition. Therefore, it is a direct statement of the second law of thermodynamics (i.e. the entropy of a closed system is maximal when it is in equilibrium).

The phase equilibrium problem that is relevant to isothermal and compositional models is the

²There are cases where it is relevant to modify the black oil models such that oil can vaporize into the gas phase, e.g. for volatile oil and for gas condensates.

³Note that the extension of this work to black oil models will require modeling of the thermodynamic properties of the pseudo-oil and pseudo-gas components. We do not address this issue.

⁴The fugacity is a thermodynamic function which is defined in terms of the derivative of the Gibbs energy of a given phase with respect to the mole number of a given chemical component.

VT (or VTn) flash. It involves constraints on the volume, V , temperature, T , and total mixture composition, n , and it is also called the isochoric-isothermal (i.e. constant volume and temperature) flash. The condition of maximal entropy does not apply directly to isothermal systems because they are not closed. However, the second law of thermodynamics is valid for the combined system consisting of the isothermal system and its surroundings. The condition of maximal entropy for the combined system is equivalent to a condition of minimal Helmholtz energy for the isothermal system alone. Therefore, the solution to the VT flash is the equilibrium pressure and phase compositions that minimize Helmholtz energy subject to the constraints on the volume, temperature, and total composition. As for the UV flash, the volume is the size of the grid cell, and the total composition is given by the mass conservation equations.

A similar adaptation of the second law of thermodynamics to isothermal and isobaric systems results in a condition of minimal Gibbs energy. This is the condition that is used in the PT flash, and its solution is the equilibrium phase compositions which minimize Gibbs energy at specified temperature, pressure, and total composition.

In this work, we use the UV and the VT flash in the thermal and isothermal compositional reservoir flow models, respectively, because they are directly related to the second law of thermodynamics. Furthermore, because both the UV and the VT flash involve an extremum condition (maximal entropy and minimal Helmholtz energy), we formulate them as equality constrained optimization problems and the phase equilibrium conditions as the corresponding first order optimality conditions. Therefore, the reservoir flow model equations are a set of differential-algebraic equations (DAEs). The differential equations are the conservation equations, and the phase equilibrium conditions are the algebraic equations.

1.2 Literature review

In this section, we present a literature review that provides the context and background for the content and objectives of this thesis. In particular, we discuss 1) algorithms for state estimation and history matching, 2) dynamic optimization algorithms and production optimization, and 3) NMPC for oil recovery processes and CLRM. Furthermore, we discuss 4) models that involve the UV and the VT flash, and 5) currently available open-source thermodynamic software. Finally, we provide 6) references to supplementary literature that may be useful to the reader.

State estimation and history matching. The Kalman filter is an optimal state estimation algorithm for linear systems [89]. However, oil recovery processes, and other phase equilibrium processes, are inherently nonlinear. There exist several state estimation algorithms (filters) for nonlinear systems, e.g. the extended Kalman filter (EKF), the unscented Kalman filter (UKF), and particle filters (PFs) [160]. In the EKF, the original Kalman filter is applied to a linearization of the nonlinear model equations. Consequently, the accuracy of the EKF can be insufficient for systems with significant nonlinearities. The UKF represents the distribution of the state variables using a set of deterministic particles [85].⁵ These particles are chosen such that the approximation of the covariance matrix is accurate up to third order.⁶ However, the UKF can also suffer from limited accuracy for severely nonlinear systems. PFs represent the distribution using a set of random particles [169]. Therefore, they can be more accurate than the EKF and the UKF. A particular PF, the ensemble Kalman filter (EnKF), has gained much attention in the fields of oceanography and oil reservoir characterization because it is computationally tractable for very large-scale systems [54–56, 63]. Originally, research on state estimation (including the above methods) has been focused on stochastic difference and differential equations. However, authors have recently developed the

⁵The EKF represents the distribution of the states by the mean and covariance.

⁶The approximation of the covariance matrix in the EKF is accurate up to first order.

EKF [14, 83, 112], the UKF [106, 107, 119, 131, 132], PFs [69, 70], and the EnKF [128] for stochastic DAEs.

An alternative to the above state estimation algorithms is moving horizon estimation [3] in which the state estimation problem is formulated as a dynamic optimization problem [20]. Consequently, the state estimation problem can be solved using the dynamic optimization algorithms that we discuss later in this literature review.

History matching is a data assimilation technique which aims at estimating the parameters (e.g. petrophysical parameters) in a reservoir model such as to minimize the distance between predicted and observed (i.e. measured) data. History matching in reservoir engineering typically involves 1) the numerical solution of a dynamic optimization problem [38, 100, 189] or 2) the EnKF [1, 71, 88, 187]. It is outside the scope of this review to discuss all relevant work on history matching. Instead, we refer to the review by Udy et al. [170] and the benchmark study by Peters et al. [120] which involves participants from both industry and academia.

In this work, we consider the EKF, UKF, PF, and EnKF for stochastic DAEs, although it is also relevant to consider optimization-based approaches. For large-scale processes, we expect that the EnKF is the most computationally tractable. However, it is outside the scope of this work to test the state estimation algorithms using the thermal and isothermal reservoir flow models. Instead, we test the algorithms using a representative small-scale flash separation process.

Dynamic optimization and production optimization. There exist several methods for solving dynamic optimization problems. The single-shooting, multiple-shooting [22, 157], and simultaneous collocation [18] methods have proven to be the most successful for large-scale dynamic optimization problems relevant to real-world applications [20]. They are collectively known as direct methods because they transform the infinite-dimensional dynamic optimization problem into a finite-dimensional nonlinear program (NLP). Single- and multiple-shooting methods involve the numerical solution of initial value problems (IVPs), i.e. they involve numerical simulations. In simultaneous collocation methods, the dynamical model equations are discretized and incorporated directly as constraints in the resulting NLP. Efficient optimization algorithms for solving NLPs require the gradient of the objective function and the Jacobian of the constraints [115]. Furthermore, it can improve the computational efficiency to use the Hessians of the objective function and the constraint functions. However, the necessary Hessian information can also be approximated efficiently, e.g. with the BFGS method.⁷ For simultaneous collocation, the gradient, the Jacobian, and the Hessians can be derived by straightforward differentiation. For single- and multiple-shooting methods, the objective and constraint functions depend on the numerical solution of one or more IVPs. In such cases, both adjoint methods [82] and forward methods [95, 96] can be used to compute the relevant gradients and Jacobians. There exist both discrete and continuous versions of adjoint methods [29, 33, 94] and forward methods. Recent research shows that continuous adjoint methods can be more efficient than discrete adjoint methods [32, 33].

Research on production optimization often involves simple models, e.g. immiscible or partially miscible two-phase flow models [161] or polymer flooding [98, 99, 191]. Data driven models, e.g. neural networks, are also used [152]. Kourounis et al. [93] solve production optimization problems that involve an isothermal and compositional model. Furthermore, Zaydullin et al. [190] and Garipov et al. [61] consider simulation of thermal and compositional models. The above models do not involve the UV or the VT flash. Polívka and Mikyška [125] consider simulation of an isothermal and compositional model that does involve the VT flash. However, production optimization of thermodynamically rigorous models of oil recovery processes based on the UV and VT flash has not been considered before.

It is common to use single-shooting methods together with an adjoint method to solve production

⁷The BFGS method is named after Charles George Broyden, Roger Fletcher, Donald Goldfarb, and David Shanno who developed it.

optimization problems [25, 33, 59]. Multiple-shooting [31, 45] and simultaneous collocation [72] methods have also been used. As an alternative to the adjoint method, authors also use the simultaneous perturbation stochastic approximation (SPSA) method [192] and an ensemble-based method called EnOpt [39–41, 76]. Do and Reynolds [48] discuss the connection between the SPSA method and the EnOpt method. The adjoint and forward methods compute the gradient exactly whereas the SPSA and EnOpt algorithms approximate the gradient. In SPSA algorithms, the gradient is approximated with a single random perturbation of all decision variables simultaneously, in contrast to finite difference methods that involve a deterministic perturbation of each decision variable individually. Similarly, the EnOpt algorithm uses an ensemble of perturbations to approximate the gradient. Alternatively, (gradient-free) metaheuristic methods can be used, e.g. the particle swarm optimization algorithm [116].

In this work, we use a single-shooting method together with a discrete adjoint method for the numerical solution of dynamic optimization problems that involve DAEs in a specific semi-explicit form (the DAEs represent the reservoir flow models). We implement and test the algorithm using both a representative small-scale flash separation process and the thermal and the isothermal compositional reservoir flow models.

NMPC of oil recovery processes and CLRM. As for production optimization, most research on CLRM and NMPC of oil recovery processes involves two-phase flow models [62, 65, 78, 80, 177]. Neural network models are also used [66, 165]. The EnKF is often used for the history matching [43, 114, 174], and the single-shooting method combined with the adjoint method is often used to solve the production optimization problem [7–9, 114, 124, 153, 155, 156, 174, 175, 184]. However, both the SPSA method [184, 192] and EnOpt [41, 158, 159] are used as well.

Existing reservoir simulation software is frequently used in CLRM, e.g. the open-source Matlab Reservoir Simulation Toolbox (MRST) [7–9, 124], Stanford’s General Purpose Research Simulator (GPRS) [154–156], and Schlumberger’s ECLIPSE [109]. Purkayastha et al. [129, 130] use the CMG STARS software for linear MPC of steam-assisted gravity drainage (SAGD) processes, and Guevara et al. [67] apply NMPC to SAGD processes.⁸ MPC can also be used for setpoint tracking based on setpoints that are computed using production optimization [175]. Finally, Foss and Jensen [60] analyze certain aspects of the control performance of CLRM, and Hou et al. [75] present a review of CLRM.

In this work, we formulate an NMPC algorithm based on either the EKF, UKF, PF, or EnKF as well as a gradient-based dynamic optimization algorithm which uses the single-shooting method and the discrete adjoint method (as mentioned previously). As for the state estimation algorithms, we do not implement and test the NMPC algorithm using the thermal and isothermal compositional reservoir flow models. Instead, we use a representative small-scale flash separation process to test the algorithm.

The UV flash and the VT flash. The UV flash has not been used in models of oil recovery processes before, and only Polívka and Mikyška [125] have used the VT flash (not formulated as an optimization problem) in an isothermal and compositional reservoir flow model. However, the UV flash has been used in models of fluid vessels and flash drums [5, 35, 102], distillation columns [57], and fluid dynamical problems [136]. The VT flash has been used in models of CO₂-H₂O systems⁹ [81], gravitational systems¹⁰ [52], centrifugation of natural gas [36], adsorption processes [27, 151], diffusive phenomena [92], and permeation in flexible pipes [16]. Algorithms for robust simulation of UV flash processes have been developed previously [34, 149, 188]. However, state estimation, dynamic optimization, and NMPC of UV and VT flash processes, which we consider in this work, have not been reported in the open literature.

⁸SAGD is an enhanced oil recovery method for the production of heavy oil, e.g. from tar sands.

⁹Such systems are relevant to CO₂ injection in 1) oil recovery processes and 2) CO₂ sequestration using subsurface reservoirs.

¹⁰For instance, oil reservoirs.

Thermodynamic software. Thermodynamic models are central to phase equilibrium computations. A few open-source thermodynamic software libraries have been reported in the open literature, e.g. the C++ software CoolProp by Bell et al. [15], the Matlab software by Martín et al. [108], and the thermodynamic module in the Open-Calphad software by Sundman et al. [164]. Furthermore, there exist open-source libraries that are not reported in the literature, e.g. the Simulink toolbox T-MATS by Chapman et al. [37] and the thermodynamic library in DWSIM [50].¹¹

However, none of the above software libraries provide both first and second order derivatives of the thermodynamic properties. Such derivatives are necessary in efficient gradient-based algorithms for simulation, state estimation, dynamic optimization, and NMPC of UV and VT flash processes (and of other phase equilibrium processes).

Therefore, in this work, we develop an open-source thermodynamic software library (called ThermoLib) which provides both the first and second order derivatives of the thermodynamic properties.

Supplementary literature. The books by Chen [42] and by Chen et al. [44] describe models and algorithms for numerical simulation of reservoir flow. For more information about thermodynamics, we refer to the book on classical thermodynamics by Callen [28] and the books on chemical engineering thermodynamics by Smith et al. [162], Gmehling et al. [64], and Koretsky [91]. For more on phase equilibrium processes, we refer to the books by Walas [181] and Michelsen and Mollerup [111]. The book by Holman [74] describes models of heat transfer which are relevant to models of thermal oil recovery processes. As mentioned previously, phase equilibrium problems can be formulated as optimization problems [110]. Consequently, dynamic optimization problems that involve phase equilibrium processes belong to the class of bilevel optimization problems [46] and to the closely related class of mathematical programs with equilibrium constraints [104, 118]. The numerical solution of optimization problems is central to the direct dynamic optimization methods (single-shooting, multiple-shooting, and simultaneous collocation). For more information about numerical optimization, we refer to the book by Nocedal and Wright [115] and the books on 1) gradient-based methods and process optimization by Biegler [19], 2) metaheuristic methods by Burke and Kendall [26], and 3) stochastic optimization algorithms by Bhatnagar et al. [17]. For an introduction and overview of state estimation algorithms, we refer to the book by Simon [160]. The book on neural network-based state estimation by Talebi et al. [166] may also be relevant. Finally, we refer to the book on predictive control by Maciejowski [105] and the book on MPC by Rawlings et al. [138].

1.3 Main objectives and contributions

The main objectives of this work are to 1) formulate thermodynamically rigorous thermal and isothermal compositional reservoir flow models using the UV and the VT flash (formulated as optimization problems) and 2) develop algorithms for NMPC of thermal and isothermal oil recovery processes using these models. First, we provide an overview of the objectives (and thereby also the content) of this summary report, and next, we discuss the main contributions of the work presented in this thesis. The objectives of this summary report are to

- a) describe a thermodynamic model based on
 - data and correlations from the DIPPR database
 - cubic equations of state
- b) describe thermodynamically rigorous models of

¹¹The GitHub repository AwesomeThermodynamics [6] contains a list of open-source thermodynamic software as well as other resources on thermodynamics.

- a flash separation process (using the UV flash)
 - a thermal and compositional waterflooding process (also using the UV flash)
 - an isothermal and compositional waterflooding process (using the VT flash)
- c) demonstrate that the models in item b) are in a specific semi-explicit DAE form
- d) describe state estimation algorithms for continuous-discrete stochastic semi-explicit DAEs:
- the EKF
 - the UKF
 - a PF
 - the EnKF
- e) describe an algorithm for gradient-based dynamic optimization of semi-explicit DAEs using
- the single-shooting method
 - the discrete adjoint method (for gradient computations)
 - off-the-shelf numerical optimization software (i.e. NLP solvers)
- f) describe an algorithm for NMPC of semi-explicit DAEs using either of the state estimation algorithms in item d) and the dynamic optimization algorithm in item e)
- g) test the algorithms for state estimation, dynamic optimization, and NMPC using Matlab and/or C implementations and the flash separation process
- h) test the dynamic optimization algorithm using a C/C++ implementation and the thermal and isothermal compositional reservoir flow models

Furthermore, in this summary report, we discuss the computational performance of the implementations in item g) and h) for a number of numerical examples. In particular, we present a performance study of the implementation of the dynamic optimization algorithm in item g). In this performance study, we compare 1) different approaches for the numerical solution of the involved semi-explicit DAEs (in the dynamic optimization algorithm), 2) Matlab, C, and mixed implementations, 3) different off-the-shelf optimization software, and 4) different compilers and linear algebra software. The results of this performance study are relevant to the state estimation algorithms as well because they also involve the numerical solution of semi-explicit DAEs. The results are also relevant to the NMPC algorithm because it is based on the state estimation algorithms and the dynamic optimization algorithm. In item g), we demonstrate (using a numerical example) that the state estimation algorithms can be used for soft sensing of the phase compositions in the flash separation process. Furthermore, we consider a numerical example of economical NMPC (using the EKF in the NMPC algorithm) for disturbance rejection in the flash separation process.

Next, we describe the main contributions of this work, and we provide references to the publications (which are part of this thesis) in which the contributions are described. We also present some of the key equations involved in this work (i.e. the semi-explicit DAE form, the state estimation problem, and the dynamic optimization problem) in order to give a brief overview of the types of models and problems that we consider. Finally, we mention that although several of the main contributions are related to UV flash processes, these contributions are also relevant for other dynamical flash processes (e.g. VT flash processes) because they can be modeled using DAEs in the same semi-explicit form. The main contributions of this thesis are as follows.

ThermoLib. We develop an open-source thermodynamic software library called ThermoLib [142, 143]. The thermodynamic model in ThermoLib uses 1) correlations and data from the DIPPR database and 2) cubic equations of state. ThermoLib provides Matlab and C routines for evaluating the enthalpy, entropy, and volume of ideal gas mixtures, ideal liquid mixtures, and nonideal mixtures (both vapor and liquid). The main novelty of ThermoLib is that its routines also evaluate the first and second order derivatives of the thermodynamic functions. We derive the expressions for these derivatives analytically.

Thermodynamically rigorous thermal and isothermal compositional reservoir flow models. We present novel and thermodynamically rigorous thermal and isothermal compositional reservoir flow models using the UV and the VT flash formulated as equality constrained optimization problems [145, 147, 148]. Thermal and compositional reservoir flow models based on the UV flash have not been considered before, and the VT flash has not previously been formulated as an optimization problem in isothermal and compositional reservoir flow models.

Semi-explicit DAE form. We demonstrate that it is natural to formulate the thermal and isothermal compositional reservoir flow models as DAEs in the semi-explicit form [145, 147, 148]

$$G(x(t), y(t), z(t)) = 0, \quad (1.1a)$$

$$\dot{x}(t) = F(y(t), u(t), d(t)). \quad (1.1b)$$

$x(t)$ is a vector of state variables, $y(t)$ is a vector of algebraic variables, and $z(t)$ is a vector of adjoint algebraic variables. $u(t)$ are manipulated inputs, and $d(t)$ are disturbance variables. The algebraic equations (1.1a) represent phase equilibrium conditions, the differential equations (1.1b) represent conservation equations, and $x(t)$ are the conserved quantities. In the semi-explicit DAE form (1.1), the right-hand side of the conservation equations is independent of the conserved quantities, i.e. $x(t)$ does not appear in the right-hand side of (1.1b).

We formulate the phase equilibrium problems as equality constrained optimization problems in the form

$$\min_y f(y), \quad (1.2a)$$

$$\text{subject to } g(y) = x, \quad (1.2b)$$

$$h(y) = 0. \quad (1.2c)$$

The first order optimality conditions (also called Karush-Kuhn-Tucker or KKT conditions) of the optimization problem (1.2) are the phase equilibrium conditions which are represented by the algebraic equations (1.1a). We describe the first order optimality conditions of (1.2) in Appendix A.

State estimation of UV flash processes. We consider state estimation of UV flash processes [144, 146]. We use the continuous-discrete EKF, UKF, PF, and EnKF, and we consider stochastic semi-explicit DAEs in the form

$$G(\mathbf{x}(t), \mathbf{y}(t), \mathbf{z}(t)) = 0, \quad (1.3a)$$

$$d\mathbf{x}(t) = F(\mathbf{y}(t), u(t), d(t))dt + \sigma(\mathbf{y}(t), u(t), d(t))d\boldsymbol{\omega}(t). \quad (1.3b)$$

The states, $\mathbf{x}(t)$, the algebraic variables, $\mathbf{y}(t)$, and the adjoint algebraic variables, $\mathbf{z}(t)$, are stochastic variables. The second term in (1.3b) represents the uncertainty in the process, e.g. due to 1) uncertainty in the process inputs, 2) unmodeled dynamics, or 3) limited accuracy of model parameters. $\boldsymbol{\omega}(t)$ is a standard Wiener process. The measurements, $\mathbf{y}^m(t_k)$, of the process outputs, $\mathbf{z}^m(t_k)$, are obtained at discrete times, t_k :

$$\mathbf{z}^m(t_k) = H(\mathbf{y}(t_k)), \quad (1.4a)$$

$$\mathbf{y}^m(t_k) = \mathbf{z}^m(t_k) + \mathbf{v}(t_k). \quad (1.4b)$$

(1.4a) is a model of the sensors, and $\mathbf{v}(t_k)$ represents the measurement noise.

Dynamic optimization of UV flash processes. We consider dynamic optimization of UV flash processes using a novel gradient-based dynamic optimization algorithm based on the single-shooting method and the discrete adjoint method [139–141]. We formulate the algorithm for dynamic optimization problems in the form

$$\min_{[x(t); y(t); z(t)]_{t_0}^{t_f}, \{u_k\}_{k=0}^{N-1}} \phi = \int_{t_0}^{t_f} \Phi(y(t), u(t), d(t))dt, \quad (1.5a)$$

subject to

$$x(t_0) = \hat{x}_0, \tag{1.5b}$$

$$G(x(t), y(t), z(t)) = 0, \quad t \in [t_0, t_f], \tag{1.5c}$$

$$\dot{x}(t) = F(y(t), u(t), d(t)), \quad t \in [t_0, t_f], \tag{1.5d}$$

$$u(t) = u_k, \quad t \in [t_k, t_{k+1}[, \quad k = 0, \dots, N-1, \tag{1.5e}$$

$$d(t) = \hat{d}_k, \quad t \in [t_k, t_{k+1}[, \quad k = 0, \dots, N-1, \tag{1.5f}$$

$$\{u_k\}_{k=0}^{N-1} \in \mathcal{U}. \tag{1.5g}$$

(1.5b) is an initial condition, and (1.5c)-(1.5d) are the semi-explicit DAEs (1.1) used to model the UV flash processes. (1.5e) and (1.5f) are zero-order-hold (ZOH) parametrizations of the manipulated inputs and the disturbance variables. Finally, (1.5g) represents constraints on the manipulated inputs. t_0 and $t_N = t_f$ are the initial and final time, and N is the number of control intervals. $[x(t); y(t); z(t)]_{t_0}^{t_f}$ is a vector of dependent decision variables, and $\{u_k\}_{k=0}^{N-1}$ are independent decision variables. The initial estimate of the states, \hat{x}_0 , and the predicted disturbance variables, $\{\hat{d}_k\}_{k=0}^{N-1}$, are parameters in the optimization problem.

NMPC of UV flash processes. We formulate an algorithm for NMPC of UV flash processes using the state estimation algorithms and the gradient-based dynamic optimization algorithm mentioned above. For each new set of measurements, the state estimation algorithm is used to estimate the current states. Next, the dynamic optimization algorithm is used to solve a dynamic optimization problem in the form (1.5) where the current state estimate is the initial condition in (1.5b). The NMPC algorithm is described in the paper in Appendix N which is in preparation for submission.

Dynamic optimization of thermal and isothermal waterflooding processes. We use the gradient-based dynamic optimization algorithm and the thermodynamically rigorous thermal and isothermal compositional reservoir flow models to compute optimized waterflooding strategies [145, 147, 148]. Dynamic optimization of thermal and isothermal compositional reservoir flow models based on the UV and the VT flash has not been considered before.

1.4 Outline of the thesis

Part I Models (Chapter 2-4). In this part, we formulate the models of the flash separation process and the thermal and isothermal waterflooding processes. In Chapter 2, we describe the thermodynamic model (implemented in ThermoLib) that we use to evaluate the thermodynamic functions in these models. We present the model of the flash separation process in Chapter 3, and we present the thermal and isothermal compositional reservoir flow models in Chapter 4.

Part II Algorithms (Chapter 5-8). We describe numerical algorithms for simulation, state estimation, dynamic optimization, and NMPC of semi-explicit DAEs in this part. In Chapter 5, we describe two approaches for the numerical simulation of (deterministic) semi-explicit DAEs based on Euler's implicit method. Furthermore, we present an approach for numerical simulation of stochastic semi-explicit DAEs based on a semi-implicit discretization scheme. In Chapter 6, we describe the continuous-discrete EKF, UKF, PF, and EnKF, and we present the gradient-based dynamic optimization algorithm in Chapter 7. Finally, we discuss the NMPC algorithm in Chapter 8.

Part III Examples (Chapter 9-10). In this part, we present numerical examples which involve the models described in Part I and the algorithms described in Part II. In Chapter 9, we present numerical examples of state estimation, dynamic optimization, and NMPC of the flash separation process. Furthermore, we discuss the computational performance of the Matlab and/or C implementations of the algorithms for these examples, and we present a performance study of

the dynamic optimization algorithm. In Chapter 10, we present numerical examples of dynamic optimization of the thermal and isothermal compositional waterflooding processes, and we discuss the computational efficiency of the C/C++ implementation of the dynamic optimization algorithm for these examples.

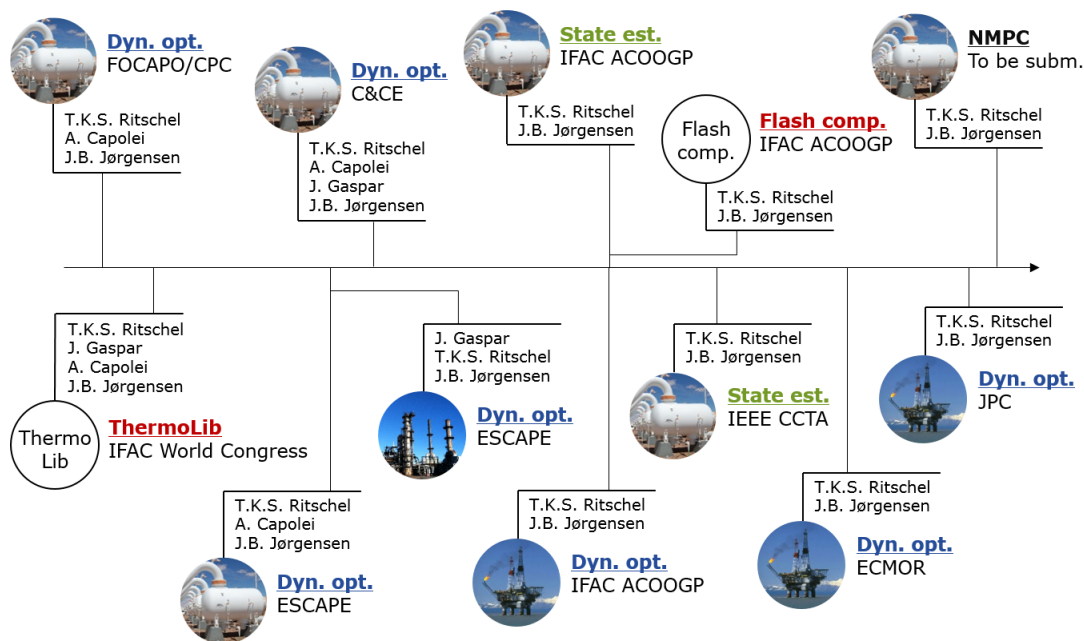
Part IV Conclusions (Chapter 11-12). We present conclusions in Chapter 11, and we provide a number of suggestions for future work in Chapter 12.

Part V Appendix (Appendix A-P). This part contains 1) appendices and 2) the twelve research papers and the two technical reports written during this Ph.D. project. In Appendix A, we describe the first order optimality conditions of the optimization problem (1.2), and in Appendix B, we describe the workstation that we use for carrying out the numerical examples presented in Part III. In the appendix, the papers and the technical reports are organized in chronological order as shown in Fig. 1.4 (the paper that is published in *Computers and Chemical Engineering*, abbreviated C&CE, is positioned after the date of online publication). Here, we present the papers and the technical reports based on subject.

In the technical report in Appendix O, we describe the thermodynamic model that is implemented in ThermoLib. The technical report also contains analytical expressions for the first and second order derivatives of the thermodynamic functions. The paper in Appendix D presents a performance study of the ThermoLib routines.

The technical report in Appendix P describes 1) the computation of fugacities (and fugacity coefficients) based on the thermodynamic model in ThermoLib, 2) the solution of the PT flash, and 3) the computation of phase envelopes. The paper in Appendix H discusses 1) the formulation of flash problems as optimization problems, 2) the relations between different types of flash problems, and 3) the sensitivity computations that are relevant to efficient gradient-based dynamic simulation and optimization of flash processes.

The papers in Appendix C, E, and G describe the algorithm for gradient-based dynamic optimization of UV flash processes, i.e. of semi-explicit DAEs. We use that algorithm for dynamic optimization of the thermodynamically rigorous models of thermal and isothermal compositional reservoir flow in the papers in Appendix I, L, and M. The papers in Appendix J and K describe the continuous-discrete EKF, UKF, PF, and EnKF for state estimation of UV flash processes, and the paper in Appendix N describes NMPC of UV flash processes (using the EKF in the NMPC algorithm). Finally, the author of this thesis has coauthored the paper in Appendix F on gradient-based dynamic optimization of a distillation column (which is modeled using the UV flash).



Sources





www.technopow.com/2018/03/15/oil-and-gas-separation-equipment-market-demand-and-supply-2018-to-2025 (Accessed: 4th of June 2018)
 
www.wst-me.com/petrochemicals (Accessed: 4th of June 2018)
 
www.wallscover.com/oil-platform.html (Accessed: 4th of June 2018)

Fig. 1.4: An overview of the papers written during this Ph.D. project. In the papers, we present numerical examples of dynamic optimization (Dyn opt.), state estimation (State est.), and NMPC of a flash separation process (leftmost picture in the bottom), a distillation column (middle picture in the bottom), and waterflooding processes (rightmost picture in the bottom). Furthermore, one paper discusses the ThermoLib software library and another paper discusses phase equilibrium computations in reservoir simulation and optimization (Flash comp.). We refer to Appendix C-N for more information about where the papers are published.

Part I

Models

Thermodynamic model

In this chapter, we describe the thermodynamic model that is implemented in ThermoLib. The model consists of a set of expressions for the enthalpy, $H = H(T, P, n)$, entropy, $S = S(T, P, n)$, and volume, $V = V(T, P, n)$, of a mixture as functions of temperature, T , pressure, P , and mixture composition (in moles), n . All other thermodynamic functions can be computed using fundamental thermodynamic relations: the internal energy is $U = H - PV$, the Gibbs energy is $G = H - TS$, and the Helmholtz energy is $A = U - TS$. The thermodynamic functions appear in the model of the flash separation process described in Chapter 3 and in the reservoir flow models described in Chapter 4. The first and second order derivatives of the thermodynamic functions are necessary in the gradient-based algorithms described in Part II. However, we omit the description of these derivatives for the sake of the presentation. Instead, we refer to the technical report in Appendix O where they are described. We briefly describe the thermodynamic model in Section 2.1. In Section 2.2, we describe the DIPPR correlations that are relevant to the thermodynamic properties of ideal gas and ideal liquid mixtures described in Section 2.3 and Section 2.4. In Section 2.5, we present expressions for the thermodynamic properties of nonideal mixtures. Finally, we present a brief summary of this chapter in Section 2.6. The content of this chapter is based on the technical report in Appendix O.

2.1 Brief description

In this chapter, we present expressions for the enthalpy, entropy, and volume of 1) ideal gas mixtures, 2) ideal liquid mixtures, and 3) nonideal mixtures (i.e. both nonideal gas and liquid mixtures). We use the ideal gas law and a correlation for the pure component ideal gas heat capacity (from the DIPPR database) to describe the thermodynamic properties of ideal gas mixtures. In order to describe the thermodynamic properties of ideal liquid mixtures, we use 1) a DIPPR correlation for the pure component vaporization (or saturation) pressure, 2) a DIPPR correlation for the pure component liquid volume, and 3) the expressions for the thermodynamic properties of ideal gas mixtures. Finally, we use the expressions for the thermodynamic properties of ideal gas mixtures together with expressions based on the cubic equations of state to describe the thermodynamic properties of nonideal mixtures.

2.2 DIPPR correlations

In this section, we describe correlations for the ideal gas heat capacity, the vaporization pressure, and the liquid volume of a pure component as functions of temperature, T , in K. The correlations and the parameters come from the DIPPR database. These properties are relevant to the thermodynamic properties of ideal gas and liquid mixtures that are described in Section 2.3 and 2.4.

2.2.1 The ideal gas heat capacity

The DIPPR correlation for the ideal gas heat capacity of component k , $c_{P,k}^{ig} = c_{P,k}^{ig}(T)$, is

$$c_{P,k}^{ig} = A_k + B_k \left(\frac{C_k}{T} \right)^2 + D_k \left(\frac{E_k}{T} \right)^2. \quad (2.1)$$

The parameters in the correlation, A_k , B_k , C_k , D_k , and E_k , are specific to this correlation and to each chemical component. The unit of the heat capacity is J/(kmol · K). We use the integral of the heat capacity in the expressions for the ideal gas enthalpy. We express the integral using the auxiliary function $\Gamma_k = \Gamma_k(T)$:

$$\int_{T_0}^{T_1} c_{P,k}^{ig} dT = \Gamma_k(T_1) - \Gamma_k(T_0), \quad (2.2a)$$

$$\Gamma_k(T) = A_k T + B_k C_k \coth \left(\frac{C_k}{T} \right) - D_k E_k \tanh \left(\frac{E_k}{T} \right). \quad (2.2b)$$

Furthermore, we use the integral of the heat capacity divided by temperature in the expression for the ideal gas entropy. We express this integral using another auxiliary function, $\Pi_k = \Pi_k(T)$:

$$\int_{T_0}^{T_1} \frac{c_{P,k}^{ig}}{T} dT = \Pi_k(T_1) - \Pi_k(T_0), \quad (2.3a)$$

$$\begin{aligned} \Pi_k(T) = & A_k \ln(T) + B_k \left(\frac{C_k}{T} \coth \left(\frac{C_k}{T} \right) - \ln \sinh \left(\frac{C_k}{T} \right) \right) \\ & - D_k \left(\frac{E_k}{T} \tanh \left(\frac{E_k}{T} \right) - \ln \cosh \left(\frac{E_k}{T} \right) \right). \end{aligned} \quad (2.3b)$$

2.2.2 Vaporization pressure

The vaporization pressure (or saturation pressure) of component k , $P_k^{sat} = P_k^{sat}(T)$, is

$$P_k^{sat} = \exp(\ln P_k^{sat}), \quad (2.4a)$$

$$\ln P_k^{sat} = A_k + \frac{B_k}{T} + C_k \ln(T) + D_k T^{E_k}. \quad (2.4b)$$

Again, the correlation parameters, A_k , B_k , C_k , D_k , and E_k , are specific to the above correlation and to each chemical component. The unit of the vaporization pressure is Pa. Furthermore, the expressions for the thermodynamic properties of ideal liquid mixtures involve the first order derivative of the vaporization pressure,

$$\frac{\partial P_k^{sat}}{\partial T} = P_k^{sat} \frac{\partial \ln P_k^{sat}}{\partial T}, \quad (2.5a)$$

$$\frac{\partial \ln P_k^{sat}}{\partial T} = \frac{1}{T} \left(C_k - \frac{B_k}{T} + D_k E_k T^{E_k} \right). \quad (2.5b)$$

2.2.3 Liquid volume

The DIPPR correlation for the molar liquid volume of the k 'th component, $v_k^l = v_k^l(T)$, is

$$v_k^l = \frac{B_k^{1+(1-\frac{T}{C_k})^{D_k}}}{A_k}. \quad (2.6)$$

As with the other two correlations, the parameters, A_k , B_k , C_k , and D_k , are specific to each chemical component and to the above correlation. The unit of the molar liquid volume is m³/kmol.

The expressions for the ideal liquid properties also involve the first order derivative of the molar liquid volume,

$$\frac{\partial v_k^l}{\partial T} = -\ln B_k \frac{D_k}{C_k} \left(1 - \frac{T}{C_k}\right)^{D_k-1} v_k^l. \quad (2.7a)$$

2.3 Ideal gas mixtures

In this section, we present expressions for the enthalpy, entropy, and volume of an ideal gas mixture that consists of N_C chemical components. The composition of the mixture (in moles) is $n = [n_1; \dots; n_{N_C}]$. In Section 2.3.1, we describe the molar enthalpy, entropy, and volume of a pure component ideal gas that consists of the k 'th component. Next, we use those properties to describe the ideal gas mixture properties in Section 2.3.2.

2.3.1 Pure component ideal gas

The DIPPR database provides the molar enthalpy and entropy at a reference temperature, T_0 , and pressure, P_0 , i.e. $h_k^{ig}(T_0, P_0)$ and $s_k^{ig}(T_0, P_0)$. In order to derive expressions for the molar enthalpy and entropy at arbitrary temperature and pressure, we consider the combination of an isobaric (constant pressure) and an isothermal (constant temperature) process. The derivations are based on the following fundamental thermodynamic relations applied to the molar ideal gas properties [111, Table 3]. For the isobaric process,

$$\frac{\partial h_k^{ig}}{\partial T} = c_{P,k}^{ig}, \quad (2.8a)$$

$$\frac{\partial s_k^{ig}}{\partial T} = \frac{c_{P,k}^{ig}}{T}, \quad (2.8b)$$

and for the isothermal process,

$$\frac{\partial h_k^{ig}}{\partial P} = v_k^{ig} - T \frac{\partial v_k^{ig}}{\partial T}, \quad (2.9a)$$

$$\frac{\partial s_k^{ig}}{\partial P} = -\frac{\partial v_k^{ig}}{\partial T}. \quad (2.9b)$$

The molar ideal gas volume of the k 'th component, $v_k^{ig} = v_k^{ig}(T, P)$, is given by the ideal gas law:

$$v_k^{ig} = \frac{RT}{P}. \quad (2.10)$$

R is the gas constant. The partial derivative of the molar ideal gas volume with respect to temperature is

$$\frac{\partial v_k^{ig}}{\partial T} = \frac{R}{P}. \quad (2.11)$$

For the isobaric process, we integrate the thermodynamic relations (2.8) from T_0 to T :

$$h_k^{ig}(T, P_0) = h_k^{ig}(T_0, P_0) + \int_{T_0}^T c_{P,k}^{ig} dT, \quad (2.12a)$$

$$s_k^{ig}(T, P_0) = s_k^{ig}(T_0, P_0) + \int_{T_0}^T \frac{c_{P,k}^{ig}}{T} dT. \quad (2.12b)$$

For the isothermal process, we integrate the relations (2.9) from P_0 to P :

$$h_k^{ig}(T, P) = h_k^{ig}(T, P_0) + \int_{P_0}^P v_k^{ig} - T \frac{\partial v_k^{ig}}{\partial T} dP, \quad (2.13a)$$

$$s_k^{ig}(T, P) = s_k^{ig}(T, P_0) - \int_{P_0}^P \frac{\partial v_k^{ig}}{\partial T} dP. \quad (2.13b)$$

The integrand in (2.13a) is zero because v_k^{ig} is linear in the temperature. Therefore, the molar ideal gas enthalpy is independent of pressure, i.e. $h_k^{ig}(T, P) = h_k^{ig}(T, P_0) = h_k^{ig}(T)$. We use (2.2a) and (2.3a) to obtain the final expressions for the molar ideal gas enthalpy, $h_k^{ig} = h_k^{ig}(T)$, and entropy, $s_k^{ig} = s_k^{ig}(T, P)$:

$$h_k^{ig} = h_k^{ig}(T_0, P_0) + \Gamma_k(T) - \Gamma_k(T_0), \quad (2.14a)$$

$$s_k^{ig} = s_k^{ig}(T_0, P_0) + \Pi_k(T) - \Pi_k(T_0) - R(\ln(P) - \ln(P_0)). \quad (2.14b)$$

The auxiliary functions, $\Gamma_k(T)$ and $\Pi_k(T)$, are given by (2.2b) and (2.3b).

2.3.2 Ideal gas mixture

The enthalpy, $H^{ig} = H^{ig}(T, n)$, entropy, $S^{ig} = S^{ig}(T, P, n)$, and volume, $V^{ig} = V^{ig}(T, P, n)$, of the ideal gas mixture are

$$H^{ig} = \sum_{k=1}^{N_C} n_k h_k^{ig}, \quad (2.15a)$$

$$S^{ig} = \sum_{k=1}^{N_C} n_k s_k^{ig} - R \sum_{k=1}^{N_C} n_k \ln(y_k), \quad (2.15b)$$

$$V^{ig} = \frac{NRT}{P}, \quad (2.15c)$$

where the vapor mole fraction, y_k , and the total amount of moles, N , are

$$y_k = \frac{n_k}{N}, \quad (2.16a)$$

$$N = \sum_{k=1}^{N_C} n_k. \quad (2.16b)$$

The second term in (2.15b) is the entropy of mixing, and (2.15c) is the ideal gas law.

2.4 Ideal liquid mixtures

In this section, we present expressions for the enthalpy, entropy, and volume of an ideal liquid mixture of N_C components with a composition of $n = [n_1; \dots; n_{N_C}]$. As for the ideal gas properties, we 1) describe the molar enthalpy, entropy, and volume of a pure component ideal liquid consisting of the k 'th component and 2) use those expressions to describe the properties of the ideal liquid mixture. Unlike the ideal gas properties, the ideal liquid properties are not relevant to the thermodynamic properties of nonideal mixtures that we describe in Section 2.5.

2.4.1 Pure component ideal liquid

First, we describe the molar enthalpy, entropy, and volume of vaporization, and use those properties to express the ideal liquid properties at vaporization (or saturation) conditions, i.e. at T and $P_k^{sat} = P_k^{sat}(T)$. Finally, we consider an isothermal process from $P_k^{sat}(T)$ to P as in Section 2.3.1. The volume, $\Delta v_k^{vap} = \Delta v_k^{vap}(T)$, entropy, $\Delta s_k^{vap} = \Delta s_k^{vap}(T)$, and enthalpy, $\Delta h_k^{vap} = \Delta h_k^{vap}(T)$, of vaporization are

$$\Delta v_k^{vap} = \frac{RT}{P_k^{sat}} - v_k^l, \quad (2.17a)$$

$$\Delta s_k^{vap} = \frac{\partial P_k^{sat}}{\partial T} \Delta v_k^{vap}, \quad (2.17b)$$

$$\Delta h_k^{vap} = T \Delta s_k^{vap}. \quad (2.17c)$$

The first term in (2.17a) is the molar ideal gas volume (2.10), and $v_k^l = v_k^l(T)$ is the liquid volume given by the DIPPR correlation (2.6). (2.17b) is the Clausius-Clapeyron relation, and (2.17c) is a consequence of the fact that the Gibbs energy of vaporization is zero, i.e. $\Delta g_k^{vap} = \Delta h_k^{vap} - T\Delta s_k^{vap} = 0$. The molar ideal liquid enthalpy, $h_k^{sat} = h_k^{sat}(T)$, and entropy, $s_k^{sat} = s_k^{sat}(T)$, at saturation conditions are

$$h_k^{sat} = h_k^{ig}(T) - \Delta h_k^{vap}, \quad (2.18a)$$

$$s_k^{sat} = s_k^{ig}(T, P_k^{sat}) - \Delta s_k^{vap}, \quad (2.18b)$$

where the molar ideal gas enthalpy and entropy are given by (2.14a) and (2.14b). As mentioned, we integrate the fundamental relations (2.9), applied to the molar ideal liquid properties, from $P_k^{sat} = P_k^{sat}(T)$ to P in order to obtain expressions for the molar ideal liquid enthalpy, $h_k^{id} = h_k^{id}(T, P)$, and entropy, $s_k^{id} = s_k^{id}(T, P)$:

$$h_k^{id} = h_k^{sat} + \left(v_k^l - T \frac{\partial v_k^l}{\partial T} \right) (P - P_k^{sat}), \quad (2.19a)$$

$$s_k^{id} = s_k^{sat} - \frac{\partial v_k^l}{\partial T} (P - P_k^{sat}). \quad (2.19b)$$

2.4.2 Ideal liquid mixture

The enthalpy, $H^{id} = H^{id}(T, P, n)$, entropy, $S^{id} = S^{id}(T, P, n)$, and volume, $V^{id} = V^{id}(T, n)$, of the ideal liquid mixture are

$$H^{id} = \sum_{k=1}^{N_C} n_k h_k^{id}, \quad (2.20a)$$

$$S^{id} = \sum_{k=1}^{N_C} n_k s_k^{id} - R \sum_{k=1}^{N_C} n_k \ln x_k, \quad (2.20b)$$

$$V^{id} = \sum_{k=1}^{N_C} n_k v_k^l, \quad (2.20c)$$

where the liquid mole fraction, x_k , and the total amount of moles, N , are

$$x_k = \frac{n_k}{N}, \quad (2.21a)$$

$$N = \sum_{k=1}^{N_C} n_k. \quad (2.21b)$$

The second term in (2.20b) is the entropy of mixing.

2.5 Nonideal mixtures

In this section, we describe the enthalpy, entropy, and volume of a nonideal mixture (either vapor or liquid) based on a cubic equation of state, e.g. the Soave-Redlich-Kwong (SRK) or Peng-Robinson (PR) equation of state. The mixture consists of N_C components and the composition (in moles) is $n = [n_1; \dots; n_{N_C}]$. First, we describe the cubic equations of state and the residual properties that are derived from them. Next, we describe the enthalpy, entropy, and volume of the nonideal mixture.

2.5.1 Residual properties

We compute the molar residual enthalpy and entropy, as well as the volume, from a cubic equation of state,

$$P = \frac{RT}{v - b_m} - \frac{a_m}{(v + \epsilon b_m)(v + \sigma b_m)}. \quad (2.22)$$

The scalars ϵ and σ are specific to each equation of state. We omit the description of the parameters $a_m = a_m(T, n)$ and $b_m = b_m(n)$. They are described in the technical report in Appendix O. In practice, the equation of state (2.22) is often solved for the compressibility factor, $Z = Z(T, P, n) = Pv/(RT)$.¹ Consequently, the molar volume of the mixture, $v = v(T, P, n)$, is

$$v = \frac{RTZ}{P}. \quad (2.23)$$

For cubic equations of state, the molar residual enthalpy, $h^R = h^R(T, P, n)$, and entropy, $s^R = s^R(T, P, n)$, are

$$h^R = RT(Z - 1) + \frac{1}{\epsilon - \sigma} \frac{1}{b_m} \left(T \frac{\partial a_m}{\partial T} - a_m \right) \ln \left(\frac{Z + \epsilon B}{Z + \sigma B} \right), \quad (2.24a)$$

$$s^R = R \ln(Z - B) + \frac{1}{\epsilon - \sigma} \frac{1}{b_m} \frac{\partial a_m}{\partial T} \ln \left(\frac{Z + \epsilon B}{Z + \sigma B} \right). \quad (2.24b)$$

We omit the definition of $B = B(T, P, n)$, and we refer to the technical reports in Appendix O and Appendix P for more details on residual properties based on cubic equations of state.

2.5.2 Nonideal mixture

The molar enthalpy, $h = h(T, P, n)$, and entropy, $s = s(T, P, n)$, of a nonideal mixture are

$$h = h^{ig} + h^R, \quad (2.25a)$$

$$s = s^{ig} + s^R, \quad (2.25b)$$

where $h^{ig} = h^{ig}(T, n)$ and $s^{ig} = s^{ig}(T, P, n)$ are the molar enthalpy and entropy of an ideal gas mixture, i.e. $h^{ig} = H^{ig}/N$ and $s^{ig} = S^{ig}/N$ where H^{ig} and S^{ig} are given by (2.15a) and (2.15b). The total amount of moles, N , is

$$N = \sum_{k=1}^{N_C} n_k. \quad (2.26)$$

Finally, the enthalpy, $H = H(T, P, n)$, entropy, $S = S(T, P, n)$, and volume, $V = V(T, P, n)$, of the nonideal mixture are

$$H = Nh, \quad (2.27a)$$

$$S = Ns, \quad (2.27b)$$

$$V = Nv. \quad (2.27c)$$

2.6 Summary

In this chapter, we have presented expressions for the enthalpy, entropy, and volume of 1) ideal gas mixtures, 2) ideal liquid mixtures, and 3) nonideal mixtures (either vapor or liquid). The thermodynamic properties are central to the model equations in Chapter 3 and Chapter 4. Chapter 9 contains a numerical example that involves an ideal gas and liquid mixture. However, almost all of the numerical examples in this work involve nonideal mixtures, and we compute the involved thermodynamic functions using the expressions in Section 2.5, i.e. using a cubic equation of state.

¹The cubic equation of state (2.22) has up to three solutions where one corresponds to a vapor phase, and another corresponds to a liquid phase.

Flash separation

In this chapter, we present a model of a flash separation process. The model is based on 1) a set of mass and energy conservation equations and 2) a set of phase equilibrium conditions derived from the second law of thermodynamics. The flash separation process does not involve spatial dynamics, and it is a small-scale process compared to the reservoir flow models that we present in Chapter 4. However, it is representative of the thermal and compositional reservoir flow model because the involved conservation equations and phase equilibrium problems are conceptually similar. In Chapter 9, we test the algorithms that we present in Part II using the flash separation process presented in this chapter.

We present a brief description of the separation process in Section 3.1. In Section 3.2, we describe the phase equilibrium problem (i.e. the UV flash), and we present the conservation equations in Section 3.3. In Section 3.4, we demonstrate that the model of the flash separation process is in a semi-explicit DAE form. The algorithms in Part II are formulated for systems in this form. Finally, we briefly summarize this chapter in Section 3.5. The model presented in this chapter is based on the models that are presented in the papers in Appendix C, E, G, J, K, and N.

3.1 Brief description

The flash separation process is illustrated in Fig. 3.1. The separator is continuously supplied by a vapor-liquid mixture feed that consists of N_C chemical components. The vapor phase and the liquid phase exit the separator through separate vapor-liquid streams. The separator is subject to either heating or cooling. The vapor phase and the liquid phase in the flash separator are assumed to be in thermal, mechanical, and chemical equilibrium at all times, and we assume that both phases always exist. The total vapor flow rate, F_V , the total liquid flow rate, F_L , and the heat input, Q , are manipulated inputs, i.e. they are used to control the process. In Chapter 9, we present several numerical examples that involve this flash separation process. The examples include 1) estimation of the compositions of the vapor-liquid mixture based on temperature and pressure measurements (an example of soft sensing) and 2) control strategies that minimize energy consumption (in terms of cooling) while satisfying an upper bound on the separator pressure (as well as other constraints).

3.2 Phase equilibrium

As mentioned, the vapor phase (v) and the liquid phase (l) in the separator are in thermal, mechanical, and chemical equilibrium. The thermal equilibrium implies that the temperature of the vapor phase, T^v , and the liquid phase, T^l , are identical, and the mechanical equilibrium implies that the pressure of the vapor phase, P^v , and the liquid phase, P^l , are identical, i.e. $T = T^v = T^l$

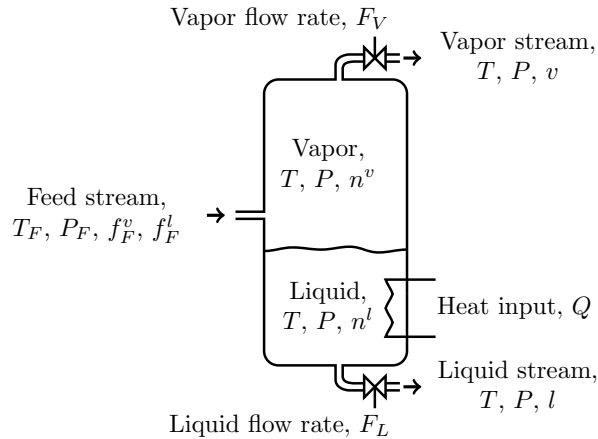


Fig. 3.1: A schematic of the flash separator. This figure originally appeared in [139].

and $P = P^v = P^l$. The chemical equilibrium implies that for each chemical component, there is no net exchange of mass between the two phases. The second law of thermodynamics states that the entropy, S , of a closed system in equilibrium is maximal. Furthermore, the volume, V , of the separator is fixed, and the internal energy, U , and the total amounts of moles of each component, $n = [n_1; \dots; n_{N_C}]$, are given by the conservation equations that we describe in Section 3.3. Consequently, the equilibrium temperature, T , pressure, P , and phase compositions, $n^v = [n_1^v; \dots; n_{N_C}^v]$ and $n^l = [n_1^l; \dots; n_{N_C}^l]$, maximize the entropy while satisfying the constraints on the internal energy, the volume, and the total amounts of moles:

$$\max_{T, P, n^v, n^l} \quad S = S^v(T, P, n^v) + S^l(T, P, n^l), \quad (3.1a)$$

$$\text{subject to} \quad U^v(T, P, n^v) + U^l(T, P, n^l) = U, \quad (3.1b)$$

$$V^v(T, P, n^v) + V^l(T, P, n^l) = V, \quad (3.1c)$$

$$n_k^v + n_k^l = n_k, \quad k = 1, \dots, N_C. \quad (3.1d)$$

(3.1) is referred to as the UV flash optimization problem because U and V are specified parameters in the problem. It is also called the UVn flash (because n is also specified) or the isoenergetic-isochoric (constant energy-constant volume) flash. The phase equilibrium conditions are the corresponding first order optimality conditions (also called Karush-Kuhn-Tucker or KKT conditions). The thermodynamic functions involved in (3.1) are discussed in Chapter 2.

3.3 Mass and energy conservation

The conservation equations describe the temporal evolution of the internal energy, U , and the total amount of moles of the k 'th component, n_k , in the vapor-liquid mixture in the separator:

$$\dot{U}(t) = H_F^v(t) + H_F^l(t) - H_V(t) - H_L(t) + Q(t), \quad (3.2a)$$

$$\dot{n}_k(t) = f_{F,k}^v(t) + f_{F,k}^l(t) - v_k(t) - l_k(t), \quad k = 1, \dots, N_C. \quad (3.2b)$$

In (3.2), we have assumed that the kinetic and potential energies of the feed stream and the vapor-liquid streams are negligible. $H_F^v = H_F^v(T_F, P_F, f_F^v)$ and $H_F^l = H_F^l(T_F, P_F, f_F^l)$ are the vapor-liquid enthalpies of the feed stream. T_F and P_F are the temperature and pressure of the feed stream, and $f_{F,k}^v$ and $f_{F,k}^l$ are the molar vapor and liquid feed flow rates of the k 'th component. $H_V = F_V h^v$ and $H_L = F_L h^l$ are the enthalpies of the vapor-liquid streams where $h^v = h^v(T, P, n^v)$ and $h^l = h^l(T, P, n^l)$ are the molar enthalpies of the vapor and liquid phases of the mixture in the separator. $v_k = y_k F_V$ and $l_k = x_k F_L$ are the molar flow rates of the k 'th component in the vapor

and liquid streams, and y_k and x_k are the vapor and liquid mole fractions of the mixture in the separator. As mentioned previously, F_V and F_L are the total flow rates of the vapor-liquid streams, and Q is the heat input. The total flow rate of the feed stream is $F_F = \sum_{k=1}^{N_C} (f_{F,k}^v + f_{F,k}^l)$.

3.4 Semi-explicit differential-algebraic form

In this section, we demonstrate that the model of the flash separator is in the semi-explicit DAE form (1.1) which we repeat here:

$$G(x(t), y(t), z(t)) = 0, \quad (3.3a)$$

$$\dot{x}(t) = F(y(t), u(t), d(t)). \quad (3.3b)$$

The states, $x(t)$, the algebraic variables, $y(t)$, and the adjoint algebraic variables, $z(t)$, are

$$x = [U; n] \in \mathbb{R}^{1+N_C}, \quad (3.4a)$$

$$y = [T; P; n^v; n^l] \in \mathbb{R}^{2+2N_C}, \quad (3.4b)$$

$$z = [\eta; \mu] \in \mathbb{R}^{2+N_C}, \quad (3.4c)$$

and the manipulated inputs, $u(t)$, and the disturbance variables, $d(t)$, are

$$u = [Q; F_V; F_L] \in \mathbb{R}^3, \quad (3.4d)$$

$$d = [T_F; P_F; f_F^v; f_F^l] \in \mathbb{R}^{2+2N_C}. \quad (3.4e)$$

The algebraic equations (3.3a) represent the phase equilibrium conditions, i.e. the first order optimality conditions of the UV flash optimization problem (3.1). η and μ in (3.4c) are the corresponding Lagrange multipliers. The differential equations (3.3b) represent the conservation equations (3.2). We discuss the phase equilibrium conditions in Section 3.4.1 and the conservation equations in Section 3.4.2.

3.4.1 Phase equilibrium conditions

The UV flash optimization problem (3.1) is in the form

$$\min_y f(y), \quad (3.5a)$$

$$\text{subject to } g(y) = x, \quad (3.5b)$$

$$h(y) = 0. \quad (3.5c)$$

The objective function in (3.5a) is the entropy of the vapor-liquid mixture in the separator with a negative sign, $-S$.¹ The constraint (3.5b) represents the specification of the internal energy, U , and the total amounts of moles, n , and the constraint (3.5c) represents the specification of the volume, V . The first order optimality conditions of the optimization problem (3.5) are derived in Appendix A. They are in the form

$$G(x, y, z) = 0. \quad (3.6)$$

The Lagrange multipliers, $z = [\eta; \mu]$, are associated with the constraints (3.5b) and (3.5c).

¹The sign is negative because (3.1) is a maximization problem and (3.5) is a minimization problem.

3.4.2 Conservation equations

The left-hand side of the conservation equations (3.2) contains the time derivatives of the internal energy, U , and the total amounts of moles, n , i.e. the time derivatives of the state variables, x . The quantities related to the feed stream depend only on the properties of the feed: T_F , P_F , f_F^v , and f_F^l . The quantities related to the vapor-liquid streams depend on 1) the properties of the mixture in the separator, i.e. T , P , n^v , and n^l , and 2) the manipulated inputs F_V and F_L . Finally, the heat input, Q , in the energy conservation equation is a manipulated input. No quantity on the right-hand side depends on the internal energy, U , or the total amounts of moles, n . In conclusion, the right-hand side of the conservation equations (3.2) depends only on the algebraic variables, y , the manipulated inputs, u , and the disturbance variables, d , i.e. the conservation equations are in the form

$$\dot{x}(t) = F(y(t), u(t), d(t)). \quad (3.7)$$

3.5 Summary

In this chapter, we have presented a model of a flash separation process. The model consists of a set of phase equilibrium conditions (algebraic equations) and a set of conservation equations (differential equations). We formulate the phase equilibrium problem (i.e. the UV flash) as an optimization problem based on the second law of thermodynamics, and we have demonstrated that the model is in a semi-explicit DAE form. The algorithms in Part II are formulated for systems in this form. We use the model of the flash separation process in the numerical examples and performance tests in Chapter 9. Furthermore, the conservation equations and the phase equilibrium problem in the model of the flash separation process are conceptually similar to those in the thermal and compositional reservoir flow model that we present in Chapter 4.

Reservoir flow

In this chapter, we present models of thermal and isothermal waterflooding processes. The models are compositional, and they consist of 1) a set of conservation equations (partial differential equations) and 2) phase equilibrium conditions (algebraic equations) derived from the second law of thermodynamics. The thermal and compositional model involves both mass and energy conservation equations. The isothermal and compositional model only involves mass conservation equations, and they are identical to the ones in the thermal and compositional model. The phase equilibrium problem in the thermal model (the UV flash) is conceptually similar to the phase equilibrium problem in the model of the flash separation process presented in Chapter 3. The second law of thermodynamics states that the entropy of a closed system in equilibrium is maximal. Consequently, it does not apply to isothermal systems because they are not closed. However, the combination of an isothermal system and its surroundings is closed. The condition of maximal entropy of the combination of the isothermal system and its surroundings is equivalent to a condition of minimal Helmholtz energy of the isothermal system alone [28]. This is the condition used in the VT flash which is the phase equilibrium problem that is relevant to the isothermal and compositional model. We use a finite volume method to discretize the conservation equations, and we enforce the condition of phase equilibrium in each grid cell in the discretized reservoir. In Chapter 10, we use the dynamic optimization algorithm that we describe in Chapter 7 to optimize thermal and isothermal waterflooding strategies based on the two models that we present in this chapter.

We briefly describe the thermal and isothermal waterflooding processes in Section 4.1. In Section 4.2 and 4.3, we present the thermal and isothermal compositional models and demonstrate that they are both in the semi-explicit DAE form. Finally, we briefly summarize this chapter in Section 4.4. The content of this chapter is based on the papers in Appendix I, L, and M.

4.1 Brief description

Fig. 4.1 illustrates the waterflooding process for a rectangular (discretized) reservoir. We consider an oil reservoir that consists of a porous rock which contains water, oil, and gas as illustrated in Fig. 4.2. The water phase is not miscible with the oil and the gas phases, and it only contains a single water component. We assume that the oil and gas phases contain N_C chemical components that exist in both phases. The three fluid phases are in thermal and mechanical equilibrium with each other and with the reservoir rock. Furthermore, the oil and the gas phases are in chemical equilibrium. We assume that all three fluid phases always exist and that equilibrium is reached instantaneously, i.e. that the three fluid phases and the rock are in equilibrium at all times. There can be several injector and producer wells drilled into the reservoir. The injector wells inject water into the reservoir, and the producer wells produce all three fluid phases. The injectors and producers create pressure gradients which, together with gravity, cause an advective flow of the reservoir fluid.

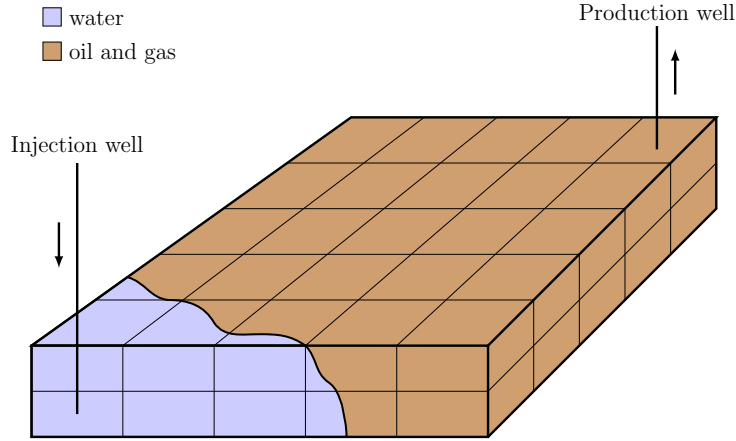


Fig. 4.1: An illustration of waterflooding. This figure originally appeared in [145].

The flow of energy in the reservoir is caused by 1) the advection of the fluid, 2) thermal conduction in the rock due to temperature gradients, and 3) exchange of energy with the surroundings of the reservoir. The isothermal model involves an assumption of constant temperature throughout the reservoir, and therefore, it does not contain an energy conservation equation. We consider the porosity field to be heterogeneous and the permeability field to be heterogeneous and anisotropic. The manipulated inputs are the bottom-hole pressures (BHPs) in the injectors and producers. In Chapter 10, we present optimized thermal and isothermal waterflooding strategies which maximize the amount of oil produced over a period of several years.

4.2 Thermal and compositional reservoir flow model

In this section, we describe the thermal and compositional reservoir flow model. We describe the phase equilibrium problem in Section 4.2.1 and the conservation equations in Section 4.2.2 and Section 4.2.3. We describe the well flow equations in Section 4.2.4, and we discuss Darcy's law, relative permeability, and viscosity in Section 4.2.5. Finally, we discretize the mass and energy conservation equations using a finite volume method in Section 4.2.7, and we demonstrate that the resulting equations are in a semi-explicit DAE form in Section 4.2.8.

4.2.1 Phase equilibrium

As mentioned previously, the reservoir fluid consists of water (w), oil (o), and gas (g). These three fluid phases are in thermal and mechanical equilibrium with the reservoir rock (r), i.e. $T^\alpha = T$ and $P^\alpha = P$ for $\alpha \in \{w, o, g, r\}$. Furthermore, the oil and the gas phases are in chemical equilibrium. The phase equilibrium problem is the UV flash optimization problem:

$$\max_{T, P, n^w, n^o, n^g} \quad S = S^w + S^o + S^g + S^r, \quad (4.1a)$$

$$\text{subject to} \quad U^w + U^o + U^g + U^r = U, \quad (4.1b)$$

$$V^w + V^o + V^g + V^r = V, \quad (4.1c)$$

$$n^w = n_w, \quad (4.1d)$$

$$n_k^o + n_k^g = n_k, \quad k = 1, \dots, N_C. \quad (4.1e)$$

$S^\alpha = S^\alpha(T, P, n^\alpha)$ is the entropy, $U^\alpha = U^\alpha(T, P, n^\alpha)$ is the internal energy, and $V^\alpha = V^\alpha(T, P, n^\alpha)$ is the volume of phase $\alpha \in \{w, o, g, r\}$. n^α is the composition (in moles) of phase $\alpha \in \{w, o, g, r\}$. The condition of maximal entropy is given by the second law of thermodynamics. U , n_w , and n_k

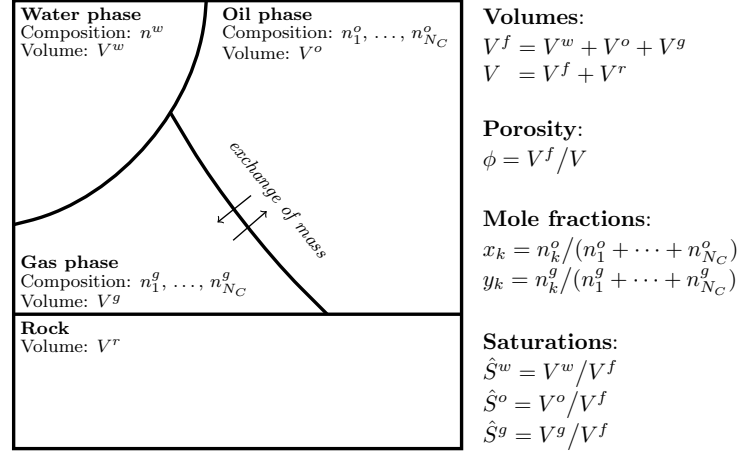


Fig. 4.2: An illustration of the fluid phases (water, oil, and gas) and the rock in a grid cell. This figure originally appeared in [145].

are the specified internal energy and the total amounts of moles of water and component k . They are given by the conservation equations. V is the specified volume, i.e. the volume of the grid cell in the discretized reservoir. The solution to the UV flash optimization problem (4.1) is the equilibrium temperature, pressure, and fluid phase compositions, i.e. T , P , and n^α for $\alpha \in \{w, o, g\}$. The rock composition, n^r , and the specified quantities, U , V , n_w , and $n = [n_1; \dots; n_{N_C}]$, are parameters in the phase equilibrium problem.

4.2.2 Conservation of mass

The mass conservation equations describe the temporal evolution of the molar concentrations of the water component, C_w , and the k 'th component, C_k . The conservation equations contain 1) a flux term that describes the flow in the reservoir and 2) a source term that describes the well flows:

$$\partial_t C_w = -\nabla \cdot \mathbf{N}^w + Q^w, \quad (4.2a)$$

$$\partial_t C_k = -\nabla \cdot \mathbf{N}_k + Q_k, \quad k = 1, \dots, N_C. \quad (4.2b)$$

\mathbf{N}_k is the molar flux of the k 'th component:

$$\mathbf{N}_k = x_k \mathbf{N}^o + y_k \mathbf{N}^g, \quad k = 1, \dots, N_C. \quad (4.3)$$

We describe the molar flux of phase $\alpha \in \{w, o, g\}$, \mathbf{N}^α , in Section 4.2.5. x_k and y_k are the oil and gas mole fractions of the k 'th component, respectively. The source terms are

$$Q^w = Q^{w,\text{inj}} - Q^{w,\text{prod}}, \quad (4.4a)$$

$$Q_k = -(x_k Q^{o,\text{prod}} + y_k Q^{g,\text{prod}}), \quad k = 1, \dots, N_C. \quad (4.4b)$$

In Section 4.2.4, we describe the molar water injection flow rate, $Q^{w,\text{inj}}$, and the molar production flow rate of phase $\alpha \in \{w, o, g\}$, $Q^{\alpha,\text{prod}}$.

4.2.3 Conservation of energy

First, we formulate the energy conservation equations without assuming thermal equilibrium between the fluid (f) and the rock (r). Therefore, we distinguish between their temperatures, T^f and T^r . Subsequently, we assume that energy is transferred instantaneously between the fluid and the rock such that they are always in thermal equilibrium, and we derive a single combined energy

conservation equation. The energy conservation equations describe the temporal evolution of the internal energy per unit volume of the fluid, u^f , and the rock, u^r :

$$\partial_t u^f = -\nabla \cdot \mathbf{N}_u^f + Q_u^f, \quad (4.5a)$$

$$\partial_t u^r = -\nabla \cdot \mathbf{N}_u^r + Q_u^r. \quad (4.5b)$$

The heat flux of the fluid is

$$\mathbf{N}_u^f = h^w \mathbf{N}^w + h^o \mathbf{N}^o + h^g \mathbf{N}^g, \quad (4.6)$$

where h^α is the molar enthalpy of phase $\alpha \in \{w, o, g\}$. We describe the conductive heat flux in the rock using Fourier's law of thermal conduction [74, Chap. 1]:

$$\mathbf{N}_u^r = -k_T^r \nabla T^r. \quad (4.7)$$

k_T^r is the thermal conductivity of the rock. The source term in (4.5a) represents 1) the flow of energy to and from the wells and 2) the flow of energy from the rock to the fluid:

$$Q_u^f = h^{w,\text{inj}} Q^{w,\text{inj}} - \sum_{\alpha \in \{w, o, g\}} h^\alpha Q^{\alpha, \text{prod}} + Q^{rf}. \quad (4.8)$$

$h^{w,\text{inj}}$ is the molar enthalpy of the water that is injected into the reservoir. We use Newton's law of cooling [74, Chap. 1] to model the thermal conduction from the rock to the fluid:

$$Q^{rf} = -k_T^{rf} (T^f - T^r). \quad (4.9)$$

k_T^{rf} is the thermal conductivity of the interface between the fluid and the rock. The source term in (4.5b) represents the flow of energy from the rock to 1) the fluid and 2) the surroundings (s) of the reservoir:

$$Q_u^r = -Q^{rf} - Q^{rs}. \quad (4.10)$$

We use Newton's law of cooling to describe the flow of energy from the rock to the surroundings of the reservoir:

$$Q^{rs} = -k_T^{rs} (T^s - T^r). \quad (4.11)$$

k_T^{rs} is the thermal conductivity of the interface between the rock and the reservoir surroundings. The temperature of the reservoir surroundings is T^s .

Now, we assume that the thermal conductivity of the interface between the fluid and the rock, k_T^{rf} , is infinite. Consequently, energy is transferred instantaneously between the fluid and the rock such that they are in thermal equilibrium at all times, i.e. $T^f = T^r = T$. We add the two conservation equations, (4.5a) and (4.5b), in order to obtain a single conservation equation for the internal energy of both the fluid and the rock, $u = u^f + u^r$:

$$\partial_t u = -\nabla \cdot \mathbf{N}_u + Q_u. \quad (4.12)$$

The heat flux and the source term in the combined energy conservation equation are

$$\mathbf{N}_u = h^w \mathbf{N}^w + h^o \mathbf{N}^o + h^g \mathbf{N}^g - k_T^r \nabla T, \quad (4.13a)$$

$$Q_u = h^{w,\text{inj}} Q^{w,\text{inj}} - \sum_{\alpha \in \{w, o, g\}} h^\alpha Q^{\alpha, \text{prod}} - Q^{rs}. \quad (4.13b)$$

4.2.4 Well flow rates

The injection and production well flow rates are only nonzero for grid cells in which the injection and production wells are perforated. The water injection flow rate is

$$Q^{w,\text{inj}} = \frac{1}{V} \text{WI} \rho^w \frac{k_r^w}{\mu^w} (P^{\text{bhp}} - P), \quad (4.14)$$

and the production flow rates are

$$Q^{\alpha,\text{prod}} = \frac{1}{V} \text{WI} \rho^\alpha \frac{k_r^\alpha}{\mu^\alpha} (P - P^{\text{bhp}}), \quad \alpha \in \{w, o, g\}. \quad (4.15)$$

V is the volume of the grid cell, WI is the well index, ρ^α is the molar density of phase $\alpha \in \{w, o, g\}$, and P^{bhp} is the BHP. We describe the relative permeability, k_r^α , and the viscosity, μ^α , in Section 4.2.5.

4.2.5 Darcy's law, relative permeability, and viscosity

The molar phase flux is

$$\mathbf{N}^\alpha = \rho^\alpha \mathbf{u}^\alpha, \quad \alpha \in \{w, o, g\}, \quad (4.16)$$

where the volumetric phase flux is given by Darcy's law:

$$\mathbf{u}^\alpha = -\frac{k_r^\alpha}{\mu^\alpha} \mathbf{K} (\nabla P - \bar{\rho}^\alpha g \nabla z), \quad \alpha \in \{w, o, g\}. \quad (4.17)$$

\mathbf{K} is the permeability tensor, and g is the gravity acceleration. z is the depth, and $\bar{\rho}^\alpha$ is the mass density of phase $\alpha \in \{w, o, g\}$. We describe the relative permeabilities using Stone's model II [47]. The relative permeability of phase $\alpha \in \{w, o, g\}$ is a function of the saturation, $\hat{S}^\alpha = V^\alpha / (V^w + V^o + V^g)$. Therefore, the relative permeabilities are functions of temperature, pressure, and all phase compositions:

$$k_r^\alpha = k_r^\alpha(T, P, n^w, n^o, n^g), \quad \alpha \in \{w, o, g\}. \quad (4.18)$$

We use the model by Lohrenz et al. [103] to describe the viscosities of the oil and the gas phases as functions of temperature, pressure, and phase composition.¹ We describe the water viscosity using the relation $(1/\mu^w)(\partial\mu^w/\partial P) = c_\mu^w$. c_μ^w is the viscosibility of the water phase. In conclusion, the viscosities depend on the temperature, pressure, and phase composition:

$$\mu^\alpha = \mu^\alpha(T, P, n^\alpha), \quad \alpha \in \{w, o, g\}. \quad (4.19)$$

We describe the relative permeabilities and the oil and gas viscosities in the paper in Appendix M.

4.2.6 Thermodynamics

We model the oil and the gas phase as nonideal mixtures, i.e. we use the expressions from Section 2.5 to describe their thermodynamic properties. We model the water phase as an ideal liquid, i.e. we use the expressions from Section 2.4. The thermodynamic properties of the rock, $H^r = H^r(T, P, n^r)$, $S^r = S^r(T, P, n^r)$, and $V^r = V^r(T, P, n^r)$, are based on a temperature-independent equation of state, i.e. $(1/V^r)(\partial V^r/\partial P) = c^r$ where c^r is the rock compressibility.

¹In equation (14) in the paper by Lohrenz et al. [103], they use an expression for the pure component viscosities by Jossi et al. [84]. However, there is a typo, i.e. the factor 0.40758 should be 0.040758 (a factor of 10 smaller).

4.2.7 Finite volume discretization

In this section, we first describe the finite volume discretization of partial differential equations in the form

$$\partial_t C = -\nabla \cdot \mathbf{N} + Q. \quad (4.20)$$

Next, we apply the finite volume discretization to the mass and energy conservation equations, (4.2) and (4.12), which are in this form. We discretize the reservoir using a set of grid cells, $\{\Omega_i\}_{i \in \mathcal{N}}$. The set \mathcal{N} contains the indices of the grid cells. We assume that 1) each grid cell is a polyhedron and 2) each face of the grid cell is shared by exactly two cells.

We integrate the partial differential equation (4.20) over each grid cell:

$$\partial_t \int_{\Omega_i} C dV = - \int_{\Omega_i} \nabla \cdot \mathbf{N} dV + \int_{\Omega_i} Q dV, \quad i \in \mathcal{N}. \quad (4.21)$$

We have interchanged integration and differentiation in the left-hand side of (4.21). Next, we apply Gauss' divergence theorem to the integral of the divergence of the flux:

$$\int_{\Omega_i} \nabla \cdot \mathbf{N} dV = \int_{\partial\Omega_i} \mathbf{N} \cdot \mathbf{n} dA, \quad i \in \mathcal{N}. \quad (4.22)$$

$\partial\Omega_i$ denotes the boundary of Ω_i , and \mathbf{n} denotes the outward normal vector. We express the boundary integral as the sum of integrals over each face of the grid cell:

$$\int_{\partial\Omega_i} \mathbf{N} \cdot \mathbf{n} dA = \sum_{j \in \mathcal{N}^{(i)}} \int_{\gamma_{ij}} \mathbf{N} \cdot \mathbf{n} dA, \quad i \in \mathcal{N}. \quad (4.23)$$

The set $\mathcal{N}^{(i)}$ contains the indices of the cells that share a face with Ω_i , and γ_{ij} is the face shared by Ω_i and Ω_j . We approximate the integral of the source term in (4.21) and the integral over the face in (4.23) using quadrature:

$$\int_{\Omega_i} Q dV \approx (QV)_i, \quad i \in \mathcal{N}, \quad (4.24a)$$

$$\int_{\gamma_{ij}} \mathbf{N} \cdot \mathbf{n} dA \approx (A\mathbf{N} \cdot \mathbf{n})_{ij}, \quad i \in \mathcal{N}, \quad j \in \mathcal{N}^{(i)}. \quad (4.24b)$$

V_i is the volume of the grid cell Ω_i , and A_{ij} is the area of the face γ_{ij} .

Now, we apply the finite volume discretization to the mass and energy conservation equations. We can evaluate the integrals of the internal energy per unit volume and the concentrations exactly:

$$\int_{\Omega_i} u dV = U_i, \quad i \in \mathcal{N}, \quad (4.25a)$$

$$\int_{\Omega_i} C_w dV = n_{w,i}, \quad i \in \mathcal{N}, \quad (4.25b)$$

$$\int_{\Omega_i} C_k dV = n_{k,i}, \quad i \in \mathcal{N}. \quad (4.25c)$$

We use the two-point flux approximation described by Lie [101] to evaluate the flux at the center of γ_{ij} in the right-hand side of (4.24b). For the mass and energy conservation equations, this approximation results in

$$(A\mathbf{N}_u \cdot \mathbf{n})_{ij} \approx - \sum_{\alpha \in \{w,o,g\}} (h^\alpha \Gamma \hat{H}^\alpha \Delta \Phi^\alpha)_{ij} + (\Gamma_T \Delta T)_{ij}, \quad i \in \mathcal{N}, \quad j \in \mathcal{N}^{(i)}, \quad (4.26a)$$

$$(A\mathbf{N}^w \cdot \mathbf{n})_{ij} \approx -(\Gamma \hat{H}^w \Delta \Phi^w)_{ij}, \quad i \in \mathcal{N}, \quad j \in \mathcal{N}^{(i)}, \quad (4.26b)$$

$$(A\mathbf{N}_k \cdot \mathbf{n})_{ij} \approx -(x_k \Gamma \hat{H}^o \Delta \Phi^o + y_k \Gamma \hat{H}^g \Delta \Phi^g)_{ij}, \quad i \in \mathcal{N}, \quad j \in \mathcal{N}^{(i)}. \quad (4.26c)$$

$\Delta T_{ij} = T_j - T_i$ is the temperature difference. The geometric part of the transmissibilities, Γ_{ij} , is

$$\Gamma_{ij} = A_{ij} \left(\hat{\Gamma}_{ij}^{-1} + \hat{\Gamma}_{ji}^{-1} \right)^{-1}, \quad i \in \mathcal{N}, \quad j \in \mathcal{N}^{(i)}, \quad (4.27a)$$

$$\hat{\Gamma}_{ij} = \left(\mathbf{K}_i \frac{c_{ij} - c_i}{|c_{ij} - c_i|^2} \right) \cdot \mathbf{n}_{ij}, \quad i \in \mathcal{N}, \quad j \in \mathcal{N}^{(i)}. \quad (4.27b)$$

c_i is the cell center, c_{ij} is the face center, and $\hat{\Gamma}_{ij}$ is referred to as the one-sided transmissibility. $\Gamma_{T,ij}$ is defined similarly:

$$\Gamma_{T,ij} = A_{ij} \left(\hat{\Gamma}_{T,ij}^{-1} + \hat{\Gamma}_{T,ji}^{-1} \right)^{-1}, \quad i \in \mathcal{N}, \quad j \in \mathcal{N}^{(i)}, \quad (4.28a)$$

$$\hat{\Gamma}_{T,ij} = \left(k_{T,i}^r \frac{c_{ij} - c_i}{|c_{ij} - c_i|^2} \right) \cdot \mathbf{n}_{ij}, \quad i \in \mathcal{N}, \quad j \in \mathcal{N}^{(i)}. \quad (4.28b)$$

The main difference between (4.27) and (4.28) is that the thermal rock conductivity, $k_{T,i}^r$, appears in place of the permeability tensor, \mathbf{K}_i , in (4.28). The difference in potential and the fluid part of the transmissibilities are given by

$$\Delta \Phi_{ij}^\alpha = (\Delta P - \rho^\alpha g \Delta z)_{ij}, \quad i \in \mathcal{N}, \quad j \in \mathcal{N}^{(i)}, \quad (4.29a)$$

$$\hat{H}_{ij}^\alpha = \begin{cases} (\rho^\alpha k_r^\alpha / \mu^\alpha)_i, & \Delta \Phi_{ij}^\alpha < 0, \\ (\rho^\alpha k_r^\alpha / \mu^\alpha)_j, & \Delta \Phi_{ij}^\alpha \geq 0, \end{cases} \quad i \in \mathcal{N}, \quad j \in \mathcal{N}^{(i)}. \quad (4.29b)$$

The pressure difference is $\Delta P_{ij} = P_j - P_i$, and the difference in depth is $\Delta z_{ij} = z_j - z_i$. We use averaging to approximate the density at the face center in (4.29a), i.e. $\rho_{ij}^\alpha \approx (\rho_i^\alpha + \rho_j^\alpha)/2$. We have upwinded the fluid part of the transmissibilities in (4.29b) in order to ensure numerical stability. We also upwind the phase enthalpy, h_{ij}^α , in (4.26a) and the oil and gas mole fractions, $x_{k,ij}$ and $y_{k,ij}$, in (4.26c).

In conclusion, we obtain a set of differential equations which describe the temporal evolution of the internal energy and the total amounts of moles in each grid cell:

$$\dot{U}_i = \sum_{j \in \mathcal{N}^{(i)}} \left(\sum_{\alpha \in \{w,o,g\}} (h^\alpha \Gamma \hat{H}^\alpha \Delta \Phi^\alpha)_{ij} + (\Gamma_T \Delta T)_{ij} \right) + (Q_u V)_i, \quad i \in \mathcal{N}, \quad (4.30a)$$

$$\dot{n}_{w,i} = \sum_{j \in \mathcal{N}^{(i)}} (\Gamma \hat{H}^w \Delta \Phi^w)_{ij} + (Q^w V)_i, \quad i \in \mathcal{N}, \quad (4.30b)$$

$$\dot{n}_{k,i} = \sum_{j \in \mathcal{N}^{(i)}} (x_k \Gamma \hat{H}^o \Delta \Phi^o + y_k \Gamma \hat{H}^g \Delta \Phi^g)_{ij} + (Q_k V)_i, \quad i \in \mathcal{N}. \quad (4.30c)$$

The internal energy, U_i , and the total amounts of moles, $n_{w,i}$ and $n_{k,i}$, appear as specified quantities in the UV flash optimization problem (4.1) for the i 'th grid cell.

4.2.8 Semi-explicit DAE form

The UV flash optimization problem (4.1) is in the form

$$\min_{y_i} f(y_i), \quad (4.31a)$$

$$\text{subject to } g(y_i) = x_i, \quad (4.31b)$$

$$h(y_i) = 0. \quad (4.31c)$$

The state variables, the algebraic variables, and the adjoint algebraic variables associated with the i 'th grid cell are

$$x_i = [U; n_w; n]_i \in \mathbb{R}^{2+N_C}, \quad (4.32a)$$

$$y_i = [T; P; n^w; n^o; n^g]_i \in \mathbb{R}^{3+2N_C}, \quad (4.32b)$$

$$z_i = [\eta; \mu]_i \in \mathbb{R}^{3+N_C}. \quad (4.32c)$$

η and μ in (4.32c) are Lagrange multipliers associated with the constraints (4.31b)-(4.31c). In Appendix A, we show that the first order optimality conditions of the optimization problem (4.31) are in the form

$$G_i(x_i, y_i, z_i) = 0. \quad (4.33)$$

The left-hand side of the discretized conservation equations (4.30) consists of the time derivatives of the state variables. The right-hand side depends on the algebraic variables in the i 'th cell and its neighboring cells, i.e. y_i and $\{y_j\}_{j \in \mathcal{N}(i)}$, as well as the manipulated inputs and the disturbance variables:

$$\dot{x}_i(t) = F_i(y_i(t), \{y_j(t)\}_{j \in \mathcal{N}(i)}, u_i(t), d_i(t)). \quad (4.34)$$

The manipulated inputs and the disturbance variables associated with the i 'th grid cell are

$$u_i = P_i^{\text{bhp}} \in \mathbb{R}, \quad (4.35a)$$

$$d_i = T_i^{\text{inj}} \in \mathbb{R}. \quad (4.35b)$$

The temperature of the injected water, T^{inj} , affects its molar enthalpy, $h^{w,\text{inj}}$, which enters into the expression for the source term (4.13b) in the energy conservation equation. If no well is perforated in the i 'th cell, there are no corresponding manipulated inputs. Similarly, if no injector is perforated in the i 'th cell, there are no corresponding disturbance variables. Finally, the collection of the phase equilibrium conditions (4.33) and the discretized conservation equations (4.34) for all grid cells is in the semi-explicit DAE form

$$G(x(t), y(t), z(t)) = 0, \quad (4.36a)$$

$$\dot{x}(t) = F(y(t), u(t), d(t)). \quad (4.36b)$$

4.3 Isothermal and compositional reservoir flow model

As mentioned previously, the isothermal and compositional reservoir flow model consists of 1) a set of mass conservation equations and 2) a set of phase equilibrium conditions. The mass conservation equations in the isothermal model are identical to the mass conservation equations in the thermal model described in Section 4.2. The two key assumptions in the isothermal model are that 1) all involved thermal conductivities are infinite and 2) the heat capacity of the surroundings is infinite. The consequence of the first assumption is that energy is transferred instantaneously between the reservoir fluid, the reservoir rock, and the surroundings, i.e. they reach thermal equilibrium instantaneously. The consequence of the second assumption is that the temperature of the surroundings is unaffected by the transfer of energy to or from the reservoir fluid and rock. Therefore, the temperature of the fluid and the rock will be exactly the temperature of the surroundings. We describe the phase equilibrium problem in Section 4.3.1, and we argue that the isothermal and compositional model is also in a semi-explicit DAE form in Section 4.3.2.

4.3.1 Phase equilibrium

The phase equilibrium problem in the isothermal and compositional model is the VT flash optimization problem:

$$\min_{P, n^w, n^o, n^g} A = A^w + A^o + A^g + A^r, \quad (4.37a)$$

$$\text{subject to } V^w + V^o + V^g + V^r = V, \quad (4.37b)$$

$$n^w = n_w, \quad (4.37c)$$

$$n_k^o + n_k^g = n_k, \quad k = 1, \dots, N_C. \quad (4.37d)$$

As mentioned previously, the condition that the Helmholtz energy is minimal for an isothermal system in equilibrium is derived from the second law of thermodynamics. $A^\alpha = A^\alpha(T, P, n^\alpha)$ is the Helmholtz energy of phase $\alpha \in \{w, o, g, r\}$. The solution to the VT flash optimization problem (4.37) is the equilibrium pressure, P , and phase compositions, n^α for $\alpha \in \{w, o, g\}$, which minimize the Helmholtz energy while satisfying the constraints on the volume and the total amounts of moles at the specified temperature, T . The total amounts of moles, n_w and n_k , are determined by the mass conservation equations.

4.3.2 Semi-explicit DAE form

The VT flash optimization problem (4.37) is in the same form as the UV flash optimization problem (4.1) in the thermal model:

$$\min_{y_i} f(y_i), \quad (4.38a)$$

$$\text{subject to } g(y_i) = x_i, \quad (4.38b)$$

$$h(y_i) = 0. \quad (4.38c)$$

The state variables, the algebraic variables, and the adjoint algebraic variables associated with the i 'th grid cell are

$$x_i = [n_w; n]_i \in \mathbb{R}^{1+N_C}, \quad (4.39a)$$

$$y_i = [P; n^w; n^o; n^g]_i \in \mathbb{R}^{2+2N_C}, \quad (4.39b)$$

$$z_i = [\eta; \mu]_i \in \mathbb{R}^{2+N_C}, \quad (4.39c)$$

and the manipulated inputs are the same as in the thermal model:

$$u_i = P_i^{\text{bhp}} \in \mathbb{R}. \quad (4.39d)$$

There are no disturbance variables in the isothermal model. The main differences between the variables in the thermal and the isothermal models are that in the isothermal model 1) the internal energy is not a state variable, 2) the temperature is a parameter instead of an algebraic variable, and 3) there is one less Lagrange multiplier because there is one less constraint in the VT flash than in the UV flash. In Section 4.2.8, we demonstrated that the thermal model is in the semi-explicit DAE form

$$G(x(t), y(t), z(t)) = 0, \quad (4.40a)$$

$$\dot{x}(t) = F(y(t), u(t), d(t)). \quad (4.40b)$$

The same arguments are valid for the isothermal model, i.e. the phase equilibrium conditions are in the same form, the left-hand side of the discretized mass conservation equations contains the time derivatives of the state variables, and the right-hand side is independent of the state variables. In conclusion, the isothermal model is in the form (4.40).

4.4 Summary

In this chapter, we have presented 1) a thermal and compositional reservoir flow model and 2) an isothermal and compositional reservoir flow model. Both models consist of 1) a set of conservation equations (partial differential equations) and 2) phase equilibrium conditions (algebraic equations) derived from the second law of thermodynamics. The phase equilibrium problems that are relevant to the thermal and isothermal models are the UV and the VT flash, respectively. We discretize the conservation equations in space using a finite volume method, and we have demonstrated that the

resulting equations, together with the phase equilibrium conditions, are in a semi-explicit DAE form. In Part II, we describe algorithms for simulation, state estimation, dynamic optimization, and NMPC of models in this form. The two reservoir flow models describe waterflooding processes, and in Chapter 10, we present examples of optimized waterflooding strategies.

Part II

Algorithms

Simulation

In this chapter, we describe numerical methods for the solution of IVPs (i.e. numerical simulations) that involve deterministic and stochastic semi-explicit DAEs in the forms (1.1) and (1.3), respectively. We assume that the DAEs are of index 1, i.e. that the algebraic equations, $G(x, y, z) = 0$, can be solved for $y = y(x)$ and $z = z(x)$ when x is specified. This is true for the flash separation process and the reservoir flow processes that we describe in Chapter 3 and Chapter 4.

In Section 5.1, we describe the deterministic and the stochastic IVPs, and in Section 5.2, we present two approaches for the numerical solution of the deterministic IVP: 1) a simultaneous approach and 2) a nested approach. We use Euler's implicit method in both approaches. In the simultaneous approach, the discretized differential equations and the algebraic equations are solved simultaneously. In the nested approach, the solution of the algebraic equations is nested into the solution of the discretized differential equations. In Section 5.3, we consider the numerical solution of the stochastic IVP which refers to the computation of a single realization of the stochastic state variables, algebraic variables, and adjoint algebraic variables.¹ The numerical method for the stochastic IVP is based on a semi-implicit discretization scheme which solves the discretized differential equations and the algebraic equations simultaneously. We briefly summarize this chapter in Section 5.4.

The state estimation algorithms that we describe in Chapter 6 involve the numerical solution of either deterministic or stochastic IVPs. Furthermore, the dynamic optimization algorithm that we present in Chapter 7 involves the numerical solution of a deterministic IVP. In Chapter 9, we compare the efficiency of the dynamic optimization algorithm when using the simultaneous approach and the nested approach. In the same chapter, we test the state estimation algorithms and the NMPC algorithm using the flash separation process, and we use the numerical solution of a stochastic IVP to represent the true real-life process. The content of this chapter is based on content from the papers in Appendix C, E, G, I, J, K, and N.

5.1 Problem formulation

In this section, we describe the deterministic and the stochastic IVPs that we consider in this work. We consider deterministic IVPs in the form

$$x(t_0) = x_0, \tag{5.1a}$$

$$G(x(t), y(t), z(t)) = 0, \quad t \in [t_0, t_f], \tag{5.1b}$$

$$\dot{x}(t) = F(y(t), u(t), d(t)), \quad t \in [t_0, t_f]. \tag{5.1c}$$

(5.1a) is an initial condition for the states at the initial time, t_0 , and t_f is the final time.

¹Whereas deterministic IVPs have a single unique solution (under suitable conditions), stochastic IVPs allow for an entire family of solutions (or realizations) which can be characterized by its distribution (as a function of time).

In contrast to the deterministic IVPs, the stochastic IVPs represent the uncertainty in the process, and they can be derived by assuming that expressions or parameters in the differential equations (5.1c) are stochastic (i.e. uncertain). Examples of sources of uncertainty are 1) unmodeled dynamics, 2) uncertain process inputs, and 3) limited accuracy of model parameters. We consider stochastic IVPs in the form

$$\mathbf{x}(t_0) \sim N(\hat{x}_0, P_0), \quad (5.2a)$$

$$G(\mathbf{x}(t), \mathbf{y}(t), \mathbf{z}(t)) = 0, \quad t \in [t_0, t_f], \quad (5.2b)$$

$$d\mathbf{x}(t) = F(\mathbf{y}(t), u(t), d(t))dt + \sigma(\mathbf{y}(t), u(t), d(t))d\boldsymbol{\omega}(t), \quad t \in [t_0, t_f]. \quad (5.2c)$$

The initial condition (5.2a) requires the states to be normally distributed with mean \hat{x}_0 and covariance P_0 at the initial time, t_0 . As for the deterministic IVP, t_f is the final time. $\boldsymbol{\omega}(t)$ is a standard Wiener process, i.e. the incremental covariance of $\boldsymbol{\omega}(t)$ is Idt . The right-hand side of the stochastic differential equation (5.2c) contains a drift term (the first term) and a diffusion term (the second term). The drift term represents the dynamics of the process, and the diffusion term represents the uncertainty in the process.

In this work, we assume a ZOH parametrization of the manipulated inputs, $u(t)$, and the disturbance variables, $d(t)$:

$$u(t) = u_k, \quad t \in [t_k, t_{k+1}[, \quad k = 0, \dots, N-1, \quad (5.3a)$$

$$d(t) = \hat{d}_k, \quad t \in [t_k, t_{k+1}[, \quad k = 0, \dots, N-1. \quad (5.3b)$$

We often refer to both $u(t)$ and $\{u_k\}_{k=0}^{N-1}$ as the manipulated inputs, and $\{\hat{d}_k\}_{k=0}^{N-1}$ are predictions of the disturbance variables. N is the number of control intervals.

5.2 Numerical solution of deterministic initial value problems

In this section, we present the simultaneous and the nested approach for the numerical solution of deterministic IVPs in the form (5.1) subject to the ZOH parametrizations (5.3) of the manipulated inputs and the disturbance variables. For notational simplicity, we assume that the time steps in the two approaches coincide with the boundaries of the control intervals, i.e. that there is one time step per control interval.

5.2.1 Simultaneous approach

In the simultaneous approach, we do not consider the algebraic variables, $y(t)$, and the adjoint algebraic variables, $z(t)$, to be implicit functions of the states, $x(t)$, as we do in the nested approach in Section 5.2.2. We use Euler's implicit method to approximate the solution to (5.1) for given initial condition, x_0 , manipulated inputs, $\{u_k\}_{k=0}^{N-1}$, and predicted disturbances, $\{\hat{d}_k\}_{k=0}^{N-1}$, i.e. we sequentially solve the residual equations,

$$\begin{aligned} R_{k+1} &= R_{k+1}(w_{k+1}) = R_{k+1}(w_{k+1}; x_k, u_k, \hat{d}_k) \\ &= R_{k+1}(x_{k+1}, y_{k+1}, z_{k+1}; x_k, u_k, \hat{d}_k) \\ &= \begin{bmatrix} D_{k+1}(x_{k+1}, x_k, y_{k+1}, u_k, \hat{d}_k) \\ G(x_{k+1}, y_{k+1}, z_{k+1}) \end{bmatrix} = 0, \quad k = 0, \dots, N-1, \end{aligned} \quad (5.4)$$

for $w_{k+1} = [x_{k+1}; y_{k+1}; z_{k+1}] \approx [x(t_{k+1}); y(t_{k+1}); z(t_{k+1})]$. $D_{k+1} = 0$ are the discretized differential equations where

$$\begin{aligned} D_{k+1} &= D_{k+1}(x_{k+1}, x_k, y_{k+1}, u_k, \hat{d}_k) \\ &= x_{k+1} - F(y_{k+1}, u_k, \hat{d}_k)\Delta t_k - x_k. \end{aligned} \quad (5.5)$$

We solve the residual equations, $R_{k+1} = 0$, with an inexact Newton method, i.e. we sequentially update the estimate of w_{k+1} by

$$w_{k+1}^{m+1} = w_{k+1}^m + \Delta w_{k+1}^m. \quad (5.6)$$

The Newton update, Δw_{k+1}^m , is the solution to the linear system

$$M_R \Delta w_{k+1}^m = -R_{k+1}(w_{k+1}^m). \quad (5.7)$$

We use $w_{k+1}^0 = w_k$ as the initial guess for the Newton iterations.² The iteration matrix is

$$M_R \approx \frac{\partial R_{k+1}}{\partial w_{k+1}} = \begin{bmatrix} \mathbf{I} & -\frac{\partial F}{\partial y} \Delta t_k & 0 \\ \frac{\partial G}{\partial x} & \frac{\partial G}{\partial y} & \frac{\partial G}{\partial z} \end{bmatrix}. \quad (5.8)$$

The Jacobian matrix in (5.8) contains an identity matrix. Next, we describe how we exploit this structure when we solve the linear system for the Newton update in (5.7).

5.2.1.1 Efficient computation of the Newton update. Because of the structure of the Jacobian matrix in (5.8), we can obtain an explicit expression for Δx_{k+1}^m from (5.7):

$$\Delta x_{k+1}^m = \left(\frac{\partial F}{\partial y} \Delta t_k \right) \Delta y_{k+1}^m - D_{k+1}. \quad (5.9)$$

Next, we obtain a reduced linear system for Δy_{k+1}^m and Δz_{k+1}^m using (5.7) and (5.9):

$$\bar{M} \begin{bmatrix} \Delta y_{k+1}^m \\ \Delta z_{k+1}^m \end{bmatrix} = \frac{\partial G}{\partial x} D_{k+1} - G(x_{k+1}^m, y_{k+1}^m, z_{k+1}^m). \quad (5.10)$$

The reduced iteration matrix is

$$\bar{M} \approx \begin{bmatrix} \frac{\partial G}{\partial y} + \frac{\partial G}{\partial x} \frac{\partial F}{\partial y} \Delta t_k & \frac{\partial G}{\partial z} \end{bmatrix}. \quad (5.11)$$

To summarize, we first solve (5.10) for Δy_{k+1}^m and Δz_{k+1}^m , and next, we compute Δx_{k+1}^m from (5.9). Finally, the Newton update is given by $\Delta w_{k+1}^m = [\Delta x_{k+1}^m; \Delta y_{k+1}^m; \Delta z_{k+1}^m]$.

5.2.2 Nested approach

In the nested approach, we formulate the deterministic IVP (5.1) as

$$x(t_0) = x_0, \quad (5.12a)$$

$$\dot{x}(t) = F(y(x(t)), u(t), d(t)), \quad (5.12b)$$

where the algebraic variables, $y(x(t))$, and the adjoint algebraic variables, $z(x(t))$, are implicit functions of the state variables, $x(t)$, given by the algebraic equations,

$$G(x(t), y(x(t)), z(x(t))) = 0. \quad (5.13)$$

This formulation of the IVP is equivalent to the formulation (5.1). As in the simultaneous approach, we use Euler's implicit method to approximate the solution to (5.12) for given initial condition, x_0 , manipulated inputs, $\{u_k\}_{k=0}^{N-1}$, and predicted disturbances, $\{\hat{d}_k\}_{k=0}^{N-1}$. We sequentially solve the residual equations,

$$\begin{aligned} D_{k+1} &= D_{k+1}(x_{k+1}) = D_{k+1}(x_{k+1}, x_k, y_{k+1}, u_k, \hat{d}_k) \\ &= x_{k+1} - F(y_{k+1}, u_k, \hat{d}_k) \Delta t_k - x_k = 0, \end{aligned} \quad k = 0, \dots, N-1, \quad (5.14)$$

²Note that for $k = 0$, y_0 and z_0 (which are part of w_0) are not available from the previous time step. They can be computed by solving $G(x_0, y_0, z_0) = 0$ using an inexact Newton method as described in Section 5.2.2.1. However, this also requires an initial guess (of y_0 and z_0).

for the state variables, $x_{k+1} \approx x(t_{k+1})$, where $y_{k+1} = y(x_{k+1}) \approx y(x(t_{k+1}))$. We solve the residual equations, $D_{k+1} = 0$, with an inexact Newton method, i.e. we sequentially update the estimate of the states by

$$x_{k+1}^{m+1} = x_{k+1}^m + \Delta x_{k+1}^m. \quad (5.15)$$

The Newton update, Δx_{k+1}^m , is the solution to the linear system

$$M_D \Delta x_{k+1}^m = -D_{k+1}(x_{k+1}^m). \quad (5.16)$$

We use $x_{k+1}^0 = x_k$ as initial guess, and the iteration matrix is

$$M_D \approx \frac{\partial D_{k+1}}{\partial x_{k+1}} = \mathbf{I} - \left(\frac{\partial F}{\partial y} \Delta t_k \right) \frac{\partial y_{k+1}}{\partial x_{k+1}}. \quad (5.17)$$

Next, we describe the numerical solution of the algebraic equations (5.13) for the algebraic and adjoint algebraic variables which is necessary when evaluating the right-hand side function in the differential equations (5.12b), F , and its Jacobian. We also describe the computation of the sensitivities of the algebraic variables which are necessary when evaluating the Jacobian matrix in (5.17).

5.2.2.1 Solution of the algebraic equations. We use an inexact Newton method to solve the algebraic equations,

$$G(x_{k+1}, y_{k+1}, z_{k+1}) = 0, \quad (5.18)$$

for y_{k+1} and $z_{k+1} = z(x_{k+1}) \approx z(x(t_{k+1}))$, i.e. we sequentially update the estimates of the algebraic and adjoint algebraic variables by

$$\begin{bmatrix} y_{k+1}^{l+1} \\ z_{k+1}^{l+1} \end{bmatrix} = \begin{bmatrix} y_{k+1}^l \\ z_{k+1}^l \end{bmatrix} + \begin{bmatrix} \Delta y_{k+1}^l \\ \Delta z_{k+1}^l \end{bmatrix}. \quad (5.19)$$

The Newton updates, Δy_{k+1}^l and Δz_{k+1}^l , are the solutions to the linear system

$$M_G \begin{bmatrix} \Delta y_{k+1}^l \\ \Delta z_{k+1}^l \end{bmatrix} = -G(x_{k+1}, y_{k+1}^l, z_{k+1}^l). \quad (5.20)$$

We use $y_{k+1}^0 = y_k$ and $z_{k+1}^0 = z_k$ as initial guesses,³ and the iteration matrix is

$$M_G \approx \begin{bmatrix} \frac{\partial G}{\partial y} & \frac{\partial G}{\partial z} \end{bmatrix}. \quad (5.21)$$

We compute the sensitivities with respect to the states, $\frac{\partial y_{k+1}}{\partial x_{k+1}}$ and $\frac{\partial z_{k+1}}{\partial x_{k+1}}$, using the implicit function theorem:

$$\begin{bmatrix} \frac{\partial y_{k+1}}{\partial x_{k+1}} \\ \frac{\partial z_{k+1}}{\partial x_{k+1}} \end{bmatrix} = - \begin{bmatrix} \frac{\partial G}{\partial y} & \frac{\partial G}{\partial z} \end{bmatrix}^{-1} \frac{\partial G}{\partial x}. \quad (5.22)$$

The sensitivities of the algebraic variables are necessary for evaluating the Jacobian matrix in (5.17). The sensitivities of the adjoint algebraic variables are not necessary in the nested approach. However, $\frac{\partial y_{k+1}}{\partial x_{k+1}}$ cannot be isolated in (5.22). Therefore, it is necessary to compute $\frac{\partial z_{k+1}}{\partial x_{k+1}}$ as well.

³Note that for $k = 0$, y_0 and z_0 are not available from the previous time step.

5.3 Numerical solution of stochastic initial value problems

In this section, we describe the numerical solution of the stochastic IVP (5.2). We use a semi-implicit numerical scheme to approximate a single realization of the solution to (5.2) when the mean and covariance of the initial states, \hat{x}_0 and P_0 , the manipulated inputs, $\{u_k\}_{k=0}^{N-1}$, and the predictions of the disturbance variables, $\{\hat{d}_k\}_{k=0}^{N-1}$, are given. In this scheme, we discretize the drift term in the stochastic differential equation (5.2c) with a right rectangle rule and the diffusion term with a left rectangle rule (this is the reason that we refer to the scheme as semi-implicit):

$$\begin{aligned} D_{k+1} &= D_{k+1}(x_{k+1}, x_k, y_{k+1}, y_k, u_k, \hat{d}_k) \\ &= x_{k+1} - F(y_{k+1}, u_k, \hat{d}_k)\Delta t_k - \sigma(y_k, u_k, \hat{d}_k)\Delta\omega_k - x_k = 0. \end{aligned} \quad (5.23)$$

The initial states, x_0 , are sampled from $N(\hat{x}_0, P_0)$, and the increment $\Delta\omega_k$ is sampled from $N(0, I\Delta t_k)$. We sequentially solve the residual equations,

$$\begin{aligned} R_{k+1} &= R_{k+1}(w_{k+1}) = R_{k+1}(w_{k+1}; x_k, y_k, u_k, \hat{d}_k) \\ &= R_{k+1}(x_{k+1}, y_{k+1}, z_{k+1}; x_k, y_k, u_k, \hat{d}_k) \\ &= \begin{bmatrix} D_{k+1}(x_{k+1}, x_k, y_{k+1}, y_k, u_k, \hat{d}_k) \\ G(x_{k+1}, y_{k+1}, z_{k+1}) \end{bmatrix} = 0, \quad k = 0, \dots, N-1, \end{aligned} \quad (5.24)$$

for $w_{k+1} = [x_{k+1}; y_{k+1}; z_{k+1}]$ using an inexact Newton method, i.e. we sequentially update the estimate of w_{k+1} by

$$w_{k+1}^{m+1} = w_{k+1}^m + \Delta w_{k+1}^m. \quad (5.25)$$

The Newton update, Δw_{k+1}^m , is the solution to the linear system

$$M_R \Delta w_{k+1}^m = -R_{k+1}(w_{k+1}^m). \quad (5.26)$$

As in the simultaneous approach for the deterministic IVP, we use $w_{k+1}^0 = w_k$ as initial guess (see footnote 2 on page 41). The iteration matrix is

$$M_R \approx \frac{\partial R_{k+1}}{\partial w_{k+1}} = \begin{bmatrix} \mathbf{I} & -\frac{\partial F}{\partial y}\Delta t_k & 0 \\ \frac{\partial G}{\partial x} & \frac{\partial G}{\partial y} & \frac{\partial G}{\partial z} \end{bmatrix}. \quad (5.27)$$

The Jacobian matrix in (5.27) is identical to the Jacobian matrix in (5.8) in the simultaneous approach for the deterministic IVP. Therefore, we use the approach described in Section 5.2.1.1 to exploit the structure when we solve the linear system (5.26) for the Newton update.⁴

5.4 Summary

In this chapter, we have described 1) a simultaneous and a nested approach for the numerical solution of deterministic IVPs and 2) a simultaneous approach for the numerical solution of stochastic IVPs. The deterministic and the stochastic IVPs that we consider involve semi-explicit DAEs in the forms (1.1) and (1.3), respectively. The two approaches for the deterministic IVPs use Euler's implicit method, and the numerical method for the stochastic IVPs uses a semi-implicit discretization scheme. For notational simplicity, we have assumed that there is only one time step per control interval when we describe the methods. The papers in Appendix N and K describe the simultaneous approach for the deterministic IVPs and the numerical method for the stochastic IVPs, respectively, with multiple time steps per control interval. The numerical solution of both

⁴Note that the right-hand sides of the equations for the Newton updates, (5.7) and (5.26), are not identical because there is no diffusion term in the differential equation in the deterministic IVP.

deterministic and stochastic IVPs is relevant to the state estimation algorithms that we describe in Chapter 6, and the dynamic optimization algorithm that we present in Chapter 7 involves the numerical solution of a deterministic IVP. In Chapter 9, we compare the efficiency of using the simultaneous and the nested approach in the dynamic optimization algorithm. Furthermore, we use a stochastic IVP to simulate the real-life flash separation process when we test the state estimation algorithms and the NMPC algorithm in the same chapter.

State estimation

In this chapter, we consider state estimation of continuous-discrete systems that involve stochastic semi-explicit DAEs in the form (1.3).¹ We describe the EKF, UKF, PF, and EnKF for state estimation of such systems. These algorithms consist of 1) a measurement-update and 2) a time-update. In the measurement-update, the information from the measurements is incorporated into the state estimate using a model of the sensors (also called measurement devices). In the time-update, the dynamical model of the process is used to propagate the state estimate through the time in between measurements. The EKF is based on a linearization of the nonlinear model equations. Consequently, it can suffer from limited accuracy for highly nonlinear systems. The UKF uses a set of deterministic samples (or particles) to improve the accuracy over the EKF. However, the accuracy of the UKF can also be insufficient for severely nonlinear systems. The PF uses random particles, and it can be more precise than the EKF and the UKF if sufficiently many particles are used. The EnKF is in fact a type of PF, and the time-updates in the two algorithms are identical. However, the measurement-updates are different.

In Section 6.1, we describe the state estimation problem. Furthermore, we discuss the estimation of unknown disturbances and parameters. We describe the EKF in Section 6.2, the UKF in Section 6.3, the PF in Section 6.4, and the EnKF in Section 6.5. Finally, we summarize this chapter in Section 6.6. State estimation is central to the NMPC algorithm that we describe in Chapter 8, and we compare the accuracy and the computational performance of the four state estimation algorithms in Chapter 9 using the flash separation process described in Chapter 3. The content of this chapter is based on the papers in Appendix J, K, and N.

6.1 Problem formulation

State estimation is concerned with the reconstruction of the state of a dynamical process using measurements, a model of the process, and a model of the sensors. We formulate the state estimation algorithms for stochastic semi-explicit DAEs in the form (1.3) which we repeat here for convenience:

$$G(\mathbf{x}(t), \mathbf{y}(t), \mathbf{z}(t)) = 0, \quad (6.1a)$$

$$d\mathbf{x}(t) = F(\mathbf{y}(t), u(t), d(t))dt + \sigma(\mathbf{y}(t), u(t), d(t))d\boldsymbol{\omega}(t). \quad (6.1b)$$

At the initial time, t_0 , the states are normally distributed, i.e. $\mathbf{x}(t_0) \sim N(x_0, P_0)$. $\boldsymbol{\omega}(t)$ is a standard Wiener process (which means that its incremental covariance is Idt), and the noisy measurements,

¹Continuous-discrete systems are systems with continuous dynamics and discrete measurements.

$\mathbf{y}^m(t_k)$, of the process outputs, $\mathbf{z}^m(t_k)$, at time t_k are given by the model of the sensors:

$$\mathbf{z}^m(t_k) = H(\mathbf{y}(t_k)), \quad (6.2a)$$

$$\mathbf{y}^m(t_k) = \mathbf{z}^m(t_k) + \mathbf{v}(t_k). \quad (6.2b)$$

The measurement noise, $\mathbf{v}_k = \mathbf{v}(t_k)$, is normally distributed, i.e. $\mathbf{v}_k \sim N(0, T_k)$. To summarize, the objective of the state estimation algorithms that we present in this chapter is to estimate the state variables, algebraic variables, and adjoint algebraic variables of a real-life process using measurements together with the models (6.1) and (6.2).

As mentioned previously, the algorithms consist of a measurement-update and a time-update. The objective of the measurement-update is to compute the filtered states, $\hat{x}_{k|k}$, and the corresponding covariance matrix, $P_{k|k}$, which denote the estimates of the states and the covariance matrix at time t_k given information up to (and including) time t_k . The objective of the time-update is to compute the one-step ahead prediction of the states, $\hat{x}_{k+1|k}$, and the corresponding covariance matrix, $P_{k+1|k}$, which denote the estimates at time t_{k+1} given information up to and including time t_k . The estimates of the algebraic variables can be computed from the estimates of the states. We use the above subscript notation for all relevant variables and covariance matrices.

6.1.1 Estimation of unknown disturbances and parameters

Below, we describe an approach for estimating unknown disturbances using the state estimation algorithms that we present in this chapter. The same approach can be used for estimating parameters. We include a model of the (now stochastic) disturbance variables:

$$G(\mathbf{x}(t), \mathbf{y}(t), \mathbf{z}(t)) = 0, \quad (6.3a)$$

$$d\mathbf{x}(t) = F(\mathbf{y}(t), u(t), \mathbf{d}(t))dt + \sigma(\mathbf{y}(t), u(t), \mathbf{d}(t))d\boldsymbol{\omega}(t), \quad (6.3b)$$

$$d\mathbf{d}(t) = \sigma_d d\boldsymbol{\omega}_d(t). \quad (6.3c)$$

The diffusion coefficient, σ_d , is used for tuning purposes, and $\boldsymbol{\omega}_d(t)$ is a standard Wiener process (i.e. its incremental covariance is Idt). The state vector is augmented with the disturbance variables in order to obtain an augmented system:

$$\tilde{G}(\tilde{\mathbf{x}}(t), \mathbf{y}(t), \mathbf{z}(t)) = 0, \quad (6.4a)$$

$$d\tilde{\mathbf{x}}(t) = \tilde{F}(\tilde{\mathbf{x}}(t), \mathbf{y}(t), u(t))dt + \tilde{\sigma}(\tilde{\mathbf{x}}(t), \mathbf{y}(t), u(t))d\tilde{\boldsymbol{\omega}}(t). \quad (6.4b)$$

The augmented system (6.4) is no longer in the form (6.1). However, the state estimation algorithms that we describe can also be formulated for systems in the form (6.4), and the estimates of the augmented states will contain estimates of both the states and the disturbance variables. In the paper in Appendix N, we describe the EKF for simultaneous estimation of the states and unknown disturbances. In that description, we exploit the structure of the augmented system (6.4).

As mentioned, the above approach can also be used for parameter estimation. Whereas the disturbance variables only enter into the drift term and the diffusion term in the stochastic differential equation (6.1b), the parameters may also enter into the algebraic equations (6.1a) and the model of the sensors in (6.2a). This must be accounted for in the formulation of the algorithms for combined state and parameter estimation.

6.2 The extended Kalman filter

In the EKF, the original Kalman filter equations are applied to a linearization of the nonlinear model equations. The initial estimate of the states and the corresponding covariance matrix are

initialized with the mean and covariance of the initial states:

$$\hat{x}_{0|-1} = x_0, \quad (6.5a)$$

$$P_{0|-1} = P_0. \quad (6.5b)$$

The initial estimates of the algebraic variables, $\hat{y}_{0|-1}$, and the adjoint algebraic variables, $\hat{z}_{0|-1}$, are obtained by solving the algebraic equations,

$$G(\hat{x}_{0|-1}, \hat{y}_{0|-1}, \hat{z}_{0|-1}) = 0. \quad (6.6)$$

6.2.1 Measurement-update

The (one-step ahead) predictions of the outputs, the measurements, and the approximate covariance of the measurements are

$$\hat{z}_{k|k-1}^m = H(\hat{y}_{k|k-1}), \quad (6.7a)$$

$$\hat{y}_{k|k-1}^m = \hat{z}_{k|k-1}^m, \quad (6.7b)$$

$$T_{k|k-1} = C_k P_{k|k-1} C_k' + T_k. \quad (6.7c)$$

The estimate of the algebraic variables, $\hat{y}_{k|k-1}$, and the covariance matrix $P_{k|k-1}$ are available from the previous time-update. As mentioned previously, T_k is the covariance of the normally distributed measurement noise at time t_k , i.e. of \mathbf{v}_k . The expression (6.7c) is obtained from a linearization of the sensor model in (6.2), and the matrix C_k is

$$\begin{aligned} C_k &= \frac{\partial H}{\partial x}(\hat{y}_{k|k-1}) \\ &= \frac{\partial H}{\partial y}(\hat{y}_{k|k-1}) \frac{\partial \hat{y}_{k|k-1}}{\partial \hat{x}_{k|k-1}}. \end{aligned} \quad (6.8)$$

The sensitivities of the algebraic and adjoint algebraic variables are obtained by using the implicit function theorem, i.e. by solving the linear system

$$\begin{bmatrix} \frac{\partial G}{\partial y} & \frac{\partial G}{\partial z} \end{bmatrix} \begin{bmatrix} \frac{\partial \hat{y}_{k|k-1}}{\partial \hat{x}_{k|k-1}} \\ \frac{\partial \hat{z}_{k|k-1}}{\partial \hat{x}_{k|k-1}} \end{bmatrix} = -\frac{\partial G}{\partial x}. \quad (6.9)$$

The Jacobian matrices are evaluated at $\hat{x}_{k|k-1}$, $\hat{y}_{k|k-1}$, and $\hat{z}_{k|k-1}$ which are available from the previous time-update. The innovation is

$$e_k = y_k^m - \hat{y}_{k|k-1}^m, \quad (6.10)$$

where y_k^m are the measurements obtained at time t_k . The Kalman filter gain matrix is given by

$$K_{fx,k} = P_{k|k-1} C_k' T_{k|k-1}^{-1}. \quad (6.11)$$

The filtered state variables and the corresponding approximate covariance matrix are

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_{fx,k} e_k, \quad (6.12a)$$

$$P_{k|k} = P_{k|k-1} - K_{fx,k} T_{k|k-1} K_{fx,k}'. \quad (6.12b)$$

The estimates of the algebraic and adjoint algebraic variables corresponding to the filtered states, $\hat{y}_{k|k}$ and $\hat{z}_{k|k}$, are the solutions to the algebraic equations,

$$G(\hat{x}_{k|k}, \hat{y}_{k|k}, \hat{z}_{k|k}) = 0, \quad (6.13)$$

and the corresponding covariance matrices are

$$P_{y,k|k} = \Phi_{yx}(t_k, t_k)P_{k|k}\Phi_{yx}(t_k, t_k)', \quad (6.14a)$$

$$P_{z,k|k} = \Phi_{zx}(t_k, t_k)P_{k|k}\Phi_{zx}(t_k, t_k)'. \quad (6.14b)$$

The sensitivities, $\Phi_{yx}(t_k, t_k) = \frac{\partial \hat{y}_{k|k}}{\partial \hat{x}_{k|k}}$ and $\Phi_{zx}(t_k, t_k) = \frac{\partial \hat{z}_{k|k}}{\partial \hat{x}_{k|k}}$, are obtained by using the implicit function theorem, i.e. by solving the linear system

$$\begin{bmatrix} \frac{\partial G}{\partial y} & \frac{\partial G}{\partial z} \end{bmatrix} \begin{bmatrix} \Phi_{yx}(t_k, t_k) \\ \Phi_{zx}(t_k, t_k) \end{bmatrix} = -\frac{\partial G}{\partial x}. \quad (6.15)$$

The Jacobian matrices are evaluated at $\hat{x}_{k|k}$, $\hat{y}_{k|k}$, and $\hat{z}_{k|k}$. We have used the same notation for the sensitivities as in the description of the time-update in Section 6.2.2.

6.2.2 Time-update

In the time-update, we compute the estimates of the states, the algebraic variables, and the adjoint algebraic variables in between measurements. They satisfy the IVP

$$\hat{x}_k(t_k) = \hat{x}_{k|k}, \quad (6.16a)$$

$$G(\hat{x}_k(t), \hat{y}_k(t), \hat{z}_k(t)) = 0, \quad t \in [t_k, t_{k+1}], \quad (6.16b)$$

$$\frac{d\hat{x}_k(t)}{dt} = F(\hat{y}_k(t), u(t), d(t)), \quad t \in [t_k, t_{k+1}], \quad (6.16c)$$

where $\hat{x}_{k|k}$ is available from the previous measurement-update. The one-step ahead estimates used in the subsequent measurement-update are $\hat{x}_{k+1|k} = \hat{x}_k(t_{k+1})$, $\hat{y}_{k+1|k} = \hat{y}_k(t_{k+1})$, and $\hat{z}_{k+1|k} = \hat{z}_k(t_{k+1})$. The sensitivities of the solutions to (6.16) at time t with respect to the state estimate at time s , $\Phi_{xx}(t, s) = \frac{\partial \hat{x}_k(t)}{\partial \hat{x}_k(s)}$, $\Phi_{yx}(t, s) = \frac{\partial \hat{y}_k(t)}{\partial \hat{x}_k(s)}$, and $\Phi_{zx}(t, s) = \frac{\partial \hat{z}_k(t)}{\partial \hat{x}_k(s)}$, satisfy the IVP

$$\Phi_{xx}(s, s) = \mathbf{I}, \quad (6.17a)$$

$$\frac{\partial G}{\partial x}\Phi_{xx}(t, s) + \frac{\partial G}{\partial y}\Phi_{yx}(t, s) + \frac{\partial G}{\partial z}\Phi_{zx}(t, s) = 0, \quad (6.17b)$$

$$\frac{d\Phi_{xx}(t, s)}{dt} = \frac{\partial F}{\partial y}\Phi_{yx}(t, s). \quad (6.17c)$$

The Jacobian matrices in (6.17b) are evaluated at $\hat{x}_k(t)$, $\hat{y}_k(t)$, and $\hat{z}_k(t)$, and the Jacobian matrix in (6.17c) is evaluated at $\hat{y}_k(t)$, $u(t)$, and $d(t)$. We use the sensitivities to compute the covariance matrix [83]:

$$\begin{aligned} P_k(t) &= \Phi_{xx}(t, t_k)P_{k|k}\Phi_{xx}(t, t_k)' \\ &+ \int_{t_k}^t \Phi_{xx}(t, s)\sigma(\hat{y}_k(s), u(s), d(s))\sigma(\hat{y}_k(s), u(s), d(s))'\Phi_{xx}(t, s)'ds. \end{aligned} \quad (6.18)$$

$P_{k|k}$ is computed in the previous measurement-update. The covariance matrix that is used in the following measurement-update is $P_{k+1|k} = P_k(t_{k+1})$. Next, we briefly describe the numerical solution of the above equations.² Furthermore, we stress that it may be necessary to use multiple time steps in the numerical solution in order to achieve sufficient accuracy if the time interval between the measurements is large.

²We include the description of the numerical solution for the EKF because it does not only involve the solution of IVPs.

6.2.2.1 Numerical solution. We use the simultaneous approach described in Section 5.2.1 (which is based on Euler's implicit method) to solve the IVP (6.16). However, we consider N_k time steps per control interval. Consequently, we use an inexact Newton method to solve

$$\begin{bmatrix} D_{k,n+1}(\hat{x}_{k,n+1}, \hat{x}_{k,n}, \hat{y}_{k,n+1}, u_k, \hat{d}_k) \\ G(\hat{x}_{k,n+1}, \hat{y}_{k,n+1}, \hat{z}_{k,n+1}) \end{bmatrix} = 0, \quad n = 0, \dots, N_k - 1. \quad (6.19)$$

We have assumed the same ZOH parametrization of the manipulated inputs and the disturbance variables as described in Section 5.1. Furthermore, we assume that the boundaries of the control intervals coincide with the measurement times. The discretized differential equations are $D_{k,n+1} = 0$ where

$$\begin{aligned} D_{k,n+1} &= D_{k,n+1}(\hat{x}_{k,n+1}, \hat{x}_{k,n}, \hat{y}_{k,n+1}, u_k, \hat{d}_k) \\ &= \hat{x}_{k,n+1} - F(\hat{y}_{k,n+1}, u_k, \hat{d}_k)\Delta t_{k,n} - \hat{x}_{k,n}, \quad n = 0, \dots, N_k - 1, \end{aligned} \quad (6.20)$$

and $\hat{x}_{k,0} = \hat{x}_{k|k}$. We also use Euler's implicit method for the sensitivity equations (6.17), i.e. we compute the sensitivities by solving

$$\begin{bmatrix} \mathbf{I} & -\frac{\partial F}{\partial y}\Delta t_{k,n} & 0 \\ \frac{\partial G}{\partial x} & \frac{\partial G}{\partial y} & \frac{\partial G}{\partial z} \end{bmatrix} \begin{bmatrix} \Phi_{xx}(t_{k,n+1}, t_{k,n}) \\ \Phi_{yx}(t_{k,n+1}, t_{k,n}) \\ \Phi_{zx}(t_{k,n+1}, t_{k,n}) \end{bmatrix} = \begin{bmatrix} \mathbf{I} \\ 0 \end{bmatrix}, \quad n = 0, \dots, N_k - 1. \quad (6.21)$$

The Jacobian matrix of F is evaluated at $\hat{y}_{k,n+1}$, u_k , and \hat{d}_k , and the Jacobian matrices of G are evaluated at $\hat{x}_{k,n+1}$, $\hat{y}_{k,n+1}$, and $\hat{z}_{k,n+1}$. When we solve the linear system (6.21), we exploit the structure of the system matrix in the same way as in Section 5.2.1.1. Finally, we use a left rectangle rule to approximate the integral in the expression for the covariance matrix in (6.18). This allows us to compute the covariance matrix at each time step in a sequential manner:

$$P_{k,n+1} = \Phi_{xx}(t_{k,n+1}, t_{k,n})\Lambda_{k,n}\Phi_{xx}(t_{k,n+1}, t_{k,n})', \quad n = 0, \dots, N_k - 1, \quad (6.22a)$$

$$\Lambda_{k,n} = P_{k,n} + \sigma(\hat{y}_{k,n}, u_k, \hat{d}_k)\sigma(\hat{y}_{k,n}, u_k, \hat{d}_k)'\Delta t_{k,n}, \quad n = 0, \dots, N_k - 1. \quad (6.22b)$$

The initial covariance matrix is $P_{k,0} = P_{k|k}$.

6.3 The unscented Kalman filter

Both the measurement-update and the time-update in the UKF use unscented transformations [87] in order to compute the state estimates and the corresponding covariance matrices. Unscented transformations are used to approximately propagate a probability density function through a nonlinear function. They propagate a set of deterministic samples (also called sigma points) through the nonlinear function and use the (weighted) mean and covariance of the samples to approximate the mean and covariance of the true distribution. As in the EKF, the initial estimate of the states and the corresponding covariance matrix are

$$\hat{x}_{0|-1} = x_0, \quad (6.23a)$$

$$P_{0|-1} = P_0. \quad (6.23b)$$

6.3.1 Measurement-update

In the measurement-update, we use an unscented transformation to approximate the mean, covariance, and cross-covariance (with the states) of the process outputs in order to compute the filtered states and the corresponding covariance matrix. We compute $2n_x + 1$ samples of the states,

$\{\hat{x}_{k|k-1}^{(i)}\}_{i=0}^{2n_x}$, based on the one-step ahead prediction, $\hat{x}_{k|k-1}$, and the corresponding covariance matrix, $P_{k|k-1}$, from the previous time-update:

$$\hat{x}_{k|k-1}^{(0)} = \hat{x}_{k|k-1}, \quad (6.24a)$$

$$\hat{x}_{k|k-1}^{(i)} = \hat{x}_{k|k-1} + \sqrt{c} \left(\sqrt{P_{k|k-1}} \right)_i, \quad i = 1, \dots, n_x, \quad (6.24b)$$

$$\hat{x}_{k|k-1}^{(i+n_x)} = \hat{x}_{k|k-1} - \sqrt{c} \left(\sqrt{P_{k|k-1}} \right)_i, \quad i = 1, \dots, n_x. \quad (6.24c)$$

n_x is the dimension of the state vector, and $c = \alpha^2(n_x + \kappa)$ where $\alpha \in]0; 1]$. We set κ to zero in this work. We use a Cholesky factorization to compute $\sqrt{P_{k|k-1}}$, and we denote by $\left(\sqrt{P_{k|k-1}} \right)_i$ the i 'th column of $\sqrt{P_{k|k-1}}$ [85]. Furthermore, we introduce two sets of weights [86]:

$$W_m^{(0)} = \frac{\lambda}{n_x + \lambda}, \quad (6.25a)$$

$$W_c^{(0)} = \frac{\lambda}{n_x + \lambda} + (1 - \alpha^2 + \beta), \quad (6.25b)$$

$$W_m^{(i)} = \frac{1}{2(n_x + \lambda)}, \quad i = 1, \dots, 2n_x, \quad (6.25c)$$

$$W_c^{(i)} = \frac{1}{2(n_x + \lambda)}, \quad i = 1, \dots, 2n_x. \quad (6.25d)$$

$\lambda = \alpha^2(n_x + \kappa) - n_x$, and we use $\beta = 2$ in this work. The samples (6.24) and the weights (6.25) are chosen such that the sample mean and covariance approximate the true mean and covariance up to third order [160, Chap. 14].³ We solve the algebraic equations,

$$G(\hat{x}_{k|k-1}^{(i)}, \hat{y}_{k|k-1}^{(i)}, \hat{z}_{k|k-1}^{(i)}) = 0, \quad i = 0, \dots, 2n_x, \quad (6.26)$$

for the samples of the algebraic variables, $\{\hat{y}_{k|k-1}^{(i)}\}_{i=0}^{2n_x}$, and the adjoint algebraic variables, $\{\hat{z}_{k|k-1}^{(i)}\}_{i=0}^{2n_x}$, corresponding to the samples of the states, $\{\hat{x}_{k|k-1}^{(i)}\}_{i=0}^{2n_x}$. Next, we compute the output for each sample:

$$\hat{z}_{k|k-1}^{m,(i)} = H(\hat{y}_{k|k-1}^{(i)}), \quad i = 0, \dots, 2n_x. \quad (6.27)$$

The mean, the covariance, and the cross-covariance of the samples of the process outputs are

$$\hat{z}_{k|k-1}^m = \sum_{i=0}^{2n_x} W_m^{(i)} \hat{z}_{k|k-1}^{m,(i)}, \quad (6.28a)$$

$$T_{k|k-1} = \sum_{i=0}^{2n_x} W_c^{(i)} \left(\hat{z}_{k|k-1}^{m,(i)} - \hat{z}_{k|k-1}^m \right) \left(\hat{z}_{k|k-1}^{m,(i)} - \hat{z}_{k|k-1}^m \right)' + T_k, \quad (6.28b)$$

$$S_{k|k-1} = \sum_{i=0}^{2n_x} W_c^{(i)} \left(\hat{x}_{k|k-1}^{(i)} - \hat{x}_{k|k-1} \right) \left(\hat{z}_{k|k-1}^{m,(i)} - \hat{z}_{k|k-1}^m \right)'. \quad (6.28c)$$

The innovation is

$$e_k = y_k^m - \hat{y}_{k|k-1}^m, \quad (6.29)$$

where y_k^m are the measurements, and the estimate of the measurements is $\hat{y}_{k|k-1}^m = \hat{z}_{k|k-1}^m$ (because the measurement noise is zero-mean and Gaussian). The Kalman filter gain matrix is

$$K_{fx,k} = S_{k|k-1} T_{k|k-1}^{-1}, \quad (6.30)$$

³In comparison, the same approximations in the EKF are accurate up to first order.

and the filtered states and the corresponding covariance matrix are

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_{fx,k} e_k, \quad (6.31a)$$

$$P_{k|k} = P_{k|k-1} - K_{fx,k} T_{k|k-1} K'_{fx,k}. \quad (6.31b)$$

The corresponding estimates of the algebraic variables, $\hat{y}_{k|k}$, and the adjoint algebraic variables, $\hat{z}_{k|k}$, are computed by solving the algebraic equations,

$$G(\hat{x}_{k|k}, \hat{y}_{k|k}, \hat{z}_{k|k}) = 0. \quad (6.32)$$

6.3.2 Time-update

In the time-update, we use an unscented transformation to approximate the mean and covariance of the states at time t_{k+1} . The process noise in (6.1) is non-additive. Therefore, we sample both the states and the process noise. We define $\tilde{n} = n_x + n_\omega$ where n_ω is the dimension of the standard Wiener process $\omega(t)$. Furthermore, we define the sets $\mathcal{N}_0 = \{0\}$, $\mathcal{N}_x = \{1, \dots, 2n_x\}$, and $\mathcal{N}_\omega = \{2n_x + 1, \dots, 2n_x + 2n_\omega\}$. We compute $2\tilde{n} + 1$ samples of the states based on the filtered states, $\hat{x}_{k|k}$, and the corresponding covariance matrix, $P_{k|k}$, from the previous measurement-update:

$$\hat{x}_{k|k}^{(i)} = \hat{x}_{k|k}, \quad i \in \mathcal{N}_0 \cup \mathcal{N}_\omega, \quad (6.33a)$$

$$\hat{x}_{k|k}^{(i)} = \hat{x}_{k|k} + \sqrt{\tilde{c}} \left(\sqrt{P_{k|k}} \right)_i, \quad i = 1, \dots, n_x, \quad (6.33b)$$

$$\hat{x}_{k|k}^{(i+n_x)} = \hat{x}_{k|k} - \sqrt{\tilde{c}} \left(\sqrt{P_{k|k}} \right)_i, \quad i = 1, \dots, n_x. \quad (6.33c)$$

The definition of the parameter \tilde{c} is similar to that of the parameter c in the measurement-update, i.e. $\tilde{c} = \alpha^2(\tilde{n} + \kappa)$. We repeat that we use $\kappa = 0$ in this work. As in the measurement-update, we compute $\sqrt{P_{k|k}}$ using a Cholesky factorization, and we define two sets of weights analogous to the weights in (6.25):

$$\tilde{W}_m^{(0)} = \frac{\tilde{\lambda}}{\tilde{n} + \tilde{\lambda}}, \quad (6.34a)$$

$$\tilde{W}_c^{(0)} = \frac{\tilde{\lambda}}{\tilde{n} + \tilde{\lambda}} + (1 - \alpha^2 + \beta), \quad (6.34b)$$

$$\tilde{W}_m^{(i)} = \frac{1}{2(\tilde{n} + \tilde{\lambda})}, \quad i = 1, \dots, 2\tilde{n}, \quad (6.34c)$$

$$\tilde{W}_c^{(i)} = \frac{1}{2(\tilde{n} + \tilde{\lambda})}, \quad i = 1, \dots, 2\tilde{n}. \quad (6.34d)$$

The definition of $\tilde{\lambda}$ is similar to the definition of λ in the measurement-update: $\tilde{\lambda} = \alpha^2(\tilde{n} + \kappa) - \tilde{n}$. The estimates of the states, the algebraic variables, and the adjoint algebraic variables in between measurements satisfy the deterministic IVPs

$$\hat{x}_k^{(i)}(t_k) = \hat{x}_{k|k}^{(i)}, \quad i \in \mathcal{N}_0 \cup \mathcal{N}_x, \quad (6.35a)$$

$$G(\hat{x}_k^{(i)}(t), \hat{y}_k^{(i)}(t), \hat{z}_k^{(i)}(t)) = 0, \quad i \in \mathcal{N}_0 \cup \mathcal{N}_x, \quad t \in [t_k, t_{k+1}], \quad (6.35b)$$

$$d\hat{x}_k^{(i)}(t) = F(\hat{y}_k^{(i)}(t), u(t), d(t))dt, \quad i \in \mathcal{N}_0 \cup \mathcal{N}_x, \quad t \in [t_k, t_{k+1}], \quad (6.35c)$$

and

$$\hat{x}_k^{(i)}(t_k) = \hat{x}_{k|k}^{(i)}, \quad i \in \mathcal{N}_\omega, \quad (6.36a)$$

$$G(\hat{x}_k^{(i)}(t), \hat{y}_k^{(i)}(t), \hat{z}_k^{(i)}(t)) = 0, \quad i \in \mathcal{N}_\omega, \quad t \in [t_k, t_{k+1}], \quad (6.36b)$$

$$d\hat{x}_k^{(i)}(t) = F(\hat{y}_k^{(i)}(t), u(t), d(t))dt + \sigma(\hat{y}_k^{(i)}(t), u(t), d(t))d\omega^{(i)}(t), \quad i \in \mathcal{N}_\omega, \quad t \in [t_k, t_{k+1}]. \quad (6.36c)$$

The increments are sampled deterministically:

$$d\omega^{(i+2n_x)}(t) = \left(\sqrt{\tilde{c} dt}\right) e_i, \quad i = 1, \dots, n_\omega, \quad (6.37a)$$

$$d\omega^{(i+2n_x+n_\omega)}(t) = -\left(\sqrt{\tilde{c} dt}\right) e_i, \quad i = 1, \dots, n_\omega. \quad (6.37b)$$

All elements of the vector e_i are zero except for the i 'th element which is one. The IVP (6.36) is deterministic because the increments, $d\omega^{(i)}(t)$, are deterministic. We solve the IVP (6.35) with the simultaneous approach described in Section 5.2.1, and we use the semi-implicit discretization scheme described in Section 5.3 for the IVP (6.36). As we mentioned in the description of the EKF, it may be necessary to use multiple time steps in the numerical solution of the IVPs in order to achieve sufficiently accurate numerical solutions. The one-step ahead prediction of the states and the corresponding covariance matrix are the (weighted) sample mean and covariance, i.e.

$$\hat{x}_{k+1|k} = \sum_{i=0}^{2\tilde{n}} \tilde{W}_m^{(i)} \hat{x}_{k+1|k}^{(i)}, \quad (6.38a)$$

$$P_{k+1|k} = \sum_{i=0}^{2\tilde{n}} \tilde{W}_c^{(i)} \left(\hat{x}_{k+1|k}^{(i)} - \hat{x}_{k+1|k} \right) \left(\hat{x}_{k+1|k}^{(i)} - \hat{x}_{k+1|k} \right)', \quad (6.38b)$$

where $\hat{x}_{k+1|k}^{(i)} = \hat{x}_k^{(i)}(t_{k+1})$.

6.4 The particle filter

As in the UKF, the result of propagating a probability density function through a nonlinear function is approximated using particles (or samples) in the PF. However, the PF uses a set of random particles.⁴ In the measurement-update, the particles are resampled and the sample mean and covariance constitute the filtered states and the corresponding covariance matrix. In the time-update, the particles are simply propagated through the stochastic DAEs (6.1). We sample N_p particles, $\{\hat{x}_{0|-1}^{(i)}\}_{i=1}^{N_p}$, from the distribution of the states at the initial time, $N(x_0, P_0)$. Next, we solve the algebraic equations,

$$G(\hat{x}_{0|-1}^{(i)}, \hat{y}_{0|-1}^{(i)}, \hat{z}_{0|-1}^{(i)}) = 0, \quad i = 1, \dots, N_p, \quad (6.39)$$

for the algebraic and adjoint algebraic variables, $\{\hat{y}_{0|-1}^{(i)}\}_{i=1}^{N_p}$ and $\{\hat{z}_{0|-1}^{(i)}\}_{i=1}^{N_p}$, corresponding to the states, $\{\hat{x}_{0|-1}^{(i)}\}_{i=1}^{N_p}$.

6.4.1 Measurement-update

In the measurement-update, we resample the particles based on the likelihood of observing the actual measurements assuming that the predicted process outputs are true. The filtered states and the corresponding covariance matrix are the mean and covariance of the resampled particles. For each particle, we compute the process output:

$$\hat{z}_{k|k-1}^{m,(i)} = H(\hat{y}_{k|k-1}^{(i)}), \quad i = 1, \dots, N_p. \quad (6.40)$$

The one-step ahead predictions of the algebraic variables, $\{\hat{y}_{k|k-1}^{(i)}\}_{i=1}^{N_p}$, are available from the previous time-update. Next, we compute the difference between the measurements, y_k^m , and the process outputs:

$$e_k^{(i)} = y_k^m - \hat{z}_{k|k-1}^{m,(i)}, \quad i = 1, \dots, N_p. \quad (6.41)$$

⁴The number of particles should be chosen according to the required accuracy and the available computational resources. The accuracy of the PF is expected to increase with the number of particles. However, the computational demands increase as well.

The relative likelihood that y_k^m is observed, assuming that the process output $\hat{z}_{k|k-1}^{m,(i)}$ is true, is

$$\tilde{q}^{(i)} = \frac{1}{\sqrt{(2\pi)^{n_m} |T_k|}} \exp\left(-\frac{1}{2} \left(e_k^{(i)}\right)' T_k^{-1} e_k^{(i)}\right), \quad i = 1, \dots, N_p, \quad (6.42)$$

where n_m is the dimension of the process output vector, and $|T_k|$ is the determinant of the measurement noise covariance matrix, T_k . The expression in (6.42) is the probability density function for a normal distribution (because the measurement noise is normally distributed). The normalized relative likelihoods are

$$q^{(i)} = \frac{\tilde{q}^{(i)}}{\sum_{j=1}^{N_p} \tilde{q}^{(j)}}, \quad i = 1, \dots, N_p. \quad (6.43)$$

We resample the particles using systematic resampling [49, 73]. Consequently, we compute

$$p^{(i)} = ((i-1) + \tilde{p})/N_p, \quad i = 1, \dots, N_p, \quad (6.44)$$

where \tilde{p} is a uniformly distributed scalar, i.e. $\tilde{p} \sim U([0, 1])$. Next, we compute $m^{(i)}$ which is the number of indices, l , for which $p^{(l)}$ is contained in the interval $]s^{(i-1)}, s^{(i)}]$ where $s^{(i)} = \sum_{j=1}^i q^{(j)}$. The resampled particles, $\{\hat{x}_{k|k}^{(i)}\}_{i=1}^{N_p}$, contain $m^{(i)}$ instances (or copies) of $\hat{x}_{k|k-1}^{(i)}$. Based on the resampled particles, we compute the filtered states and the corresponding covariance matrix:

$$\hat{x}_{k|k} = W_m \sum_{i=1}^{N_p} \hat{x}_{k|k}^{(i)}, \quad (6.45a)$$

$$P_{k|k} = W_c \sum_{i=1}^{N_p} \left(\hat{x}_{k|k}^{(i)} - \hat{x}_{k|k}\right) \left(\hat{x}_{k|k}^{(i)} - \hat{x}_{k|k}\right)'. \quad (6.45b)$$

The weights are $W_m = 1/N_p$ and $W_c = 1/(N_p - 1)$. We solve the algebraic equations,

$$G(\hat{x}_{k|k}, \hat{y}_{k|k}, \hat{z}_{k|k}) = 0, \quad (6.46)$$

for the estimates of the algebraic variables, $\hat{y}_{k|k}$, and the adjoint algebraic variables, $\hat{z}_{k|k}$.

6.4.2 Time-update

In the time-update, we propagate the particles through the stochastic DAEs (6.1), i.e. we use the numerical method described in Section 5.3 to approximate the solutions to the stochastic IVPs

$$\hat{\mathbf{x}}_k^{(i)}(t_k) = \hat{\mathbf{x}}_{k|k}^{(i)}, \quad i = 1, \dots, N_p, \quad (6.47a)$$

$$G(\hat{\mathbf{x}}_k^{(i)}(t), \hat{\mathbf{y}}_k^{(i)}(t), \hat{\mathbf{z}}_k^{(i)}(t)) = 0, \quad i = 1, \dots, N_p, \quad t \in [t_k, t_{k+1}], \quad (6.47b)$$

$$d\hat{\mathbf{x}}_k^{(i)}(t) = F(\hat{\mathbf{y}}_k^{(i)}(t), u(t), d(t))dt + \sigma(\hat{\mathbf{y}}_k^{(i)}(t), u(t), d(t))d\omega(t), \quad i = 1, \dots, N_p, \quad t \in [t_k, t_{k+1}]. \quad (6.47c)$$

Note that the initial condition (6.47a) specifies the state estimates at time t_k exactly, whereas the initial condition considered in Section 5.1 specifies the distribution of the initial states. The resampled particles, $\{\hat{x}_{k|k}^{(i)}\}_{i=1}^{N_p}$, are available from the previous measurement-update. The numerical solution of the stochastic IVPs (6.47) provides the one-step ahead predictions, $\{\hat{x}_{k+1|k}^{(i)}\}_{i=1}^{N_p}$, $\{\hat{y}_{k+1|k}^{(i)}\}_{i=1}^{N_p}$, and $\{\hat{z}_{k+1|k}^{(i)}\}_{i=1}^{N_p}$, which are used in the subsequent measurement-update. As mentioned previously, it may be necessary to use multiple time steps in the numerical solution in order to obtain sufficiently accurate approximations to the solutions of the IVPs.

6.5 The ensemble Kalman filter

As mentioned previously, the EnKF is a type of PF, i.e. it involves a set of particles. However, the measurement-update does not involve resampling of the particles as in the PF described in Section 6.4. Instead, it involves an update based on the Kalman filter gain matrix as in the EKF and the UKF described in Section 6.2 and Section 6.3. We sample N_p particles, $\{\hat{x}_{0|-1}^{(i)}\}_{i=1}^{N_p}$, from $N(x_0, P_0)$, i.e. from the distribution of the initial states. Furthermore, we compute the algebraic variables, $\{\hat{y}_{0|-1}^{(i)}\}_{i=1}^{N_p}$, and the adjoint algebraic variables, $\{\hat{z}_{0|-1}^{(i)}\}_{i=1}^{N_p}$, by solving the algebraic equations,

$$G(\hat{x}_{0|-1}^{(i)}, \hat{y}_{0|-1}^{(i)}, \hat{z}_{0|-1}^{(i)}) = 0, \quad i = 1, \dots, N_p. \quad (6.48)$$

6.5.1 Measurement-update

In the measurement-update, we compute means, covariances, and cross-covariances of the samples in order to compute the Kalman filter gain matrix which we use to update the state estimate for each particle. Finally, the filtered states and the corresponding covariance matrix are given by the mean and covariance of the updated state estimates. First, we compute the process outputs:

$$\hat{z}_{k|k-1}^{m,(i)} = H(\hat{y}_{k|k-1}^{(i)}), \quad i = 1, \dots, N_p. \quad (6.49)$$

The one-step ahead predictions of the algebraic variables, $\{\hat{y}_{k|k-1}^{(i)}\}_{i=1}^{N_p}$, are available from the previous time-update. Next, we compute the means of the states and the process outputs as well as the covariance and cross-covariance (with respect to the states) of the process outputs:

$$\hat{x}_{k|k-1} = W_m \sum_{i=1}^{N_p} \hat{x}_{k|k-1}^{(i)}, \quad (6.50a)$$

$$\hat{z}_{k|k-1}^m = W_m \sum_{i=1}^{N_p} \hat{z}_{k|k-1}^{m,(i)}, \quad (6.50b)$$

$$T_{k|k-1} = W_c \sum_{i=1}^{N_p} (\hat{z}_{k|k-1}^{m,(i)} - \hat{z}_{k|k-1}^m)(\hat{z}_{k|k-1}^{m,(i)} - \hat{z}_{k|k-1}^m)' + T_k, \quad (6.50c)$$

$$S_{k|k-1} = W_c \sum_{i=1}^{N_p} (\hat{x}_{k|k-1}^{(i)} - \hat{x}_{k|k-1})(\hat{z}_{k|k-1}^{m,(i)} - \hat{z}_{k|k-1}^m)'. \quad (6.50d)$$

The weights are $W_m = 1/N_p$ and $W_c = 1/(N_p - 1)$, and the one-step ahead predictions of the states, $\{\hat{x}_{k|k-1}^{(i)}\}_{i=1}^{N_p}$, are computed in the previous time-update. For each particle, we sample a measurement:

$$\hat{y}_{k|k-1}^{m,(i)} = \hat{z}_{k|k-1}^{m,(i)} + v_k^{(i)}, \quad i = 1, \dots, N_p. \quad (6.51)$$

The sampled measurement noise, $v_k^{(i)}$, is drawn from the normal distribution $N(0, T_k)$. We compute the innovation for each particle:

$$e_k^{(i)} = y_k^m - \hat{y}_{k|k-1}^{m,(i)}, \quad i = 1, \dots, N_p. \quad (6.52)$$

y_k^m are the measurements, and the Kalman filter gain matrix is given by

$$K_{fx,k} = S_{k|k-1} T_{k|k-1}^{-1}. \quad (6.53)$$

We use the Kalman filter gain matrix to update the state estimate for each particle:

$$\hat{x}_{k|k}^{(i)} = \hat{x}_{k|k-1}^{(i)} + K_{fx,k} e_k^{(i)}, \quad i = 1, \dots, N_p. \quad (6.54)$$

Finally, the filtered states and the corresponding covariance matrix are

$$\hat{\mathbf{x}}_{k|k} = W_m \sum_{i=1}^{N_p} \hat{\mathbf{x}}_{k|k}^{(i)}, \quad (6.55a)$$

$$P_{k|k} = W_c \sum_{i=1}^{N_p} (\hat{\mathbf{x}}_{k|k}^{(i)} - \hat{\mathbf{x}}_{k|k})(\hat{\mathbf{x}}_{k|k}^{(i)} - \hat{\mathbf{x}}_{k|k})', \quad (6.55b)$$

and the estimates of the algebraic variables, $\hat{\mathbf{y}}_{k|k}$, and the adjoint algebraic variables, $\hat{\mathbf{z}}_{k|k}$, are obtained by solving the algebraic equations,

$$G(\hat{\mathbf{x}}_{k|k}, \hat{\mathbf{y}}_{k|k}, \hat{\mathbf{z}}_{k|k}) = 0. \quad (6.56)$$

6.5.2 Time-update

The time-update in the EnKF is identical to the time-update in the PF, i.e. the particles are propagated through the stochastic DAEs (6.1). As in the PF, we use the numerical method described in Section 5.3 to approximate the solutions to the IVPs

$$\hat{\mathbf{x}}_k^{(i)}(t_k) = \hat{\mathbf{x}}_{k|k}^{(i)}, \quad i = 1, \dots, N_p, \quad (6.57a)$$

$$G(\hat{\mathbf{x}}_k^{(i)}(t), \hat{\mathbf{y}}_k^{(i)}(t), \hat{\mathbf{z}}_k^{(i)}(t)) = 0, \quad i = 1, \dots, N_p, \quad t \in [t_k, t_{k+1}], \quad (6.57b)$$

$$d\hat{\mathbf{x}}_k^{(i)}(t) = F(\hat{\mathbf{y}}_k^{(i)}(t), u(t), d(t))dt + \sigma(\hat{\mathbf{y}}_k^{(i)}(t), u(t), d(t))d\boldsymbol{\omega}(t), \quad i = 1, \dots, N_p, \quad t \in [t_k, t_{k+1}], \quad (6.57c)$$

which are identical to the IVPs (6.47). The numerical solution provides the one-step ahead predictions for each particle, $\{\hat{\mathbf{x}}_{k+1|k}^{(i)}\}_{i=1}^{N_p}$, $\{\hat{\mathbf{y}}_{k+1|k}^{(i)}\}_{i=1}^{N_p}$, and $\{\hat{\mathbf{z}}_{k+1|k}^{(i)}\}_{i=1}^{N_p}$, which are used in the following measurement-update. As mentioned in the descriptions of the other filters, it may be necessary to use multiple time steps in the numerical solution in order to obtain sufficient accuracy.

6.6 Summary

In this chapter, we have described the EKF, the UKF, the PF, and the EnKF for state estimation of continuous-discrete systems that involve stochastic semi-explicit DAEs in the form (1.3). All four algorithms consist of 1) a measurement-update that incorporates information from the measurements and 2) a time-update that predicts the dynamical behavior of the system in between measurements. The EKF is based on a linearization of the nonlinear model equations while the UKF, PF, and EnKF involve samples (or particles). The UKF uses deterministic particles, whereas the PF and the EnKF use random particles. The state estimation algorithms are relevant to the NMPC algorithm that we present in Chapter 8, and in Chapter 9, we compare the four algorithms on accuracy and computational performance using the flash separation process described in Chapter 3.

Dynamic optimization

In this chapter, we present an algorithm for gradient-based numerical solution of dynamic optimization problems in the form (1.5). The algorithm uses a single-shooting approach in which the infinite-dimensional dynamic optimization problem is transcribed into a finite-dimensional NLP. We solve the NLP using off-the-shelf optimization software. The evaluation of the objective function in the NLP requires the solution of a deterministic IVP. We formulate the dynamic optimization algorithm using both a simultaneous approach as described in Section 5.2.1 and a nested approach as described in Section 5.2.2 for the numerical solution of the involved IVPs.¹ Efficient algorithms for the numerical solution of NLPs require the gradient of the objective function. The dynamic optimization algorithm that we present uses a discrete adjoint method to compute this gradient. We describe the adjoint method corresponding to both the simultaneous and the nested approach.

We describe the dynamic optimization problem in Section 7.1, and we present the dynamic optimization algorithm in Section 7.2. We briefly summarize this chapter in Section 7.3. Dynamic optimization is central to the NMPC algorithm that we describe in Chapter 8, and in Chapter 9, we test the computational performance of the dynamic optimization algorithm when using the simultaneous and the nested approach. The performance test involves the flash separation process described in Chapter 3. In Chapter 10, we use the dynamic optimization algorithm to optimize the thermal and isothermal waterflooding processes described in Chapter 4, and we discuss the computational performance of the algorithm for these examples. The content of this chapter is based on content from the papers in Appendix C, E, G, I, and N.

7.1 Problem formulation

Dynamic optimization problems are constrained optimization problems in which the constraints contain differential (or difference) equations. The differential equations constitute a model (or part of a model) of a dynamical process. The purpose of dynamic optimization is to compute an open-loop control strategy which optimizes a performance measure of the process over a given horizon.² The performance measure can represent 1) the distance to predefined setpoints (to be minimized) or 2) the economics of the dynamical process (to be optimized). We consider dynamic optimization problems in the form

$$\min_{[x(t);y(t);z(t)]_{t_0}^{t_f}, \{u_k\}_{k=0}^{N-1}} \phi = \phi \left([y(t); u(t); d(t)]_{t_0}^{t_f} \right) = \int_{t_0}^{t_f} \Phi(y(t), u(t), d(t)) dt, \quad (7.1a)$$

¹Dynamic optimization algorithms can also be described as nested or simultaneous. However, we stress that when we refer to simultaneous and nested approaches, we refer to the approaches described in Section 5.2 for solving IVPs that involve DAEs. In this work, we only use the single-shooting method (sometimes referred to as a nested approach). Therefore, there should be little risk of confusion.

²Open-loop control strategies do not involve feedback from the process (in contrast to closed-loop control strategies).

subject to

$$x(t_0) = \hat{x}_0, \quad (7.1b)$$

$$G(x(t), y(t), z(t)) = 0, \quad t \in [t_0, t_f], \quad (7.1c)$$

$$\dot{x}(t) = F(y(t), u(t), d(t)), \quad t \in [t_0, t_f], \quad (7.1d)$$

$$u(t) = u_k, \quad t \in [t_k, t_{k+1}[, \quad k = 0, \dots, N-1, \quad (7.1e)$$

$$d(t) = \hat{d}_k, \quad t \in [t_k, t_{k+1}[, \quad k = 0, \dots, N-1, \quad (7.1f)$$

$$\{u_k\}_{k=0}^{N-1} \in \mathcal{U}. \quad (7.1g)$$

The vector $[x(t); y(t); z(t)]_{t_0}^{t_f}$ contains the states, $x(t)$, the algebraic variables, $y(t)$, and the adjoint algebraic variables, $z(t)$, which are dependent decision variables. t_0 and $t_N = t_f$ are the initial and the final time. N is the number of control intervals. The manipulated inputs, $\{u_k\}_{k=0}^{N-1}$, are independent decision variables. As mentioned above, the objective function, ϕ , in (7.1a) represents a performance measure of the process over the horizon $[t_0, t_f]$. The estimated initial state, \hat{x}_0 , in the initial condition (7.1b) is a parameter in the optimization problem. The semi-explicit DAEs (7.1c)-(7.1d) constitute the model of the dynamical process. (7.1e)-(7.1f) are ZOH parametrizations of the manipulated inputs and the disturbance variables. The predicted disturbance variables, $\{\hat{d}_k\}_{k=0}^{N-1}$, in (7.1f) are also parameters in the problem. Finally, (7.1g) represents constraints on the manipulated inputs.

7.2 The dynamic optimization algorithm

In this section, we present the gradient-based algorithm for numerical solution of the dynamic optimization problem (7.1). In Section 7.2.1, we use the single-shooting method to transcribe the dynamic optimization problem into an NLP. In Section 7.2.2 and Section 7.2.3, we describe the evaluation of the objective function in the NLP using the simultaneous approach from Section 5.2.1 and the nested approach from Section 5.2.2, respectively, for the numerical solution of the involved IVPs. In the same sections, we describe the corresponding discrete adjoint methods for computing the gradients of the objective function.

7.2.1 Single-shooting

Fig. 7.1 illustrates the main principle in the single-shooting approach: the manipulated inputs, $u(t)$, are parametrized, and the continuous states, $x(t)$, algebraic variables, $y(t)$, and adjoint algebraic variables, $z(t)$, are considered implicit functions of the parametrized manipulated inputs determined by the IVP in the dynamic optimization problem. The solution to the IVP is used for evaluating the objective function. Using this principle, the single-shooting method transcribes the dynamic optimization problem (7.1) into an NLP:

$$\min_{\{u_k\}_{k=0}^{N-1}} \psi = \psi \left(\{u_k\}_{k=0}^{N-1}; \hat{x}_0, \{\hat{d}_k\}_{k=0}^{N-1} \right), \quad (7.2a)$$

$$\text{subject to } \{u_k\}_{k=0}^{N-1} \in \mathcal{U}. \quad (7.2b)$$

For given $\{u_k\}_{k=0}^{N-1}$, \hat{x}_0 , and $\{\hat{d}_k\}_{k=0}^{N-1}$, the objective function ψ in (7.2a) is defined as the objective function ϕ in (7.1a) evaluated using the solution to the IVP (7.1b)-(7.1d) subject to the ZOH parametrizations of the manipulated inputs and the disturbance variables (7.1e)-(7.1f):

$$\psi = \psi \left(\{u_k\}_{k=0}^{N-1}; \hat{x}_0, \{\hat{d}_k\}_{k=0}^{N-1} \right) = \{\phi : (7.1b)-(7.1f)\}. \quad (7.3)$$

Efficient gradient-based numerical optimization algorithms for the solution of the NLP (7.2) require the evaluation of the objective function, ψ , as well as the gradients, $\{\nabla_{u_k} \psi\}_{k=0}^{N-1}$. In order to

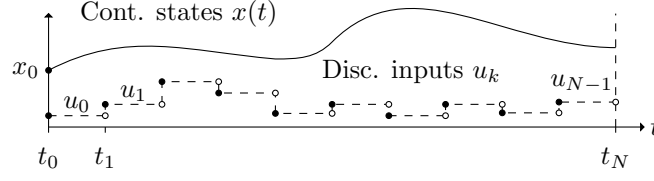


Fig. 7.1: A sketch of the continuous states, $x(t)$, which are implicit functions of the parametrized manipulated inputs, $\{u_k\}_{k=0}^{N-1}$, in the single-shooting approach. This figure originally appeared in [139].

evaluate ψ , we use the simultaneous approach from Section 5.2.1 and the nested approach from Section 5.2.2 for numerical solution of the IVP (7.1b)-(7.1d) subject to (7.1e)-(7.1f). In order to evaluate the gradients, $\{\nabla_{u_k} \psi\}_{k=0}^{N-1}$, we use a discrete adjoint method.

7.2.2 The simultaneous approach

In the simultaneous approach described in Section 5.2.1, we use Euler's implicit method to solve the IVP (7.1b)-(7.1d) subject to the ZOH parametrizations (7.1e)-(7.1f). Consequently, we solve a set of residual equations, $R_{k+1} = 0$, for $w_{k+1} = [x_{k+1}; y_{k+1}; z_{k+1}] \approx [x(t_{k+1}); y(t_{k+1}); z(t_{k+1})]$ in each time step. We substitute these residual equations into the objective function ψ in (7.3):

$$\psi = \psi \left(\{u_k\}_{k=0}^{N-1}; \hat{x}_0, \{\hat{d}_k\}_{k=0}^{N-1} \right) \quad (7.4a)$$

$$= \left\{ \phi = \sum_{k=0}^{N-1} \Phi_k(y_{k+1}, u_k, \hat{d}_k) : \right. \quad (7.4b)$$

$$x_0 = \hat{x}_0, \quad (7.4c)$$

$$R_{k+1}(w_{k+1}; x_k, u_k, \hat{d}_k) = 0, \quad k = 0, \dots, N-1, \quad (7.4d)$$

$$\left. [x_{k+1}; y_{k+1}; z_{k+1}] = w_{k+1}, \quad k = 0, \dots, N-1 \right\}. \quad (7.4e)$$

We have used a right rectangle rule to approximate the integral in the expression for the objective function ϕ in (7.1a). The result of the approximation is the sum in (7.4b) where

$$\Phi_k = \Phi_k(y_{k+1}, u_k, \hat{d}_k) = \Phi(y_{k+1}, u_k, \hat{d}_k) \Delta t_k, \quad (7.5)$$

approximates the integral from t_k to t_{k+1} . We use a right rectangle rule because it is consistent with Euler's implicit method. We repeat the residual equations (5.4) here:

$$\begin{aligned} R_{k+1} &= R_{k+1}(w_{k+1}) = R_{k+1}(w_{k+1}; x_k, u_k, \hat{d}_k) \\ &= R_{k+1}(x_{k+1}, y_{k+1}, z_{k+1}; x_k, u_k, \hat{d}_k) \\ &= \begin{bmatrix} D_{k+1}(x_{k+1}, x_k, y_{k+1}, u_k, \hat{d}_k) \\ G(x_{k+1}, y_{k+1}, z_{k+1}) \end{bmatrix} = 0, \quad k = 0, \dots, N-1. \end{aligned} \quad (7.6)$$

We also repeat the expression (5.5) for D_{k+1} :

$$\begin{aligned} D_{k+1} &= D_{k+1}(x_{k+1}, x_k, y_{k+1}, u_k, \hat{d}_k) \\ &= x_{k+1} - F(y_{k+1}, u_k, \hat{d}_k) \Delta t_k - x_k, \quad k = 0, \dots, N-1. \end{aligned} \quad (7.7)$$

In the adjoint method, we compute the adjoints, $\{\lambda_k\}_{k=1}^N$, sequentially in a backwards manner in order to compute the gradients of ψ in (7.4) with respect to the manipulated inputs, $\{\nabla_{u_k} \psi\}_{k=0}^{N-1}$. For clarity of the presentation, we first present the adjoint equations, and thereafter, we present

the expressions for the involved gradients and Jacobian matrices. First, we compute λ_N by solving the linear system

$$\left(\frac{\partial R_N}{\partial w_N}\right)' \lambda_N = -\nabla_{w_N} \Phi_{N-1}, \quad (7.8)$$

and we compute the corresponding gradient, $\nabla_{u_{N-1}} \psi$, by

$$\nabla_{u_k} \psi = \nabla_{u_k} \Phi_k + \left(\frac{\partial R_{k+1}}{\partial u_k}\right)' \lambda_{k+1}, \quad k = 0, \dots, N-1. \quad (7.9)$$

We compute the subsequent adjoints by solving the linear systems

$$\left(\frac{\partial R_k}{\partial w_k}\right)' \lambda_k = -\left(\frac{\partial R_{k+1}}{\partial w_k}\right)' \lambda_{k+1} - \nabla_{w_k} \Phi_{k-1}, \quad k = N-1, \dots, 1, \quad (7.10)$$

starting from $k = N-1$ and marching backwards. For each of the subsequent adjoints, we compute the corresponding gradient using (7.9). The adjoint equations must be solved in a backwards manner because 1) λ_N can be computed by solving (7.8) and 2) it is necessary to compute λ_{k+1} before solving (7.10) for λ_k .

The gradients of Φ_k with respect to w_{k+1} and the manipulated inputs, u_k , are

$$\nabla_{w_{k+1}} \Phi_k = \left[0; \quad \nabla_y \Phi(y_{k+1}, u_k, \hat{d}_k) \Delta t_k; \quad 0 \right], \quad k = 0, \dots, N-1, \quad (7.11a)$$

$$\nabla_{u_k} \Phi_k = \nabla_u \Phi(y_{k+1}, u_k, \hat{d}_k) \Delta t_k, \quad k = 0, \dots, N-1. \quad (7.11b)$$

The Jacobian matrices of the residual function R_{k+1} with respect to w_{k+1} , the manipulated inputs, u_k , and w_k are

$$\frac{\partial R_{k+1}}{\partial w_{k+1}} = \begin{bmatrix} \mathbf{I} & -\frac{\partial F}{\partial y} \Delta t_k & 0 \\ \frac{\partial G}{\partial x} & \frac{\partial G}{\partial y} & \frac{\partial G}{\partial z} \end{bmatrix}, \quad k = 0, \dots, N-1, \quad (7.12a)$$

$$\frac{\partial R_{k+1}}{\partial u_k} = \begin{bmatrix} -\frac{\partial F}{\partial u} \Delta t_k \\ 0 \end{bmatrix}, \quad k = 0, \dots, N-1, \quad (7.12b)$$

$$\frac{\partial R_{k+1}}{\partial w_k} = \begin{bmatrix} -\mathbf{I} & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad k = 1, \dots, N-1. \quad (7.12c)$$

To summarize, the objective function ψ in (7.4) is evaluated by 1) sequentially solving the residual equations in (7.4d) in a forward manner starting from the initial condition in (7.4c) and 2) evaluating the sum in (7.4b) using the solution to the residual equations. The gradients of ψ are computed sequentially in a backwards manner (starting from $k = N-1$) by 1) solving for the adjoints, λ_{k+1} , and 2) using the adjoints to compute the corresponding gradient, $\nabla_{u_k} \psi$. We stress that it is necessary to solve the residual equations in (7.4d) before computing the gradients using (7.8)-(7.10) in order to be able to evaluate the involved gradients and Jacobian matrices.

7.2.3 The nested approach

In the nested approach described in Section 5.2.2, the algebraic and adjoint algebraic variables are considered implicit functions of the states. The differential equations are solved with Euler's implicit method, i.e. in each time step, the discretized differential equations, $D_{k+1} = 0$, are solved for the states, $x_{k+1} \approx x(t_{k+1})$. In order to evaluate the right-hand side function in the differential equations, it is necessary to solve the algebraic equations, $G(x_{k+1}, y_{k+1}, z_{k+1}) = 0$, for the algebraic variables, $y_{k+1} = y(x_{k+1}) \approx y(x(t_{k+1}))$, and the adjoint algebraic variables, $z_{k+1} = z(x_{k+1}) \approx z(x(t_{k+1}))$.

We substitute the discretized differential equations and the algebraic equations into the objective function ψ in (7.3):

$$\psi = \psi(\{u_k\}_{k=0}^{N-1}; \hat{x}_0, \{\hat{d}_k\}_{k=0}^{N-1}) \quad (7.13a)$$

$$= \left\{ \phi = \sum_{k=0}^{N-1} \Phi_k(y_{k+1}, u_k, \hat{d}_k) : \right. \quad (7.13b)$$

$$x_0 = \hat{x}_0, \quad (7.13c)$$

$$D_{k+1}(x_{k+1}, x_k, y_{k+1}, u_k, \hat{d}_k) = 0, \quad k = 0, \dots, N-1, \quad (7.13d)$$

$$\left. G(x_{k+1}, y_{k+1}, z_{k+1}) = 0, \quad k = 0, \dots, N-1 \right\}. \quad (7.13e)$$

As in Section 7.2.2, we have approximated the integral in the expression for ϕ in (7.1a) using a left rectangle rule, and $\Phi_k = \Phi_k(y_{k+1}, u_k, \hat{d}_k)$ in (7.13b) is given by (7.5). The sum in (7.13b) is the approximation of the integral. We repeat the expression for the discretized differential equations (5.14):

$$\begin{aligned} D_{k+1} &= D_{k+1}(x_{k+1}) = D_{k+1}(x_{k+1}, x_k, y_{k+1}, u_k, \hat{d}_k) \\ &= x_{k+1} - F(y_{k+1}, u_k, \hat{d}_k)\Delta t_k - x_k = 0, \quad k = 0, \dots, N-1. \end{aligned} \quad (7.14)$$

Again, we first present the adjoint equations, and thereafter, we present the expressions for the involved gradients and Jacobian matrices. The adjoint equations corresponding to the nested approach are analogous to the adjoint equations presented in Section 7.2.2. First, we solve the linear system

$$\left(\frac{\partial D_N}{\partial x_N} \right)' \lambda_N = -\nabla_{x_N} \Phi_{N-1}, \quad (7.15)$$

for λ_N , and we compute the corresponding gradient, $\nabla_{u_{N-1}} \psi$, using

$$\nabla_{u_k} \psi = \nabla_{u_k} \Phi_k + \left(\frac{\partial D_{k+1}}{\partial u_k} \right)' \lambda_{k+1}, \quad k = 0, \dots, N-1. \quad (7.16)$$

The subsequent adjoints are computed by solving

$$\left(\frac{\partial D_k}{\partial x_k} \right)' \lambda_k = - \left(\frac{\partial D_{k+1}}{\partial x_k} \right)' \lambda_{k+1} - \nabla_{x_k} \Phi_{k-1}, \quad k = N-1, \dots, 1, \quad (7.17)$$

for λ_k marching backwards in a sequential manner (starting with $k = N-1$).

The gradients of Φ_k with respect to the states, x_{k+1} , and the manipulated inputs, u_k , are

$$\nabla_{x_{k+1}} \Phi_k = \left(\frac{\partial y_{k+1}}{\partial x_{k+1}} \right)' \nabla_y \Phi(y_{k+1}, u_k, \hat{d}_k) \Delta t_k, \quad k = 0, \dots, N-1, \quad (7.18a)$$

$$\nabla_{u_k} \Phi_k = \nabla_u \Phi(y_{k+1}, u_k, \hat{d}_k) \Delta t_k, \quad k = 0, \dots, N-1. \quad (7.18b)$$

The Jacobian matrices of D_{k+1} with respect to the states in the current time step, x_{k+1} , the manipulated inputs, u_k , and the states in the previous time step, x_k , are

$$\frac{\partial D_{k+1}}{\partial x_{k+1}} = \mathbf{I} - \left(\frac{\partial F}{\partial y} \Delta t_k \right) \frac{\partial y_{k+1}}{\partial x_{k+1}}, \quad k = 0, \dots, N-1, \quad (7.19a)$$

$$\frac{\partial D_{k+1}}{\partial u_k} = - \frac{\partial F}{\partial u} \Delta t_k, \quad k = 0, \dots, N-1, \quad (7.19b)$$

$$\frac{\partial D_{k+1}}{\partial x_k} = -\mathbf{I}, \quad k = 1, \dots, N-1. \quad (7.19c)$$

In summary, the objective function ψ in (7.13) is evaluated by computing the sum in (7.13b) using the algebraic variables corresponding to the solution to the discretized differential equations in (7.13d) which are solved sequentially starting from the initial condition in (7.13c) and marching forward. The right-hand side function in the discretized differential equations involves the algebraic variables which (together with the adjoint algebraic variables) satisfy the algebraic equations (7.13e). Furthermore, the gradients of ψ are evaluated based on the adjoints which are computed sequentially in a backwards manner. The discretized differential equations in (7.13d) must be solved before computing the gradients using the adjoint equations (7.15)-(7.17). The expressions for the gradient of Φ_k in (7.18a) and the Jacobian matrix of D_{k+1} in (7.19a) involve the sensitivities of the algebraic variables, $\frac{\partial y_{k+1}}{\partial x_{k+1}}$. We discuss the computation of these sensitivities in Section 5.2.2.1.

7.3 Summary

In this chapter, we have presented a gradient-based algorithm for numerical solution of the dynamic optimization problem (1.5). The algorithm uses the single-shooting method to transcribe the dynamic optimization problem into an NLP, and it uses the discrete adjoint method to compute the necessary gradients. We use both the simultaneous approach from Section 5.2.1 and the nested approach from Section 5.2.2 for the numerical solution of the involved deterministic IVPs. We have presented the adjoint method corresponding to both of these approaches.

We use the dynamic optimization algorithm in the NMPC algorithm that we present in Chapter 8, and we present a performance test of the dynamic optimization algorithm in Chapter 9 using the flash separation process described in Chapter 3. In the performance test, we compare the computational efficiency of the simultaneous and the nested approach (among other things). In Chapter 10, we present examples of optimized thermal and isothermal waterflooding strategies (based on the reservoir flow models described in Chapter 4), and we discuss the computational performance of the dynamic optimization algorithm for these examples.

Nonlinear model predictive control

In this chapter, we describe an algorithm for NMPC of stochastic semi-explicit DAEs in the form (1.3). NMPC algorithms use the moving horizon optimization principle to compute a closed-loop feedback control strategy, i.e. they compute a sequence of open-loop control strategies by solving a sequence of dynamic optimization problems. A new open-loop control strategy is computed every time new measurements become available. The objective of the optimization is either to 1) minimize the distance to predefined setpoints or 2) optimize the economics of the process. In order to solve the involved dynamic optimization problems, it is necessary to estimate the states (and potentially also unknown disturbances or parameters) based on the measurements. The NMPC algorithm that we present uses either of the four state estimation algorithms described in Chapter 6 together with the dynamic optimization algorithm presented in Chapter 7.

In Section 8.1, we discuss the control problem, and in Section 8.2, we discuss the NMPC algorithm. We summarize this chapter in Section 8.3. In Chapter 9, we present a numerical example of economical NMPC for disturbance rejection in the flash separation process described in Chapter 3, i.e. the NMPC algorithm estimates unknown disturbances while optimizing the economics of the process. In this work, the NMPC algorithm is only implemented using the EKF. The content of this chapter is based on the paper in Appendix N.

8.1 Problem formulation

In order to control a given process, it is necessary to compute expressions for the manipulated inputs, $u(t)$, according to some objective. The objective can either be to 1) follow predefined setpoints or 2) optimize the economics of the process. As mentioned previously, NMPC algorithms compute a closed-loop control strategy by computing a sequence of open-loop control strategies. Open-loop control strategies do not incorporate information from the actual process (such as measurements), whereas closed-loop control strategies do. In this work, we consider processes with measurements at discrete points in time, i.e. $y_k^m = y^m(t_k)$ for discrete times t_k . For such processes, the objective of closed-loop control is to compute expressions for the manipulated inputs for the time interval between t_k and t_{k+1} every time new measurements, y_k^m , arrive. We mention that NMPC algorithms consider a time horizon which extends past the next measurements when they compute expressions for the manipulated inputs. Furthermore, they use a dynamical model of the process (as well as a model of the sensors) in the computation of these expressions.

In this work, we consider processes that can be modeled by stochastic semi-explicit DAEs in the form (1.3) which we repeat here:

$$G(\mathbf{x}(t), \mathbf{y}(t), \mathbf{z}(t)) = 0, \quad (8.1a)$$

$$d\mathbf{x}(t) = F(\mathbf{y}(t), u(t), d(t))dt + \sigma(\mathbf{y}(t), u(t), d(t))d\boldsymbol{\omega}(t). \quad (8.1b)$$

The right-hand side of the stochastic differential equations (8.1b) contains 1) a drift term (the first term) which models the nominal dynamics of the process and 2) a diffusion term (the second term) which models the uncertainty in the process, e.g. due to unmodeled dynamics, uncertainty in the process inputs, or limited accuracy of the parameters in the model.

As mentioned above, the measurements, $\mathbf{y}^m(t_k)$, are obtained at discrete points in time, t_k , and the sensor model is

$$\mathbf{z}^m(t_k) = H(\mathbf{y}(t_k)), \quad (8.2a)$$

$$\mathbf{y}^m(t_k) = \mathbf{z}^m(t_k) + \mathbf{v}(t_k). \quad (8.2b)$$

$\mathbf{z}^m(t_k)$ are the process outputs, and the measurements are corrupted by measurement noise, $\mathbf{v}_k = \mathbf{v}(t_k)$, which is normally distributed, i.e. $\mathbf{v}_k \sim N(0, T_k)$.

8.2 The nonlinear model predictive control algorithm

In the NMPC algorithm, a new open-loop control strategy is computed whenever new measurements become available. We assume that the sampling times (the times of the measurements) coincide with the boundaries of the control intervals, i.e. that there is a single control interval between any two consecutive measurements. At time t_k , the open-loop control strategy is computed by solving the dynamic optimization problem

$$\min_{[x(t); y(t); z(t)]_{t_k}^{t_k+N_h}, \{u_{j|k}\}_{j=k}^{k+N_h-1}} \phi = \phi \left([y(t); u(t); d(t)]_{t_k}^{t_k+N_h} \right) = \int_{t_k}^{t_k+N_h} \Phi(y(t), u(t), d(t)) dt, \quad (8.3a)$$

subject to

$$x(t_k) = \hat{x}_{k|k}, \quad (8.3b)$$

$$G(x(t), y(t), z(t)) = 0, \quad t \in [t_k, t_k+N_h], \quad (8.3c)$$

$$\dot{x}(t) = F(y(t), u(t), d(t)), \quad t \in [t_k, t_k+N_h], \quad (8.3d)$$

$$u(t) = u_{j|k}, \quad t \in [t_j, t_{j+1}[, \quad j = k, \dots, k + N_h - 1, \quad (8.3e)$$

$$d(t) = \hat{d}_{j|k}, \quad t \in [t_j, t_{j+1}[, \quad j = k, \dots, k + N_h - 1, \quad (8.3f)$$

$$\{u_{j|k}\}_{j=k}^{k+N_h-1} \in \mathcal{U}, \quad (8.3g)$$

which is in the form (1.5). The vector $[x(t); y(t); z(t)]_{t_k}^{t_k+N_h}$ contains the state variables, algebraic variables, and adjoint algebraic variables which are dependent decision variables. The manipulated inputs, $\{u_{j|k}\}_{j=k}^{k+N_h-1}$, are independent decision variables, and the control horizon is N_h control intervals. The objective function ϕ in (8.3a) represents the objective of the control strategy (i.e. to minimize the distance to setpoints or to optimize the economics of the process). The objective function is an integral of the stage cost function, Φ , over the prediction horizon which is $[t_k, t_k+N_h]$, i.e. it is identical to the control horizon. The initial condition (8.3b) involves the filtered states, $\hat{x}_{k|k}$, which is the estimate of the states at time t_k given information (in particular from measurements) up to and including time t_k . In the dynamic optimization problem, we disregard the uncertainty in the process. Consequently, there is no diffusion term in the differential equations, and the semi-explicit DAEs (8.3c)-(8.3d) are deterministic. (8.3e)-(8.3f) are ZOH parametrizations of the manipulated inputs and the disturbance variables. $\{\hat{d}_{j|k}\}_{j=k}^{k+N_h-1}$ are predictions of the disturbance variables, and (8.3g) represents constraints on the manipulated inputs.

As mentioned previously, the NMPC algorithm combines either of the four state estimation algorithms described in Chapter 6 with the gradient-based dynamic optimization algorithm described in Chapter 7. Fig. 8.1 illustrates the exchange of information between the process, the state estimation algorithm, and the dynamic optimization algorithm. At time t_k , a set of measurements,

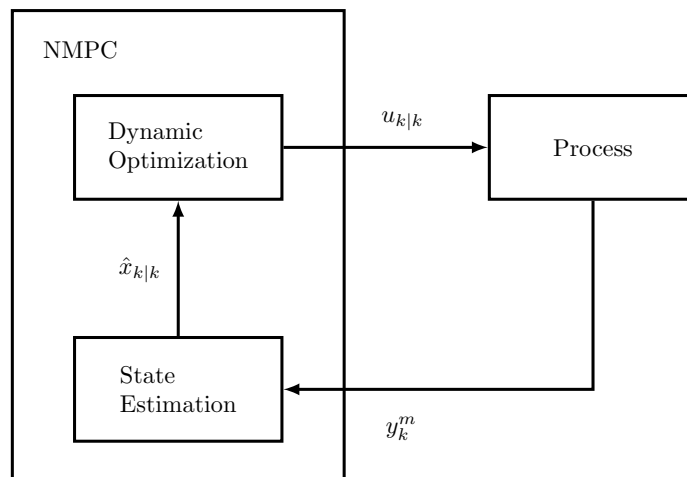


Fig. 8.1: Sketch of the exchange of information between 1) the process, 2) the state estimation algorithm, and 3) the dynamic optimization algorithm in NMPC. At time t_k , measurements, y_k^m , are obtained using the sensors. The state estimation algorithm computes the filtered states, $\hat{x}_{k|k}$, based on the measurements. The filtered states are used by the dynamic optimization algorithm which computes an open-loop control strategy, $\{u_{j|k}\}_{j=k}^{k+N_h-1}$. Finally, the manipulated inputs corresponding to the first control interval, $u_{k|k}$, are applied to the process. At time t_{k+1} , new measurements arrive, and the computations are repeated.

y_k^m , are obtained. The state estimation algorithm uses these measurements, together with the models of the process and the sensors (8.1)-(8.2), to compute the filtered states, $\hat{x}_{k|k}$. The dynamic optimization algorithm is used for numerical solution of the dynamic optimization problem (8.3) which involves $\hat{x}_{k|k}$ as mentioned above. The solution to the dynamic optimization problem includes the open-loop control strategy, i.e. the manipulated inputs $\{u_{j|k}\}_{j=k}^{k+N_h-1}$. Only the manipulated inputs corresponding to the first control interval, $u_{k|k}$, are applied to the process.

In this work, we present one numerical example that involves NMPC. In this example (which is presented in Chapter 9), we consider NMPC for disturbance rejection, i.e. the NMPC algorithm estimates unknown disturbances. Consequently, the state estimation algorithm computes estimates of both the states, $\hat{x}_{k|k}$, and the disturbance variables, $\hat{d}_{k|k}$, using the approach described in Section 6.1.1. The dynamic optimization problem (8.3) involves predictions of the disturbance variables, $\{\hat{d}_{j|k}\}_{j=k}^{k+N_h-1}$. We use $\hat{d}_{j|k} = \hat{d}_{k|k}$ for $j = k + 1, \dots, k + N_h - 1$ corresponding to an assumption of constant disturbance variables.

The state estimation algorithms that we consider in this work consist of a measurement-update and a time-update. We organize the computations in the NMPC algorithm such that after a set of measurements arrive, we 1) perform a measurement-update, 2) compute the open-loop control strategy, and 3) perform a time-update. Essentially, the time-update that is used in the estimation of the states after the next measurement arrives is initiated as soon as the current open-loop control strategy has been computed. This reduces the delay (due to nonzero computation time) from the next measurements arrive until the corresponding manipulated inputs can be applied to the process.¹ However, we assume that there is no delay (corresponding to zero computation time) in the numerical examples in this work.

8.3 Summary

In this chapter, we have described an NMPC algorithm which uses either of the state estimation algorithms from Chapter 6 as well as the gradient-based dynamic optimization algorithm from

¹Depending on the state estimation algorithm, the size of this delay may be dominated by the time used to compute the open-loop control strategy.

Chapter 7. NMPC algorithms compute a closed-loop feedback control strategy by computing a new open-loop control strategy every time new measurements become available (only the first part of the open-loop control strategy is applied to the process). We use the gradient-based dynamic optimization algorithm to compute the open-loop control strategy, i.e. as the solution to a dynamic optimization problem. In order to solve the dynamic optimization problem, it is necessary to estimate the states (using a state estimation algorithm) based on the new measurements.

In Chapter 9, we present a numerical example of economical NMPC for disturbance rejection. The example involves the flash separation process described in Chapter 3, and the NMPC algorithm uses the EKF to estimate the states (as well as the unknown disturbance).

Part III

Examples

Numerical examples – flash separation

In this chapter, we present numerical examples of 1) state estimation, 2) dynamic optimization, and 3) NMPC of the flash separation process described in Chapter 3. In Section 9.1, we compare the accuracy and the computational efficiency of the four state estimation algorithms presented in Chapter 6, i.e. the EKF, the UKF, the PF, and the EnKF, using the flash separation process. We implement the algorithms in Matlab. In Section 9.2, we present numerical solutions to two dynamic optimization problems that involve the flash separation process: 1) a tracking-type problem and 2) an economical dynamic optimization problem. We implement the dynamic optimization algorithm in Matlab and in C, and we also consider a mixed Matlab and C implementation. Furthermore, we present a performance study where we compare 1) the use of the simultaneous and the nested approach in the dynamic optimization algorithm (for numerical solution of IVPs), 2) exact and inexact Newton methods (in the simultaneous and the nested approach), 3) different optimization software, and 4) different compilers and linear algebra software. In Section 9.3, we present a numerical example of economical NMPC for disturbance rejection in the flash separation process. We consider a mixed Matlab and C implementation of the NMPC algorithm which uses the EKF to estimate the states and the unknown disturbances. The numerical examples were carried out using the workstation described in Appendix B. Finally, we summarize this chapter in Section 9.4. The numerical results presented in this chapter are based on the results presented in the papers in Appendix C, E, G, K, and N.

9.1 State estimation

In this section, we present a numerical example of state estimation of the flash separation process described in Chapter 3 using the state estimation algorithms described in Chapter 6. In this example, the state estimation algorithms use measurements of the temperature and pressure to estimate the states. Furthermore, we demonstrate that the state estimates can be used for soft sensing of the total mole fractions, the vapor-liquid mole fractions, and the vapor fraction of the mixture in the separator.¹ Physical sensors for measuring compositions (e.g. mole fractions) can be slow, expensive, and insufficiently accurate. Soft sensing based on state estimation algorithms is an economical alternative to such physical sensors. We implement the algorithms in Matlab, and we model the vapor phase and the liquid phase of the mixture in the separator as nonideal. Therefore, we use the thermodynamic model described in Section 2.5 (based on the Peng-Robinson equation of state). Furthermore, we mention that in this example, we do not reformulate the equations in the model of the flash separation process as we do in Section 9.2.

We represent the true flash separation process using the same model as the one used in the

¹Soft sensing refers to the use of software, e.g. based on state estimation algorithms, to estimate quantities that are not directly measured with physical sensors.

Table 9.1: Values of 1) the parameters and inputs in the model of the flash separation process and 2) the parameters in the state estimation algorithms. The diffusion coefficient is $\sigma(y(t), u(t), d(t)) = \sigma = \text{diag}([\sigma_U; \sigma_{C_1}; \sigma_{C_2}; \sigma_{C_3}; \sigma_{n-C_7}; \sigma_{CO_2}])$. Std. dev. refers to the standard deviation of the temperature and pressure measurement noise (temp. meas. and pres. meas.). The unit of the covariance matrix P_0 is omitted because its elements have different units. α and β are parameters in the UKF. N_p is the number of particles in the PF and the EnKF. N_k is the number of time steps in between measurements (in all four state estimation algorithms).

Model		
Quantity	Value	Unit
Total time of operation	72	h
Volume of the separator	0.2	m ³
Composition of the feed	60% C ₁ , 8% C ₂ , 5% C ₃ , 25% n-C ₇ , 2% CO ₂	
$F_F(t)$	1.0	kmol/h
$F_V(t)$	0.4	kmol/h
$F_L(t)$	0.6	kmol/h
$Q(t)$	$\begin{cases} -9 & \text{for } t \in [0 \text{ h}, 24 \text{ h}[\\ -4 & \text{for } t \in [24 \text{ h}, 72 \text{ h}] \end{cases}$	MJ/h
σ_U	1	MJ
σ_{C_1}	$1.0 \cdot 10^{-3}$	kmol
σ_{C_2}	$1.0 \cdot 10^{-3}$	kmol
σ_{C_3}	$0.1 \cdot 10^{-3}$	kmol
σ_{n-C_7}	$1.0 \cdot 10^{-3}$	kmol
σ_{CO_2}	$0.1 \cdot 10^{-3}$	kmol
Sampling time	30	min
Std. dev. (temp. meas.)	10	K
Std. dev. (pres. meas.)	$10^{-1/2}$	MPa
Algorithms		
Quantity	Value	Unit
P_0	$\sigma\sigma'$	–
α	0.1	
β	2	
N_p	100	#
N_k	6	#

state estimation algorithms, i.e. we assume that there is no plant-model mismatch. Table 9.1 shows the values of the parameters and inputs in the model of the flash separation process together with the values of the parameters in the state estimation algorithms used in this example. We use the numerical method described in Section 5.3 for the numerical simulation of the true flash separation process (with the same number of time steps between measurements as in the state estimation algorithms, i.e. $N_k = 6$). The separator feed contains CO₂ and four hydrocarbons. At the initial time, t_0 , the states of the process are normally distributed according to $N(x_0, P_0)$ where x_0 is a steady state of the separation process when the uncertainty in the system is disregarded (i.e. when the diffusion term is zero, and the semi-explicit DAEs are deterministic). The steady state is obtained using the values of the parameters and the inputs shown in Table 9.1 (with $Q(t) = -9$ MJ/h). We also use x_0 and P_0 in the initialization of the four state estimation algorithms.

Fig. 9.1 shows the true states together with the (filtered) estimates computed using the state estimation algorithms. Furthermore, it shows the true and the estimated temperature and pressure. In general, the state estimates are close to the true states. In order to quantify the accuracy of the state estimates, we introduce the root-mean-square deviation (RMSD) of the estimate of the i 'th state variable:

$$\text{RMSD}_i = \left(\frac{1}{N+1} \sum_{k=0}^N (\hat{x}_{i,k|k} - x_{i,k})^2 \right)^{1/2}. \quad (9.1)$$

$\hat{x}_{i,k|k}$ and $x_{i,k}$ are the (filtered) estimate and the true value of the i 'th state variable at time t_k . $N = 144$ is the number of sampling intervals. Both the units and the orders of magnitude vary between the state variables. Therefore, we normalize the RMSD with the time average of the true

Table 9.2: Accuracy of the EKF, the UKF, the PF, and the EnKF (in terms of average NRMSD) and average computation times of the measurement-updates and the time-updates in the algorithms. The model of the flash separation process involves 6 differential equations and 19 algebraic equations. This table originally appeared in [146].

	EKF	UKF	PF	EnKF
Avg. NRMSD (%)	2.16	1.66	1.61	1.99
Avg. CPU time of measurement-update (ms)	1.27	12.74	2.13	3.22
Avg. CPU time of time-update (ms)	8.44	161.81	773.75	805.78

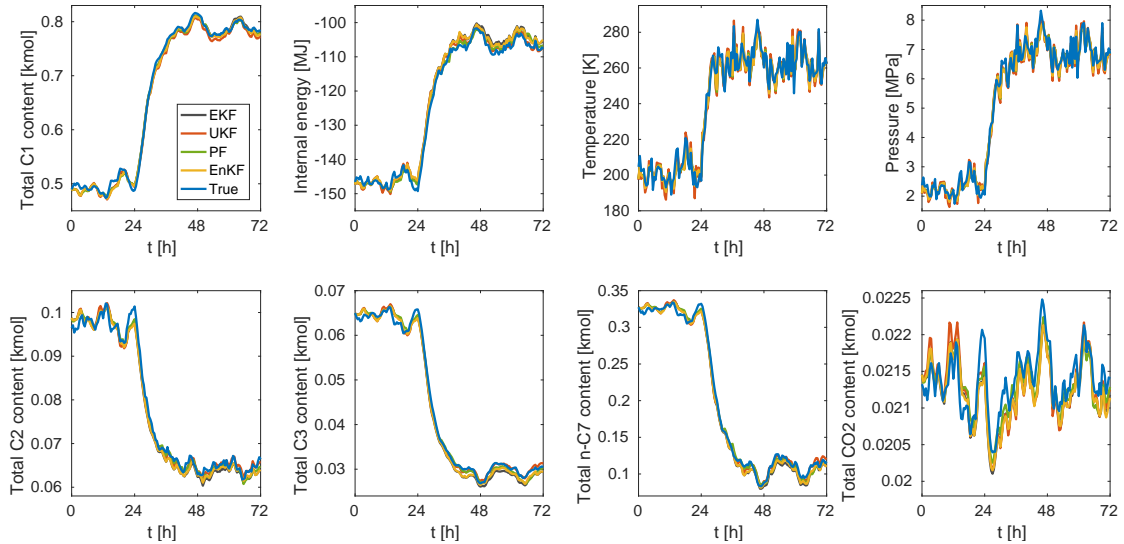


Fig. 9.1: The true values and the (filtered) estimates of the state variables (i.e. the internal energy and the total composition in moles), the temperature, and the pressure. This figure originally appeared in [146].

values of the corresponding state variables, \bar{x}_i , i.e. the average of the set $\{x_{i,k}\}_{k=0}^N$. Consequently, the normalized RMSD (NRMSD) of the i 'th state variable is $\text{NRMSD}_i = \text{RMSD}_i / \bar{x}_i$.

Table 9.2 shows the average of the set $\{\text{NRMSD}_i\}_{i=1}^{n_x}$ for each state estimation algorithm ($n_x = 6$ is the number of state variables). The table also shows the average computation time of the measurement-updates and the time-updates in the algorithms. The EKF is significantly faster than the other three algorithms while the PF is the most accurate. The accuracy of the UKF is close to that of the PF while the average NRMSD of the EnKF is significantly higher than those of the UKF and the PF. However, the EKF is the least accurate. For all four algorithms, the time-update is more computationally demanding than the measurement-update. The time-update in the EKF is close to twenty times faster than the time-update in the UKF which is almost five times faster than the time-updates in the PF and the EnKF. This is coherent with the fact that 1) the time-update in the EKF involves the numerical solution of a single deterministic IVP as well the corresponding sensitivity equations, 2) each time-update in the UKF involves the numerical solution of $2n_x + 2n_\omega + 1 = 25$ deterministic IVPs ($n_\omega = n_x = 6$ in this example), and 3) each time-update in the PF and the EnKF involves the numerical solution of $N_p = 100$ stochastic IVPs.

In Fig. 9.2, we demonstrate that the state estimates can be used for soft sensing of 1) the total mole fractions, 2) the vapor-liquid mole fractions, and 3) the vapor fraction. These estimates are computed using the (filtered) state estimates obtained with the PF.

9.2 Dynamic optimization

In this section, we present numerical solutions to 1) a tracking-type dynamic optimization problem and 2) an economical dynamic optimization problem (with the objective of minimizing the energy

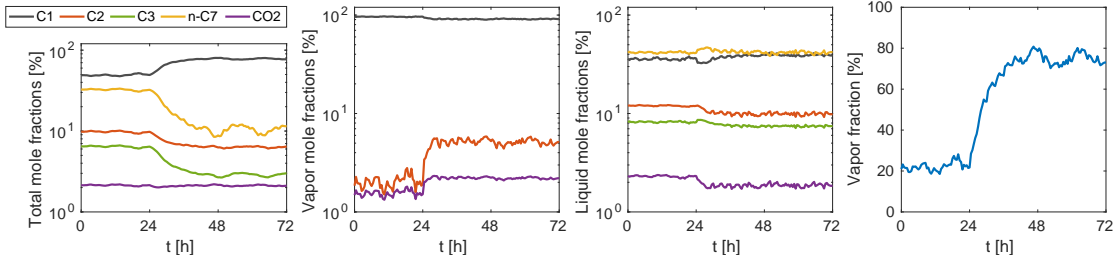


Fig. 9.2: Estimates of the total mole fractions, the vapor-liquid mole fractions, and the vapor fraction corresponding to the (filtered) state estimates computed with the PF. The graphs corresponding to the C3 and n-C7 vapor mole fractions are omitted because they are below 1%. This figure originally appeared in [146].

spent on cooling). In Section 9.2.1 and Section 9.2.2, we describe the dynamic optimization problems and discuss their numerical solutions. In Section 9.2.3, we present a performance study based on the numerical solution of the two dynamic optimization problems. The performance study involves comparisons of 1) using the simultaneous and the nested approach in the dynamic optimization algorithm (for the numerical solution of IVPs), 2) using exact and inexact Newton methods (also in the numerical solution of IVPs), 3) different optimization software, and 4) different compilers and linear algebra software. Furthermore, we consider both Matlab and C implementations (as well as a mixed implementation) in the performance study. In the numerical examples presented in this section, we have reformulated the equations in the model of the flash separation process as described in Section 6 of the paper in Appendix G. However, it is the author’s belief that the conclusions presented in this section would remain valid (and that the numerical solutions to the two dynamic optimization problems would be the same) if the model equations presented in Chapter 3 were used without reformulation.

9.2.1 The tracking-type problem

We consider the flash separation of a mixture of benzene, toluene, and diphenyl in a 10 m^3 separator. In this particular example, we model the vapor phase and the liquid phase of the mixture in the separator as an ideal gas and an ideal liquid, respectively. This is reasonable because of the similarity of the three chemical components and because we consider high temperatures and low pressures.

The objective in this tracking-type dynamic optimization problem is to compute an optimized control strategy for following predefined temperature, pressure, and liquid volume setpoints over a time interval of 4 hours, i.e. $t_0 = 0\text{ h}$ and $t_f = 4\text{ h}$. Consequently, the objective is to minimize the distance between the controlled variables (temperature, pressure, and liquid volume) and their setpoints (indicated by the superscript set):

$$\phi = \int_{t_0}^{t_f} \alpha_T (\ln T - \ln T^{\text{set}})^2 + \alpha_P (\ln P - \ln P^{\text{set}})^2 + \alpha_{V^l} (V^l - V^{l,\text{set}})^2 dt. \quad (9.2)$$

When we reformulate the equations in the model of the flash separation process (as described in Section 6 of the paper in Appendix G), the algebraic variables include the logarithmic temperature and pressure. Therefore, we also use the logarithmic temperature and pressure in the objective function (9.2) instead of the temperature and pressure. We use the weights $\alpha_T = 2000$, $\alpha_P = 20$, and $\alpha_{V^l} = 2000$ to control the relative importance of tracking each of the controlled variables. We supplement the objective function (9.2) with a regularization term which penalizes 1) the temporal change in the manipulated inputs and 2) the difference between the manipulated inputs in the first control interval and a set of reference values which are $Q_{-1} = -1\text{ MJ/h}$, $F_{V,-1} = 0.4\text{ kmol/h}$, and $F_{L,-1} = 0.6\text{ kmol/h}$. Consequently, the discrete stage costs that are used in the dynamic

optimization algorithm are

$$\hat{\Phi}_k = \Phi_k + \left(\alpha_Q (Q_k - Q_{k-1})^2 + \alpha_{F_V} (F_{V,k} - F_{V,k-1})^2 + \alpha_{F_L} (F_{L,k} - F_{L,k-1})^2 \right) \Delta t_k, \quad (9.3)$$

for $k = 0, \dots, N - 1$. The sizes of the control intervals are $\Delta t_k = 5$ min, and we use the weights $\alpha_Q = 0.05$, $\alpha_{F_V} = 10$, and $\alpha_{F_L} = 10$ to control the relative sizes of the penalizations. The derivatives of the additional regularization terms are derived analytically, and we add them to the derivatives computed with the adjoint method.

The sum of the total flow rates of the vapor-liquid streams is bounded from above by a factor of $\alpha_F = 1.2$ times the total flow rate of the feed stream:

$$F_{V,k} + F_{L,k} \leq \alpha_F F_{F,k}, \quad k = 0, \dots, N - 1. \quad (9.4)$$

Furthermore, the manipulated inputs are constrained to the intervals

$$Q_k \in [-60 \text{ MJ/h}, 10 \text{ MJ/h}], \quad k = 0, \dots, N - 1, \quad (9.5a)$$

$$F_{V,k} \in [0.1 \text{ kmol/h}, 1.5 \text{ kmol/h}], \quad k = 0, \dots, N - 1, \quad (9.5b)$$

$$F_{L,k} \in [0.1 \text{ kmol/h}, 1.5 \text{ kmol/h}], \quad k = 0, \dots, N - 1. \quad (9.5c)$$

The derivatives of the constraints (9.4)-(9.5) are derived analytically.

Fig. 9.3a shows the controlled variables (temperature, pressure, and liquid volume) obtained with an optimized strategy (blue solid) and with a reference strategy (green dashed). Both the setpoints and the manipulated inputs in the reference strategy are piecewise constant, i.e. they are constant during the first and the last 2 hours. The values of the setpoints during the first 2 hours correspond to the steady state obtained with the values of the manipulated inputs in the reference strategy in the same time interval, and similarly for the values of the setpoints during the last 2 hours. The transition to the new values of the setpoints is significantly faster with the optimized strategy than with the reference strategy. This demonstrates that the fast transition obtained with the optimized strategy is not simply due to the inherent dynamics of the process. Fig. 9.3b shows the manipulated inputs in the optimized strategy and the reference strategy together with the total feed flow rate. The two control strategies are almost identical at the beginning and the end of the considered time interval. However, they are qualitatively different during the transition between the different setpoint values. Fig. 9.4 demonstrates that the total composition and the vapor-liquid compositions of the mixture in the separator change significantly during this transition. The vapor fraction, shown in Fig. 9.3a, also changes considerably. This is due to the large differences in the values of the setpoints.

9.2.2 The economical dynamic optimization problem

We consider the flash separation of a mixture of methane (C_1), ethane (C_2), propane (C_3), n-heptane (n- C_7), and hydrogen sulfide (H_2S) in a 1 m^3 separator. In this example, we model both the vapor phase and the liquid phase of the mixture in the separator as nonideal, and we use the thermodynamic model described in Section 2.5 (with the Peng-Robinson equation of state).

The objective of the dynamic optimization problem is to minimize the energy consumption (in terms of cooling) over 24 hours of operation, i.e. $t_0 = 0$ h and $t_f = 24$ h. Since cooling corresponds to $Q \leq 0$, the objective function (to be minimized) is

$$\phi = - \int_{t_0}^{t_f} Q dt. \quad (9.6)$$

The composition of the feed is piecewise constant, i.e. it is constant during the first and the last 12 hours. The total H_2S mole fraction of the feed is 2% during the first 12 hours, and it is 6% during

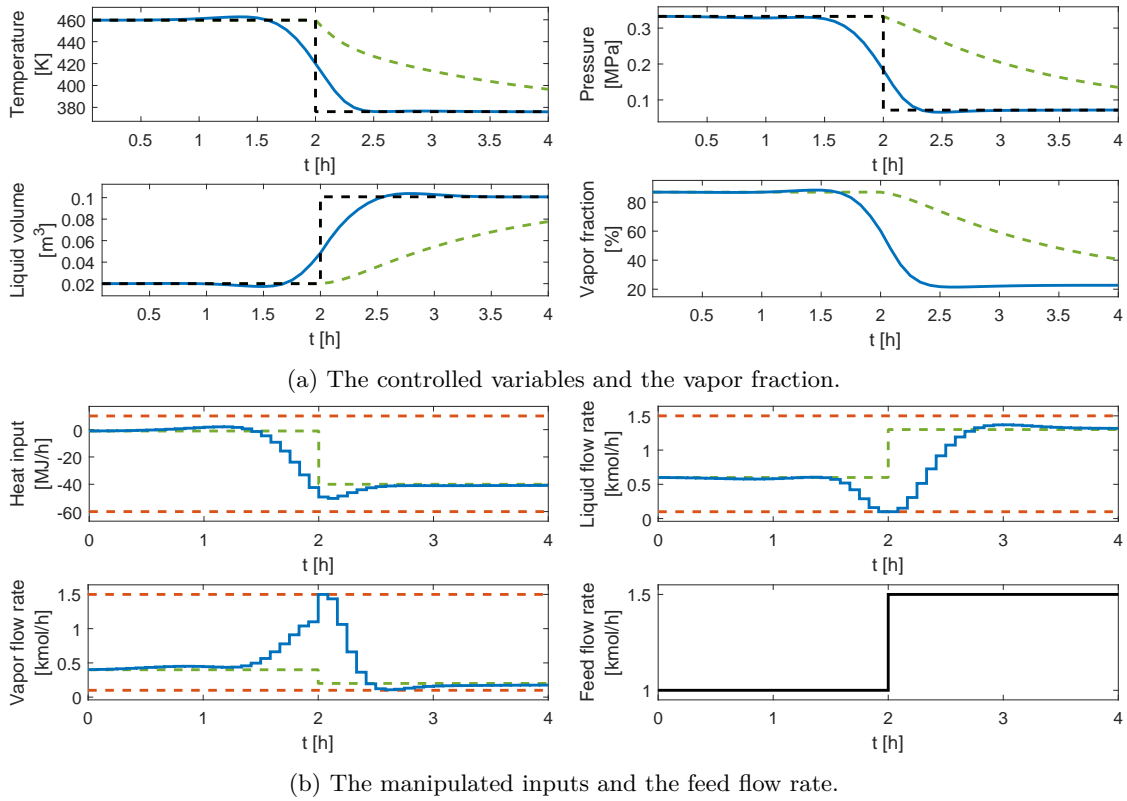


Fig. 9.3: Tracking of temperature, pressure, and liquid volume setpoints (black dashed) using 1) an optimized strategy (blue solid) subject to bound constraints (red dash-dotted) and 2) a reference strategy (green dashed). Black solid lines refer to properties of the feed. This figure originally appeared in [139].

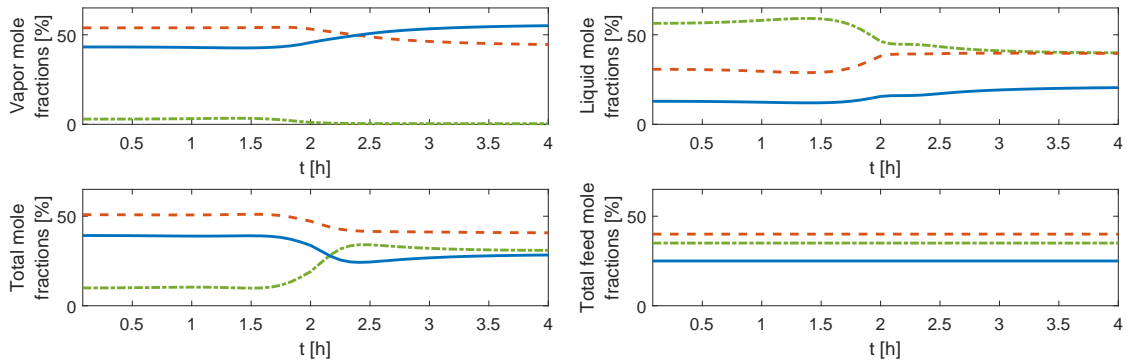


Fig. 9.4: The total mole fractions and the vapor-liquid mole fractions of the mixture in the separator obtained with the optimized tracking strategy as well as the total feed mole fractions. The mixture consists of benzene (blue solid), toluene (red dashed), and diphenyl (green dash-dotted). This figure originally appeared in [139].

the last 12 hours. The dynamic optimization problem involves an upper bound on the H₂S vapor mole fraction (i.e. the H₂S mole fraction in the vapor stream): $y_{\text{H}_2\text{S},k} \leq y_{\text{H}_2\text{S},k}^{\max}$ for $k = 1, \dots, N$. The upper bound is 2% during the first 12 hours and 4% during the last 12 hours:

$$y_{\text{H}_2\text{S},k}^{\max} = \begin{cases} 0.02, & k = 1, \dots, N/2, \\ 0.04, & k = N/2 + 1, \dots, N. \end{cases} \quad (9.7)$$

In this work, we do not model the disappearance of phases, i.e. we assume that the vapor fraction is always in the interval $]0, 1[$. Consequently, we constrain the vapor fraction such as to avoid the single-phase regions where the model of the flash separation process is invalid: $\epsilon_\beta \leq \beta_k \leq 1 - \epsilon_\beta$ for

$k = 1, \dots, N$. We consider a (small) backoff, $\epsilon_\beta = 0.05$, for practical reasons. We incorporate the constraints on the H_2S vapor mole fraction and the vapor fraction into the objective function using logarithmic barrier functions. Consequently, the stage costs are

$$\Phi_k = -Q_k \Delta t_k - \alpha_{\text{H}_2\text{S}} \ln(y_{\text{H}_2\text{S},k+1}^{\max} - y_{\text{H}_2\text{S},k+1}) - \alpha_\beta \left(\ln(\beta_{k+1} - \epsilon_\beta) + \ln((1 - \epsilon_\beta) - \beta_{k+1}) \right), \quad (9.8)$$

for $k = 0, \dots, N - 1$. The sizes of the control intervals are $\Delta t_k = 5$ min, and we choose the weights $\alpha_{\text{H}_2\text{S}} = 1.0$ and $\alpha_\beta = 0.6$. As in Section 9.2.1, we augment the objective function with a regularization term which penalizes the temporal change in the manipulated inputs, i.e. the stage costs used in the dynamic optimization algorithm are

$$\hat{\Phi}_k = \Phi_k + \left[\alpha_Q (Q_k - Q_{k-1})^2 + \alpha_{F_V} (F_{V,k} - F_{V,k-1})^2 + \alpha_{F_L} (F_{L,k} - F_{L,k-1})^2 \right] \Delta t_k, \quad (9.9)$$

for $k = 1, \dots, N - 1$, and $\hat{\Phi}_0 = \Phi_0$. In this example, we do not penalize the difference between the manipulated inputs in the first interval and a set of reference values as we did in Section 9.2.1. The derivatives of the regularization terms are derived analytically and added to the derivatives computed with the adjoint method.

The sum of the total flow rates of the vapor-liquid streams must equal the total feed flow rate:

$$F_{V,k} + F_{L,k} = F_{F,k}, \quad k = 0, \dots, N - 1. \quad (9.10)$$

Furthermore, the manipulated inputs are constrained to the intervals

$$Q_k \in [-150 \text{ MJ/h}, 0 \text{ MJ/h}], \quad k = 0, \dots, N - 1, \quad (9.11a)$$

$$F_{V,k} \in [6 \text{ kmol/h}, 8 \text{ kmol/h}], \quad k = 0, \dots, N - 1, \quad (9.11b)$$

$$F_{L,k} \in [4 \text{ kmol/h}, 6 \text{ kmol/h}], \quad k = 0, \dots, N - 1. \quad (9.11c)$$

The derivatives of the constraints (9.10)-(9.11) are derived analytically.

Fig. 9.5a shows the cumulative energy consumption and the H_2S vapor mole fraction obtained with an optimized control strategy (blue solid) and a reference strategy (green dashed). The manipulated inputs in the reference strategy are piecewise constant (i.e. they are constant during the first and the last 12 hours), and the values of the manipulated inputs are chosen such that the upper bound on the H_2S vapor mole fraction and the constraints on the vapor fraction are satisfied throughout the 24 hours of operation. During the first 12 hours, the optimized strategy operates the H_2S vapor mole fraction closer to its upper bound than the reference strategy (while still satisfying the bound). Consequently, the optimized strategy is able to reduce the energy consumption during the first 12 hours. The energy consumption in the two strategies is almost identical during the last 12 hours. This is because the total H_2S mole fraction of the feed increases more (from 2% to 6%) than the upper bound on the H_2S vapor mole fraction (which increases from 2% to 4%). Furthermore, the value of the barrier parameter $\alpha_{\text{H}_2\text{S}}$ must be chosen so high that the upper bound on the H_2S vapor mole fraction is satisfied throughout the 24 hours of operation during all iterations of the NLP solver. The optimized strategy can potentially be improved (in terms of the total energy consumption) by using different barrier parameters for each time step or by adjusting the value of the barrier parameter dynamically during the iterations of the NLP solver.² For the optimized control strategy, the total energy consumption over the 24 hours of operation is around 1700 MJ which is 26% lower than the 2400 MJ consumed with the reference strategy. This difference in energy consumption is also evident from Fig. 9.5c which shows the

²We have not adjusted the barrier parameter dynamically because we use off-the-shelf optimization software (which does not allow the user to do so).

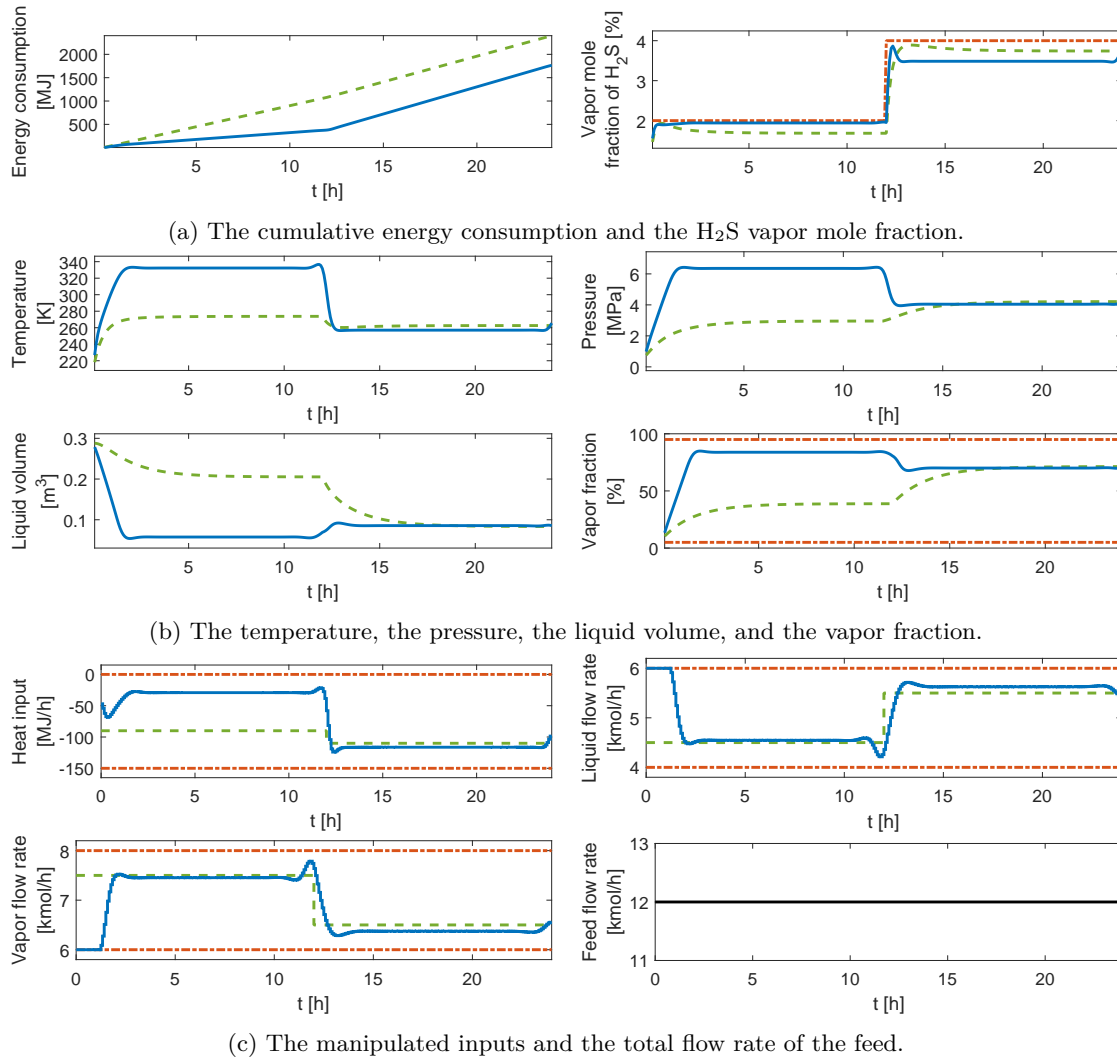


Fig. 9.5: Flash separation of a mixture consisting of four hydrocarbons and hydrogen sulfide using 1) an optimized control strategy (blue solid) subject to constraints (red dash-dotted) and 2) a reference strategy (green dashed). Black solid lines refer to properties of the feed. This figure originally appeared in [139].

manipulated inputs and the total feed flow rate. The main differences between the two strategies are 1) the values of the heat input during the first 12 hours and 2) the values of the total flow rates of the vapor-liquid streams during the first 2 hours.

The effect of the reduced cooling in the optimized strategy (during the first 12 hours) can be seen in Fig. 9.5b which shows the temperature, pressure, liquid volume, and vapor fraction. These quantities are very different for the two strategies up until time $t = 18$ h after which they are almost identical. The temperature, pressure, and vapor fractions reach significantly higher values with the optimized strategy while the liquid volume reaches significantly lower values. The total mole fractions and the vapor-liquid mole fractions corresponding to the optimized control strategy are shown (on a logarithmic scale) in Fig. 9.6 together with the mole fractions of the feed. The compositions do not change significantly during the 24 hours of operation. The vapor phase is mainly methane while the liquid phase is mainly n-heptane (and methane).

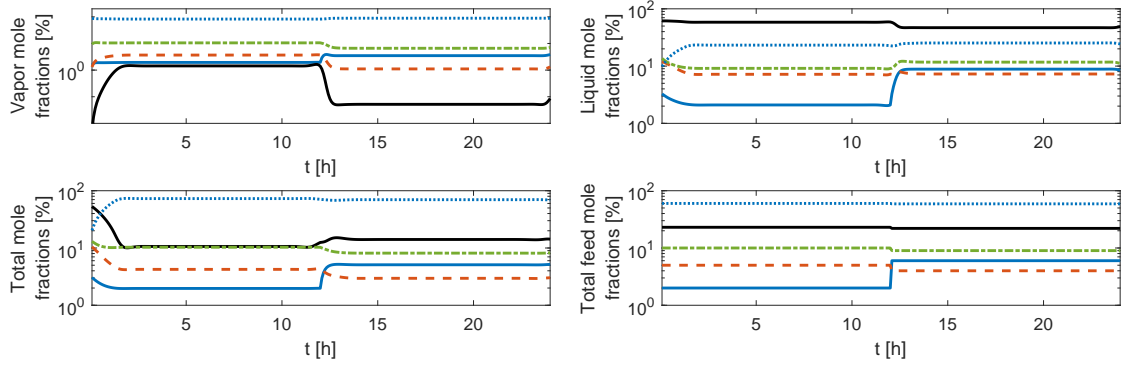


Fig. 9.6: The total mole fractions and the vapor-liquid mole fractions of the mixture in the separator corresponding to the optimized control strategy as well as the total feed mole fractions. The mixture consists of methane (blue dotted), ethane (green dash-dotted), propane (red dashed), heptane (black solid), and hydrogen sulfide (blue solid). This figure originally appeared in [139].

Table 9.3: Key characteristics of the tracking-type problem and the economical dynamic optimization problem presented in Section 9.2.1 and Section 9.2.2, respectively.

Problem	Tracking	Economical
Independent decision variables	144	864
Control intervals/time steps	48	288
Differential equations	4	6
Algebraic equations	5	7

9.2.3 Performance study

As mentioned previously, we present a performance study based on the numerical solution of the tracking-type problem discussed in Section 9.2.1 and the economical dynamic optimization problem discussed in Section 9.2.2. Table 9.3 shows four key characteristics of the two dynamic optimization problems. The main difference between the two problems is the number of control intervals (and therefore also the number of independent decision variables). The economical dynamic optimization problem involves more independent decision variables (manipulated inputs). Consequently, it is also more computationally demanding to solve. In the performance study, we compare the efficiency of using 1) the simultaneous and the nested approach (for the numerical solution of IVPs in the dynamic optimization algorithm), 2) exact and inexact Newton methods (in the simultaneous and the nested approach), 3) different NLP solvers, and 4) different compilers and linear algebra software.

9.2.3.1 The simultaneous and the nested approach. We compare the simultaneous and the nested approach using a Matlab implementation of the dynamic optimization algorithm. The implementation uses Matlab routines from the thermodynamic library ThermoLib as well as an interior point barrier method implemented in Matlab's NLP solver, `fmincon`. Table 9.4 shows several key performance indicators (KPIs) of the implementation. It shows that it is more efficient (in terms of computation time) to use the simultaneous approach in the dynamic optimization algorithm regardless of whether an exact or an inexact Newton method is used. Furthermore, the table shows that the dynamic optimization algorithm requires fewer factorizations, back substitutions, and evaluations of thermodynamic functions when the simultaneous approach is used. Table 9.5 shows the computation times of the factorizations, back substitutions, and thermodynamic function evaluations in Matlab and in C. In Matlab, the computation time of the thermodynamic function evaluations is significantly higher than the computation times of the factorizations and back substitutions. Consequently, it is mainly because of the fewer evaluations of thermodynamic functions (and their derivatives) that the dynamic optimization algorithm is more efficient when

Table 9.4: Key performance indicators of a Matlab implementation of the dynamic optimization algorithm for the numerical solution of the tracking-type problem and the economical dynamic optimization problem discussed in Section 9.2.1 and Section 9.2.2, respectively. Sim. refers to the use of the simultaneous approach for numerical solution of the involved IVPs, and Nest. refers to the use of the nested approach. We test both approaches using both exact and inexact Newton methods. CPU time refers to the relative computation times of the numerical solution of the involved IVPs and the computation of gradients using the adjoint method (Adj.). We stress that the relative computation times in the two problems are unrelated. Iter. and Func. refer to the number of iterations and function evaluations in the NLP solver (Matlab’s `fmincon`). Fact. and Back. sub. refer to the number of factorizations and back substitutions in the solution of the involved linear systems. Finally, Thermo. eval. refers to the number of evaluations of thermodynamic properties (and their derivatives) necessary to evaluate the residual equations (Res. eq.) and the Jacobian matrices (Jac.) in the numerical solution of the IVPs and the computation of the gradients with the adjoint method. This table originally appeared in [139].

The tracking-type problem								
	CPU time		NLP solver		Lin. Alg.		Thermo. eval.	
	IVP	Adj.	Iter.	Func.	Fact.	Back. sub.	Res. eq.	Jac.
Sim. (inexact)	0.271	0.118	202	203	20,881	42,946	33,202	20,881
Sim. (exact)	0.264	0.118	202	203	32,632	32,632	0	42,376
Nest. (inexact)	1.000	0.145	202	203	138,358	179,585	105,328	118,853
Nest. (exact)	0.680	0.140	202	203	142,507	142,507	0	113,792

The economical dynamic optimization problem								
	CPU time		NLP solver		Lin. Alg.		Thermo. eval.	
	IVP	Adj.	Iter.	Func.	Fact.	Back. sub.	Res. eq.	Jac.
Sim. (inexact)	0.184	0.094	156	173	103,763	159,501	109,677	103,763
Sim. (exact)	0.300	0.116	147	213	166,462	166,462	0	227,806
Nest. (inexact)	1.000	0.129	150	230	738,140	889,473	475,960	605,349
Nest. (exact)	0.792	0.123	146	205	654,674	654,674	0	507,831

the simultaneous approach is used. In C, the computation time of the thermodynamic function evaluations is also higher, but the difference is significantly larger in Matlab.

9.2.3.2 Exact and inexact Newton methods. We use the same Matlab implementation as in Section 9.2.3.1 to compare the use of exact and inexact Newton methods in the simultaneous and the nested approach. The computation times are shown in Table 9.4. When the nested approach is used in the dynamic optimization algorithm, it is more efficient to use an exact Newton method for both of the dynamic optimization problems. Presumably, this is because the Jacobian matrix is updated frequently in the inner Newton iterations when using an inexact Newton method. When the simultaneous approach is used in the dynamic optimization algorithm, it is not advantageous to use an inexact Newton method for the tracking-type problem (where the vapor phase and the liquid phase are modeled as ideal). This is because there is little difference in the computation times associated with the evaluation of the thermodynamic functions (and their derivatives) for 1) the residual equations and 2) the Jacobian matrices as can be seen from Table 9.5. For the economical dynamic optimization problem (where the vapor phase and the liquid phase are modeled as nonideal), it is significantly more computationally demanding to evaluate the thermodynamic functions (and their derivatives) for the Jacobian matrices than for the residual equations. Consequently, the computation time of the dynamic optimization algorithm is approximately 35% lower for the economical dynamic optimization problem when using the simultaneous approach with an inexact Newton method. This improvement in efficiency is also partly due to the lower number of function evaluations in the NLP solver.

9.2.3.3 Programming language, NLP solvers, and compilers. Table 9.6 shows KPIs for several implementations of the dynamic optimization algorithm. All implementations use the simultaneous approach for the numerical solution of the involved IVPs. We implement the dynamic

Table 9.5: The absolute computation times (given in microseconds) of 1) evaluating thermodynamic properties and their derivatives and 2) solving $Ax = b$ for x where A is dense. The sizes of the matrices are the same as the sizes of the system matrices involved in the Newton iterations in the simultaneous approach (Sim.) and in the inner Newton iterations (Nest. inner) and the outer Newton iterations (Nest. outer) in the nested approach. As in Table 9.4, we distinguish between evaluating the thermodynamic properties and derivatives necessary for evaluating the residual equations (Res. eq.) and the Jacobian matrices (Jac.) in the simultaneous and the nested approach and in the corresponding adjoint methods. We use an LU factorization (Fact.) and a back substitution (Back. sub.) to solve the linear systems. In C, we use Netlib's LAPACK routines `dgetrf` and `dgetrs` to carry out the factorizations and the back substitutions. This table originally appeared in [139].

The tracking-type problem								
	Thermo. eval.		Sim. ($A \in \mathbb{R}^{9 \times 9}$)		Nest. inner ($A \in \mathbb{R}^{5 \times 5}$)		Nest. outer ($A \in \mathbb{R}^{4 \times 4}$)	
	Res. eq.	Jac.	Fact.	Back. sub.	Fact.	Back. sub.	Fact.	Back. sub.
Matlab	34.588	39.829	3.514	2.271	2.844	2.117	2.634	1.887
C	1.435	1.428	0.575	0.188	0.205	0.112	0.145	0.092

The economical dynamic optimization problem								
	Thermo. eval.		Sim. ($A \in \mathbb{R}^{13 \times 13}$)		Nest. inner ($A \in \mathbb{R}^{7 \times 7}$)		Nest. outer ($A \in \mathbb{R}^{6 \times 6}$)	
	Res. eq.	Jac.	Fact.	Back. sub.	Fact.	Back. sub.	Fact.	Back. sub.
Matlab	211.319	464.254	5.121	2.516	3.169	2.213	3.032	2.151
C	4.305	6.431	1.165	0.290	0.359	0.150	0.288	0.129

optimization algorithm in both Matlab and C. The Matlab implementation uses the Matlab routines from ThermoLib and the C implementations use the C routines. The implementations use either of four different NLP solvers: Matlab's `fmincon`, IPOPT 3.12.4 [180], NPSOL 5.0, and KNITRO 10.2. Matlab's `fmincon` and KNITRO both implement four algorithms. In both cases, we use an interior point barrier method. IPOPT also uses an interior point barrier method, and NPSOL uses an active set sequential quadratic programming (SQP) method. For the C implementations, we use 1) GCC compilers and Netlib's implementation of BLAS and LAPACK (for linear algebra computations) and 2) Intel compilers and Intel's MKL (for linear algebra computations). We compile ThermoLib and IPOPT with a C compiler, and we compile NPSOL with a Fortran compiler. KNITRO is compiled by the software vendor.

The least efficient implementation is the pure Matlab implementation which uses `fmincon`. A mixed Matlab and C implementation which uses `fmincon` together with C implementations of the simultaneous approach (for the numerical solution of the IVPs) and the corresponding adjoint method (for the gradient computations) is 10 times faster than the pure Matlab implementation for the tracking-type problem and 3.7 times faster for the economical dynamic optimization problem. The C implementation that uses IPOPT is inefficient for the tracking-type problem because it uses a limited-memory BFGS approximation of the Hessian. The economical dynamic optimization problem involves more independent decision variables. Consequently, the implementation that uses IPOPT is more efficient for this problem where its performance matches the performance of the implementations that use KNITRO and NPSOL (when GCC compilers and Netlib's BLAS/LAPACK implementation are used). In general, the implementation using NPSOL is the fastest, and its performance is improved significantly when Intel compilers and Intel's MKL are used. In this case, it is 83 times faster than the pure Matlab implementation for the tracking-type problem and 55 times faster for the economical dynamic optimization problem. The use of Intel compilers and Intel's MKL does not improve the computational performance of the other implementations significantly.

Table 9.6: The absolute computation times (given in seconds) and the relative computation times of solving the tracking-type problem and the economical dynamic optimization problem using the dynamic optimization algorithm (with the simultaneous approach). Iter. and Func. refer to the number of iterations and function evaluations in the NLP solver. The computation times are averaged over 10 numerical solutions of the dynamic optimization problems. We compare the use of GCC compilers and Netlib’s implementation of BLAS/LAPACK to the use of Intel compilers and Intel’s MKL. This table originally appeared in [139].

The tracking-type problem									
	GCC and Netlib					Intel			
NLP Sol.	fmincon	fmincon	IPOPT	KNITRO	NPSOL	fmincon	IPOPT	KNITRO	NPSOL
Prog.	Matlab	C	C	C	C	C	C	C	C
Iter.	202	202	486	186	152	202	486	186	152
Func.	203	203	1517	190	153	203	1517	190	153
Absolute	8.503	0.851	1.319	0.291	0.223	0.843	0.911	0.245	0.102
Relative	1.000	0.100	0.155	0.034	0.026	0.099	0.107	0.029	0.012
Speedup	1.0	10.0	6.4	29.2	38.1	10.1	9.3	34.7	83.4

The economical dynamic optimization problem									
	GCC and Netlib					Intel			
NLP Sol.	fmincon	fmincon	IPOPT	KNITRO	NPSOL	fmincon	IPOPT	KNITRO	NPSOL
Prog.	Matlab	C	C	C	C	C	C	C	C
Iter.	156	156	328	117	150	156	328	117	150
Func.	173	173	1364	121	152	173	1364	121	152
Absolute	137.803	37.525	6.868	6.654	7.002	37.929	6.437	8.149	2.481
Relative	1.000	0.272	0.050	0.048	0.051	0.275	0.047	0.059	0.018
Speedup	1.0	3.7	20.1	20.7	19.7	3.6	21.4	16.9	55.5

9.3 Nonlinear model predictive control

In this section, we present a numerical example of economical NMPC of the flash separation process described in Chapter 3. We stress that in this example, we do not reformulate the model equations as we did in Section 9.2. We consider the separation of a mixture of hydrogen sulfide (H_2S) and four hydrocarbons. We assume that the vapor phase and the liquid phase of the mixture are nonideal, and we use the thermodynamic model presented in Section 2.5 (with the Peng-Robinson equation of state). The objective is to minimize the energy consumption in terms of cooling while satisfying constraints on 1) the H_2S vapor mole fraction, 2) the vapor fraction, and 3) the pressure. The NMPC algorithm uses the EKF to estimate the states as well as an unknown disturbance variable (the feed temperature) based on measurements of the temperature and the pressure of the mixture in the separator. We use the approach described in Section 6.1.1 to estimate the unknown disturbance with the EKF. We consider a mixed Matlab and C implementation of the NMPC algorithm where the simultaneous approach (for the numerical solution of the involved IVPs) and the corresponding adjoint method (for the computation of gradients) are implemented in C, and the remaining parts of the algorithm (e.g. the EKF) are implemented in Matlab. The implementation uses an interior point barrier method from Matlab’s fmincon to solve the involved NLPs.

We represent the true flash separation process with the same model as the one used in the NMPC algorithm. However, the NMPC algorithm does not know the true temperature of the feed, T_F , as mentioned above. The vapor phase and the liquid phase of the feed mixture is assumed to be in thermal, mechanical, and chemical equilibrium at specified temperature, T_F , pressure, P_F , and composition, f_F . Consequently, the vapor-liquid composition of the feed (in moles) is the solution to a PT flash which involves the feed temperature.³ We refer to Section VI of the paper in

³In the NMPC algorithm, we only solve the PT flash for the vapor-liquid composition of the feed once for each dynamic optimization problem because the disturbance variables are assumed to be constant in time (in the dynamic optimization problems).

Appendix N for more details on the relation between the feed temperature and the vapor-liquid composition of the feed in this example. Table 9.7 shows 1) the values of the parameters and the inputs in the model of the flash separation process and 2) the values of the parameters in the NMPC algorithm. We use the numerical method described in Section 5.3 for the numerical simulation of the true flash separation process (using the same number of time steps in between the measurements as in the NMPC algorithm, i.e. $N_k = 1$). The initial states of the true process are sampled from $N(x_0, P_0^{xx})$ where x_0 is a steady state of the nominal separation process, i.e. when the process noise is disregarded and the diffusion term in the stochastic differential equation is zero. We use P_0^{xx} instead of P_0 to distinguish it from P_0^{dd} which is the covariance of the initial estimate of the disturbance variable. x_0 and P_0^{xx} are also used in the initialization of the EKF in the NMPC algorithm. The steady state, x_0 , is obtained using the values of the parameters and the inputs shown in Table 9.7 as well as $Q(t) = -100$ MJ/h, $F_V(t) = 6.5$ kmol/h, and $F_L(t) = 5.5$ kmol/h. The values of σ_d and P_0^{dd} are relevant to the estimation of the unknown disturbance, and the values shown in Table 9.7 are chosen by numerical experiment. A high value of σ_d allows for rapid variations in the estimated feed temperature, and a high value of P_0^{dd} corresponds to low confidence in the initial estimate of the feed temperature. The initial estimate is equal to the actual temperature of the feed at the initial time which is 335 K. However, after 2 hours of operation, the real feed temperature increases instantaneously to 340 K. The NMPC algorithm does not know about this increase.

As mentioned previously, the objective is to minimize the energy spent on cooling. Consequently, the objective function in the dynamic optimization problem that is solved at time t_k is

$$\phi = - \int_{t_k}^{t_k + N_h} Q(t) dt, \quad (9.12)$$

since $Q \leq 0$ in the case of cooling. The length of the prediction and control horizon is 6 h corresponding to $N_h = 72$ since we use control intervals of length 5 min. As in the economical dynamic optimization problem described in Section 9.2.2, we consider 1) an upper bound on the H₂S vapor mole fraction (i.e. the H₂S mole fraction of the vapor stream) and 2) bounds on the vapor fraction (such as to avoid the single-phase regions where the model of the flash separation process is invalid). Furthermore, we consider an upper bound on the pressure, P . We incorporate these constraints into the objective function using logarithmic barrier functions. The constraints on the H₂S vapor mole fraction and the vapor fraction are incorporated as described in Section 9.2.2, and the upper bound on the pressure is incorporated in a similar manner (using the barrier parameter α_P). The upper bound on the H₂S vapor mole fraction is 2%, and we do not consider a backoff in the constraints on the vapor fraction, i.e. $\epsilon_\beta = 0$. The values of the barrier parameters are $\alpha_{\text{H}_2\text{S}} = 10$, $\alpha_\beta = 10$, and $\alpha_P = 5$. Furthermore, we penalize the rate of change of the manipulated inputs and the difference between the manipulated inputs in the first control interval and a set of reference values as described in Section 9.2.1. For the dynamic optimization problem that is solved at time t_k , the reference values are the manipulated inputs in the previous control interval, $u_{k-1|k-1}$. The penalization weights are $\alpha_Q = 1$, $\alpha_{F_V} = 60$, and $\alpha_{F_L} = 60$. Finally, the manipulated inputs are subject to the constraints

$$Q(t) \in [-100 \text{ MJ/h}, 0 \text{ MJ/h}], \quad (9.13a)$$

$$F_V(t) \in [6.5 \text{ kmol/h}, 8.5 \text{ kmol/h}], \quad (9.13b)$$

$$F_L(t) \in [3.5 \text{ kmol/h}, 5.5 \text{ kmol/h}]. \quad (9.13c)$$

Fig. 9.7 shows a closed-loop simulation of economical NMPC for disturbance rejection (i.e. estimation of the unknown disturbance) in the flash separation process using the NMPC algorithm. Both the H₂S vapor mole fraction and the pressure are operated close to their upper bounds which are satisfied throughout the 8 hours of operation despite the unknown disturbance and the

Table 9.7: Values of 1) the parameters and the inputs in the model of the flash separation process and 2) the parameters in the NMPC algorithm. The diffusion coefficient is $\sigma(y(t), u(t), d(t)) = \sigma = \text{diag}([\sigma_U; \sigma_{C_1}; \sigma_{C_2}; \sigma_{C_3}; \sigma_{n-C_7}; \sigma_{H_2S}])$. Std. dev. refers to the standard deviation of the temperature and pressure measurement noise (temp. meas. and pres. meas.). The unit of the covariance matrix P_0^{xx} is omitted because its elements have different units. The EKF uses σ_d and P_0^{dd} in the estimation of the unknown disturbance. N_k is the number of time steps in between measurements in the EKF and in the dynamic optimization algorithm.

Model		
Quantity	Value	Unit
Total time of operation	8	h
Volume of the separator	1	m ³
Composition of the feed	60% C ₁ , 10% C ₂ , 5% C ₃ , 23% n-C ₇ , 2% H ₂ S	
Feed temperature	$\begin{cases} 335 & \text{for } t \in [0 \text{ h}, 2 \text{ h}[\\ 340 & \text{for } t \in [2 \text{ h}, 8 \text{ h}] \end{cases}$	K
$F_F(t)$	12	kmol/h
σ_U	0.2	MJ
σ_{C_1}	$2.0 \cdot 10^{-3}$	kmol
σ_{C_2}	$2.0 \cdot 10^{-3}$	kmol
σ_{C_3}	$0.2 \cdot 10^{-3}$	kmol
σ_{n-C_7}	$2.0 \cdot 10^{-3}$	kmol
σ_{H_2S}	$0.2 \cdot 10^{-3}$	kmol
Sampling time	5	min
Std. dev. (temp. meas.)	2.5	K
Std. dev. (pres. meas.)	0.05	MPa

Algorithm		
Quantity	Value	Unit
Prediction/control horizon	6	h
Length of control intervals	5	min
P_0^{xx}	$\sigma\sigma'$	–
σ_d	0.5	K
P_0^{dd}	0.25^2	K ²
N_k	1	#

uncertainty in the process. The vapor fraction is never close to its bounds. Fig. 9.8 shows the estimated feed temperature (i.e. the unknown disturbance variable) and the manipulated inputs which are applied to the process (i.e. $u_{k|k}$ for each control interval). The EKF is able to track the increase in the true feed temperature, and 4 hours after the increase, the estimate almost equals the true value. Furthermore, the NMPC algorithm is able to reduce the energy spent on cooling, and the manipulated inputs are almost constant towards the end of the 8 hours of operation.

It is important for the computational feasibility of real-time implementation of NMPC algorithms that the computation time per control interval is significantly lower than the length of the control intervals. In this example, the computation time is highest (3.14 s) for the first control interval. This is because the initial guess for the first dynamic optimization problem is chosen manually. For the subsequent dynamic optimization problems, the previous optimized control strategy is used in the construction of the initial guess. The second-highest, the mean, and the lowest computation time are 1.11 s, 0.64 s, and 0.21 s, respectively. These computation times are significantly lower than the size of the control intervals (5 min). The computation times reported in the paper in Appendix N are based on a pure Matlab implementation of the NMPC algorithm. Therefore, they are significantly higher than the above computation times.

9.4 Summary

In this chapter, we have presented numerical examples of 1) state estimation, 2) dynamic optimization, and 3) NMPC of the flash separation process described in Chapter 3.

In the state estimation example, we compare the accuracy and the computational performance of Matlab implementations of the four state estimation algorithms. For this example, the estimates

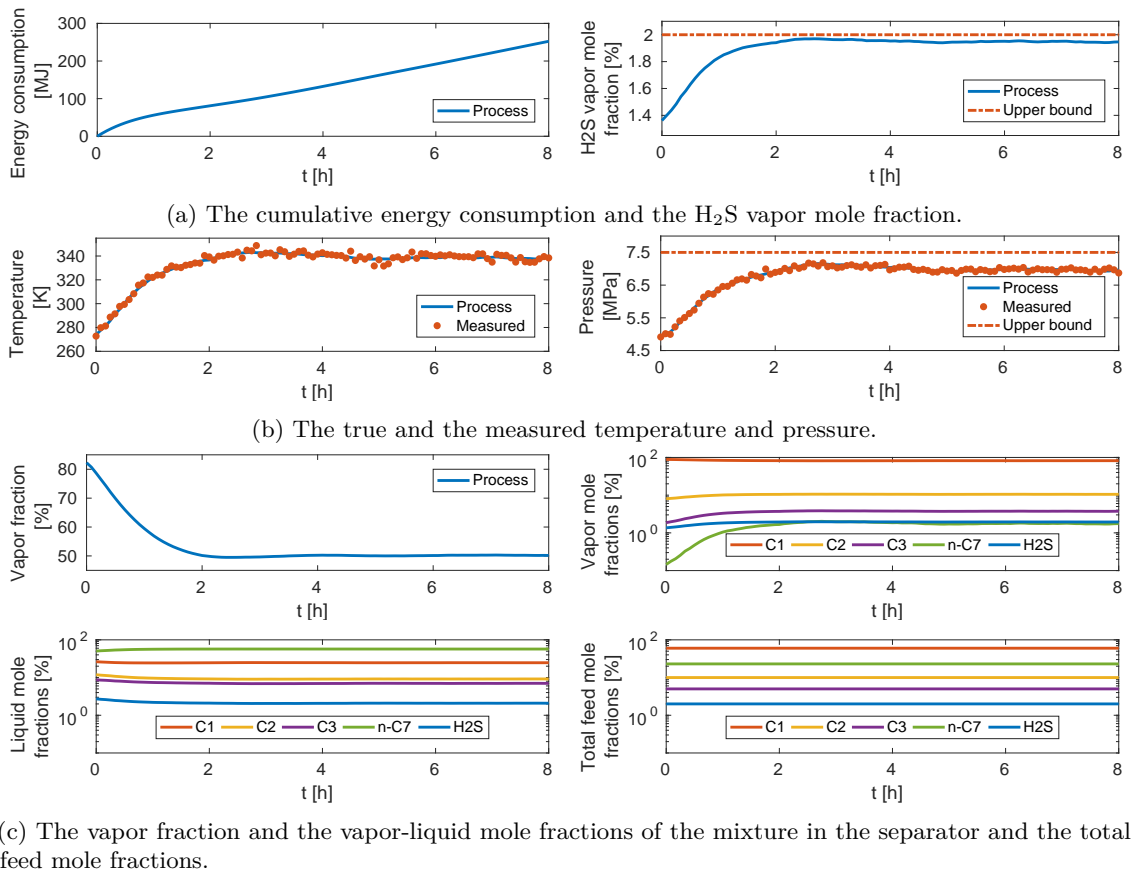


Fig. 9.7: Closed-loop simulation of economical NMPC for disturbance rejection in the flash separation process. The red dash-dotted lines indicate bounds. This figure originally appeared in the paper in Appendix N.

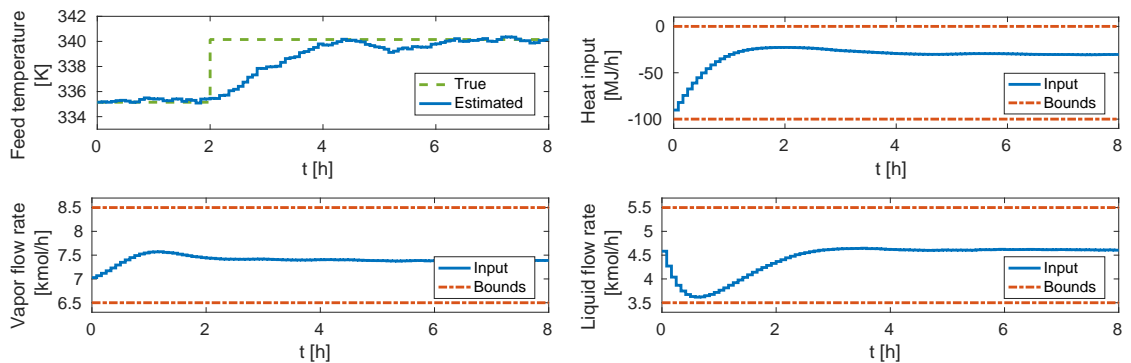


Fig. 9.8: The estimated and the true feed temperature, T_F , and the manipulated inputs. The red dash-dotted lines indicate bounds. This figure originally appeared in the paper in Appendix N.

computed with the four algorithms are all close to the true states. Furthermore, the EKF is significantly faster than the other state estimation algorithms, but it is also less accurate. The PF is the most accurate, but it is also significantly slower than the EKF and the UKF. The UKF is only slightly less accurate than the PF. Finally, the EnKF is roughly as slow as the PF but less accurate. Furthermore, we have demonstrated that the state estimation algorithms can be used for soft sensing of the vapor-liquid compositions and the vapor fraction of the mixture in the separator.

We have presented a performance study of the dynamic optimization algorithm based on the

numerical solution of two dynamic optimization problems. In this performance study, we conclude that 1) the simultaneous approach (for the numerical solution of IVPs) is more efficient than the nested approach, 2) it is more efficient to use an inexact Newton method in the simultaneous approach for one of the problems, and it is faster to use an exact Newton method in the nested approach for both problems, 3) the C implementations are significantly faster than the pure Matlab implementation and the mixed Matlab and C implementation, and 4) the NLP solver NPSOL provides the best computational performance of the dynamic optimization algorithm (when using Intel compilers and Intel's MKL).

In the NMPC example, we consider economical NMPC for disturbance rejection in the flash separation process using a mixed Matlab and C implementation of the NMPC algorithm. The algorithm uses the EKF to estimate an unknown disturbance (as well as the states), and the objective is to minimize the energy consumption while satisfying output constraints (i.e. constraints that involve the dependent decision variables in the dynamic optimization problem). We also discuss the computational performance of the NMPC algorithm in this example.

Numerical examples – waterflooding

In this chapter, we present numerical examples of optimized thermal and isothermal waterflooding strategies, i.e. we consider dynamic optimization problems that involve the thermal and isothermal compositional reservoir flow models described in Chapter 4. In Section 10.1, we use the dynamic optimization algorithm presented in Chapter 7 for the numerical solution of the two dynamic optimization problems, i.e. to compute the optimized thermal and isothermal waterflooding strategies. Furthermore, we discuss the computational performance of the algorithm for these two problems. The computations were carried out using the workstation described in Appendix B. We briefly summarize this chapter in Section 10.2. The numerical results presented in this chapter are based on the results presented in the paper in Appendix M.

10.1 Dynamic optimization

In this section, we present the optimized thermal and isothermal waterflooding strategies, and we discuss the computational performance of the dynamic optimization algorithm for the computation of these two strategies. We consider a $110 \times 110 \times 10$ m rectangular reservoir, and we discretize the reservoir using an $11 \times 11 \times 1$ rectangular grid. At the initial time, the temperature of the reservoir fluid and rock is 50°C . The temperature of the injected water is 90°C in the thermal waterflooding strategy and 50°C in the isothermal strategy. The oil and the gas phase of the reservoir fluid contain methane (C_1), ethane (C_2), propane (C_3), n-heptane (n- C_7), and hydrogen sulfide (H_2S). As mentioned in Section 4.2.6 in the description of the reservoir flow models, we model the oil and the gas phase as nonideal and the water phase as an ideal liquid. Consequently, we use the thermodynamic model described in Section 2.5 (based on the Peng-Robinson equation of state) to evaluate the thermodynamic properties of the oil and the gas phase, and we use the thermodynamic model described in Section 2.4 to evaluate the thermodynamic properties of the water phase. The objective of the optimization is to maximize the total oil production (in moles) over a period of three years, i.e. $t_0 = 0$ y and $t_f = 3$ y. The objective function is

$$\phi = - \int_{t_0}^{t_f} Q^{o,\text{prod}} dt. \quad (10.1)$$

The sign is negative because the objective is to minimize ϕ . The manipulated inputs are the BHPs of four injectors and a single producer. The BHPs are constrained to the intervals

$$P_{i,k}^{\text{bhp}} \in [10 \text{ MPa}, 12 \text{ MPa}], \quad i \in \mathcal{I}, \quad k = 0, \dots, N - 1, \quad (10.2a)$$

$$P_{i,k}^{\text{bhp}} \in [9 \text{ MPa}, 10 \text{ MPa}], \quad i \in \mathcal{P}, \quad k = 0, \dots, N - 1, \quad (10.2b)$$

where \mathcal{I} and \mathcal{P} contain the indices of the grid cells in which the injectors and producers are perforated, respectively. Each well is only perforated in one grid cell. We consider $N = 36$ control

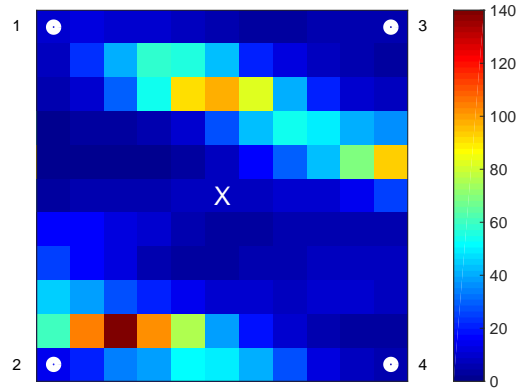


Fig. 10.1: The permeability field (in mD) and the locations of the injectors (white circles) and the producer (white X). This figure originally appeared in [145].

intervals, and each control interval can contain multiple time steps because the time step sizes are chosen automatically as described below. The locations of the injectors and the producer are shown in Fig. 10.1 together with the (heterogeneous and isotropic) permeability field. We assume that 1) there is no heat exchange with the surroundings of the reservoir, 2) the rock is incompressible, and 3) the porosity field is homogeneous. The porosity is 0.25, the thermal heat capacity of the rock is 0.92 kJ/(kg · K), and the thermal conductivity of the rock is 2.5 W/(m · K). These values resemble the properties of sandstone [51, Chap. 2].

We consider a C++ implementation of the dynamic optimization algorithm based on the open-source software DUNE for grid management [11, 12] and for the solution of linear systems with iterative methods [21]. We solve the involved linear systems using the GMRES method (an iterative method) with a block ILU(1) preconditioner. We use the simultaneous approach for the numerical solution of the involved IVPs, and we use a simplified form of the scheme described by Völcker et al. [178] for automatic time step size selection. We use a sequential linear-quadratic programming (SLQP) method [115, Chap. 18] from KNITRO 10.2 to solve the involved NLPs. Finally, we use C routines from ThermoLib, and we compile the code using GCC C/C++ compilers.

Fig. 10.2 shows the optimized thermal and isothermal waterflooding strategies, i.e. the optimized injector and producer BHPs. Furthermore, it shows the cumulative volume of the injected water and the produced oil and gas. In both strategies, the producer is operated close to its minimum BHP. Furthermore, the BHPs of injector 3 and 4 are operated close to their maxima. This is because they are located in areas with low permeability. The BHPs of injector 1 and 2 are varied significantly in both the thermal and the isothermal strategy. The two strategies inject almost the same volume of water, and they produce almost the same volume of oil and gas. However, the thermal strategy injects slightly less water, produces slightly less oil, and produces slightly more gas. Fig. 10.3 and Fig. 10.4 show the temperature (for the thermal strategy), the pressure, and the oil and gas saturations in the reservoir. The pressure and the oil and gas saturations are fairly similar in the two optimized strategies. For both strategies, some oil remains in the upper part of the reservoir after the three years of production. This suggests that it is difficult to retrieve all of the oil in this part in three years.

Table 10.1 shows four key characteristics of the two dynamic optimization problems as well as a number of KPIs of the dynamic optimization algorithm (for the numerical solution of these two problems). The number of independent decision variables (and control intervals) is the same in the two dynamic optimization problems. However, there are more differential equations and more algebraic equations in the thermal model. The dynamic optimization of the isothermal strategy is more than twice as fast as the dynamic optimization of the thermal strategy. One reason for this is that the NLP solver uses 15 iterations for the thermal strategy and 6 iterations for the isothermal

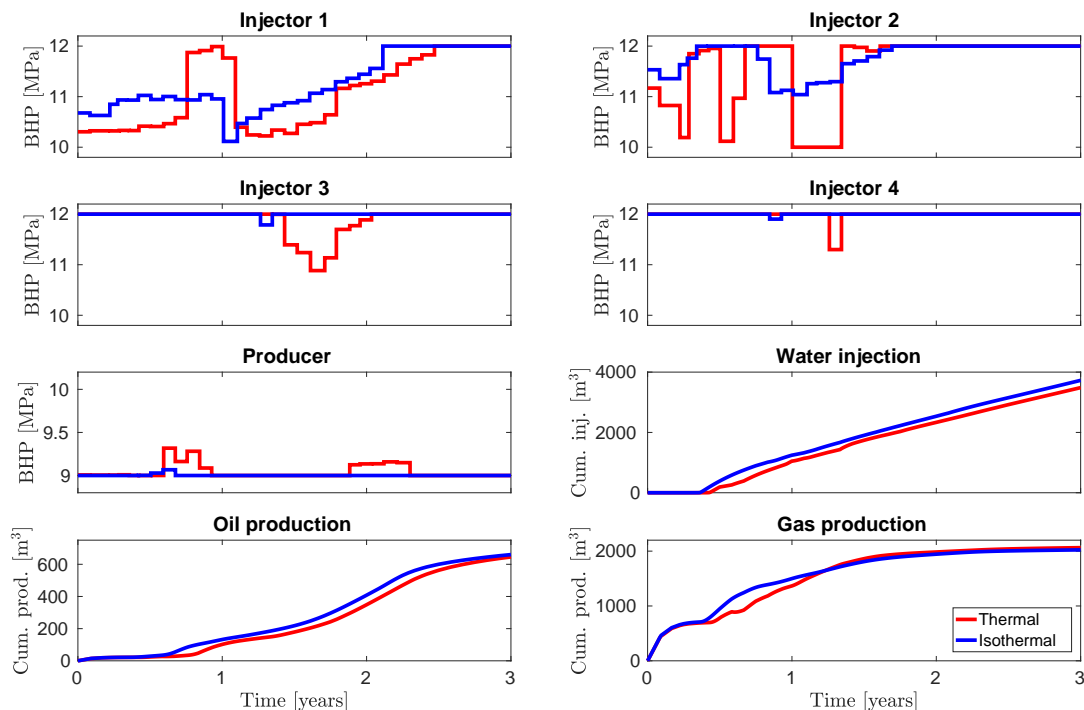


Fig. 10.2: The optimized injector and producer BHPs and the cumulative volume of the injected water, the produced oil, and the produced gas. This figure originally appeared in [145].

strategy. However, each iteration of the NLP solver requires, on average, more numerical simulations (i.e. numerical solutions of IVPs) in the optimization of the isothermal strategy. Consequently, the computation time per iteration of the NLP solver is almost the same for the two problems.

The numerical simulation of the thermal and the isothermal waterflooding strategies require almost the same number of time steps. However, the numerical simulations of the thermal strategy require roughly one more Newton iteration than the numerical simulations of the isothermal strategy. Therefore, the number of evaluations of F and G (in the semi-explicit DAEs (1.1) which constitute the reservoir flow models) and their Jacobians is also close to one higher. Finally, the GMRES method requires 0.8 more iterations on average to solve the linear systems in the numerical simulations of the thermal strategy.

The number of iterations of the NLP solver and the GMRES method may be specific to these two particular examples. However, the difference in the number of Newton iterations may be due to the additional energy balance equation in the thermal reservoir flow model, e.g. because of the different scales of 1) the mass and energy equations and 2) the internal energy and the total mole numbers. If this is the case, it may be possible to reduce the number of Newton iterations in the numerical simulation of the thermal strategy by rescaling the equations and variables appropriately.

To conclude, the above results suggest that the numerical simulation (and therefore also the dynamic optimization) of thermal waterflooding strategies is more computationally demanding than the numerical simulation (and dynamic optimization) of isothermal strategies.

10.2 Summary

In this chapter, we have presented numerical examples of optimized thermal and isothermal waterflooding strategies. The two strategies are computed using the gradient-based dynamic optimization algorithm described in Chapter 7 and the thermal and isothermal compositional reservoir flow models presented in Chapter 4. The algorithm is implemented in C/C++ using the open-source software DUNE and ThermoLib as well as the commercial software KNITRO 10.2.

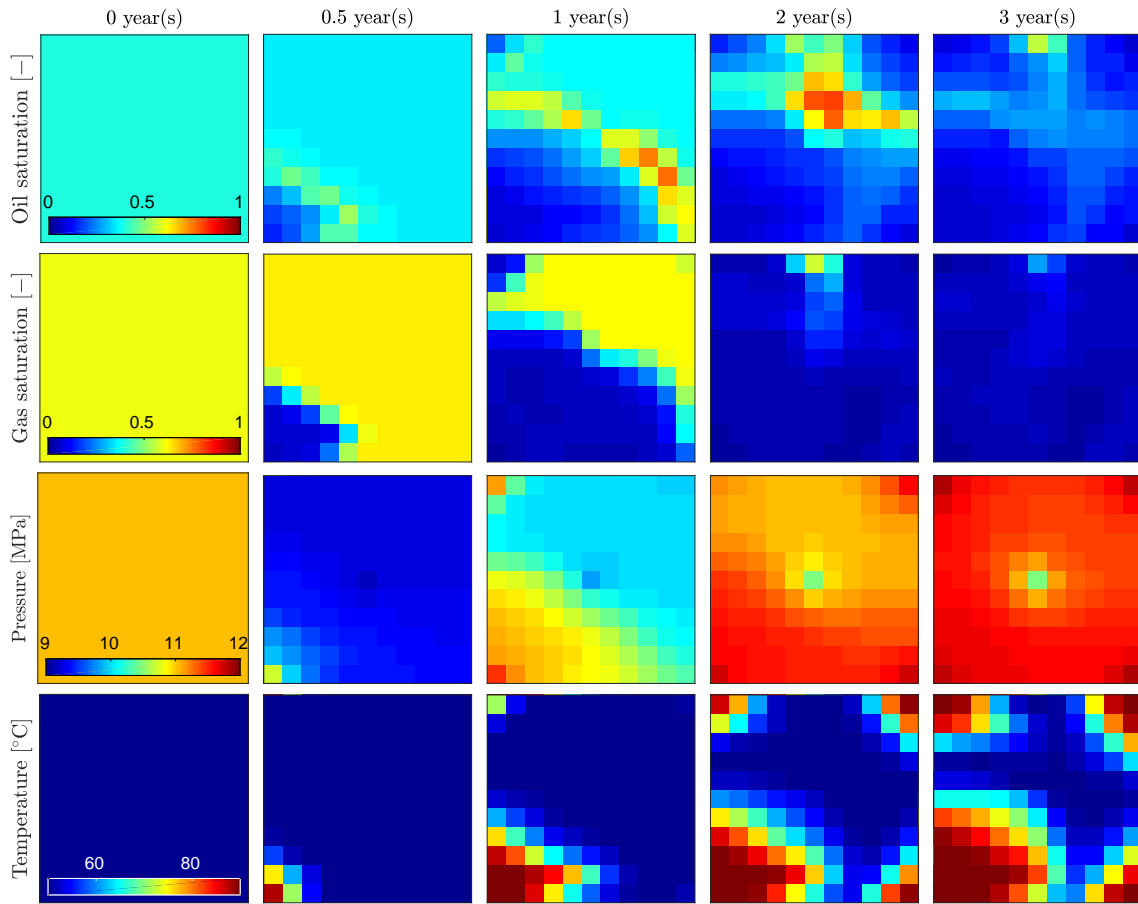


Fig. 10.3: The oil and gas saturations, the pressure, and the temperature for the optimized thermal waterflooding strategy. The temperature of the injected water is 90°C . This figure originally appeared in [145].

We compile the code using GCC compilers. The computation time of the dynamic optimization algorithm is more than two times higher for the thermal strategy than for the isothermal strategy. The reasons are that for the thermal strategy: 1) the NLP solver requires more iterations, 2) the numerical simulations require more Newton iterations (roughly one more iteration on average), and 3) the iterative solutions of the involved linear systems (with the GMRES method) require 0.8 more iterations on average. The differences in the number of iterations of the NLP solver and the GMRES method may be specific to these two examples, whereas the difference in the number of Newton iterations may be due to the additional energy conservation equation in the thermal reservoir flow model.

In conclusion, the dynamic optimization of the thermal strategy is more computationally demanding than the dynamic optimization of the isothermal strategy for the examples considered in this chapter.

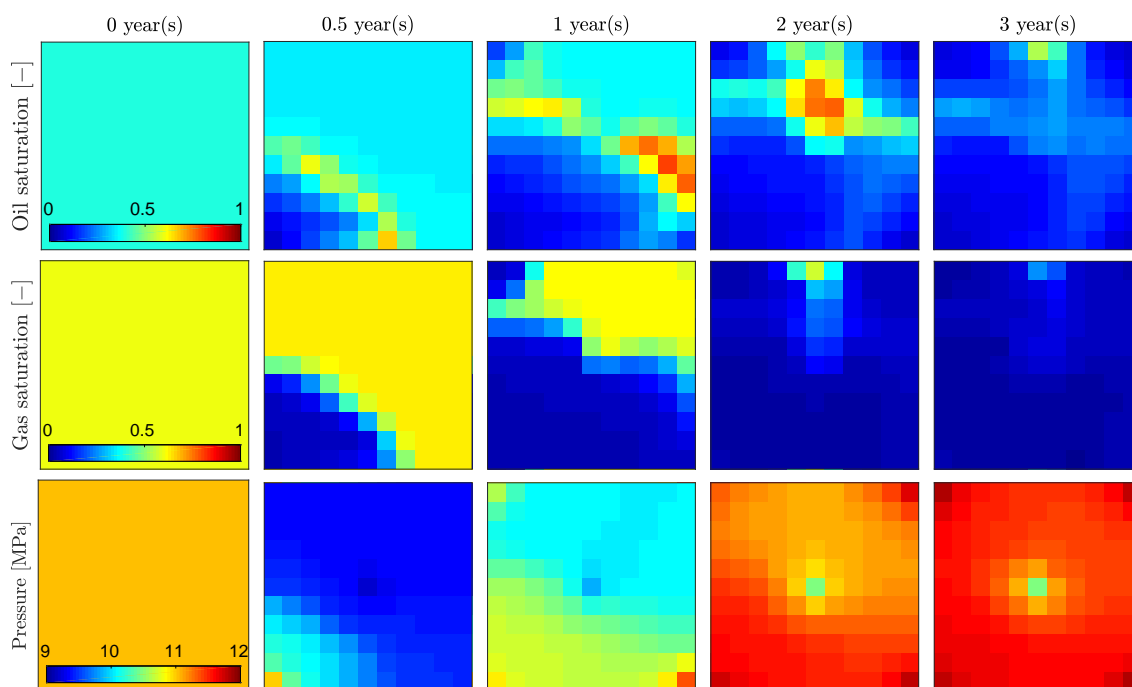


Fig. 10.4: The oil and gas saturations and the pressure for the optimized isothermal waterflooding strategy. The temperature of the injected water is 50°C , i.e. it is the same as the temperature of the reservoir fluid and the reservoir rock. This figure originally appeared in [145].

Table 10.1: Key characteristics of the two dynamic optimization problems and KPIs of the dynamic optimization algorithm. The gradient computations are carried out with the adjoint method. The function evaluations refer to the evaluations of F and G in the semi-explicit DAEs (1.1) which constitute the reservoir flow models. The Jacobian evaluations refer to the evaluations of the Jacobians of F and G . The KPIs related to the numerical solution of IVPs (also referred to as numerical simulations) are averaged over all the numerical solutions (of IVPs) involved in the numerical solution of the dynamic optimization problems. The iterations per linear system refer to the iterations in the iterative GMRES method. This table originally appeared in [145].

	Thermal	Isothermal
Problem		
Independent decision variables	180	180
Control intervals	36	36
Differential equations	847	726
Algebraic equations	2541	2299
Numerical solution of the NLP		
Iterations	15	6
Numerical solutions of IVPs	27	20
Gradient computations	16	7
CPU time (s)	661.16	287.18
CPU time per iteration (s)	44.08	47.86
Numerical solution of the IVPs		
Time steps	218.37	212.55
Newton iterations per time step	4.04	3.02
Function evaluations per time step	6.07	5.07
Jacobian evaluations per time step	5.67	4.41
Iterations per linear system	9.03	8.21

Part IV

Conclusions

Conclusions

In this chapter, we present conclusions of the work presented in this thesis. First, we briefly discuss the main subjects of this work. Thereafter, we discuss each subject and the corresponding conclusions in more detail. Finally, we discuss the perspectives of the present work in terms of its applicability to other processes than the ones considered here.

The work presented in this thesis is concerned with models and algorithms for NMPC of thermal and isothermal compositional reservoir flow processes. The main subjects of this work are

- a) ThermoLib – an open-source thermodynamic software with sensitivity capabilities
- b) state estimation of UV flash processes
- c) dynamic optimization of UV flash processes
- d) NMPC of UV flash processes
- e) dynamic optimization of thermal and isothermal compositional reservoir flow models

In this work, we present thermodynamically rigorous thermal and isothermal compositional reservoir flow models. Thermal and compositional reservoir flow is a UV flash process because the involved phase equilibrium problem is the UV flash. Similarly, isothermal and compositional reservoir flow is a VT flash process. We demonstrate that both the thermal and the isothermal model are in a semi-explicit DAE form, and we formulate algorithms for state estimation, dynamic optimization, and NMPC of such semi-explicit DAE systems. Therefore, these algorithms are applicable to both UV and VT flash processes (as well as other dynamical phase equilibrium processes), although this work is mainly concerned with UV flash processes.

The underlying thermodynamic model is central to the reservoir flow models, and in this work, we develop an open-source thermodynamic software library called ThermoLib which has capabilities for sensitivity computations. Such capabilities are necessary in the state estimation algorithms, the dynamic optimization algorithm, and the NMPC algorithm considered in this work.

We implement and test the algorithms using a small-scale UV flash separation process which is representative of the thermal and compositional reservoir flow process. In particular, the model of this flash separation process is in the same semi-explicit DAE form as the thermal and the isothermal compositional reservoir flow models. Furthermore, we implement and test the dynamic optimization algorithm using the thermal and the isothermal reservoir flow models. However, it is outside the scope of this work to implement the state estimation algorithms and the NMPC algorithm for the two reservoir flow models.

Next, we present conclusions for each of the above five subjects as well as one additional subject. The conclusions for item b) through d) are mainly related to the flash separation process, whereas the conclusions for item e) are related to the thermal and isothermal reservoir flow models (although thermal reservoir flow is also a UV flash process). Furthermore, we discuss the perspectives of the present work.

ThermoLib. In this work, we develop an open-source thermodynamic library called ThermoLib which is available at www.psetools.org. The thermodynamic model in ThermoLib is based on 1) data and correlations from the DIPPR database and 2) cubic equations of state. ThermoLib provides Matlab and C routines for evaluating the enthalpy, entropy, and volume of a mixture as functions of the temperature, pressure, and mixture composition (in moles). All other thermodynamic functions can be computed from the enthalpy, entropy, and volume using fundamental thermodynamic relations. The novelty of ThermoLib is that its routines also evaluate both the first and the second order derivatives of the thermodynamic functions with respect to the temperature, pressure, and composition. We derive the expressions for these derivatives analytically.

State estimation of UV flash processes. We describe four algorithms for state estimation of UV flash processes. State estimation of such processes has not been addressed before. We model the UV flash processes using stochastic semi-explicit DAEs, and we consider the continuous-discrete EKF, UKF, PF, and EnKF. We have implemented the four algorithms in Matlab and presented a numerical example of state estimation of a UV flash separation process. For this example: 1) the EKF is significantly faster but also less accurate than the other filters and 2) the PF and the UKF are the most accurate, but the UKF is significantly faster than the PF.

Dynamic optimization of UV flash processes. We present a novel algorithm for gradient-based numerical solution of dynamic optimization problems that involve UV flash processes. Dynamic optimization of UV flash processes has not been addressed previously. The algorithm is based on 1) the single-shooting method and 2) the discrete adjoint method for the computation of gradients. We model the UV flash processes using deterministic semi-explicit DAEs. The numerical solution of these DAEs (also referred to as a numerical simulation) is central to the dynamic optimization algorithm, and we describe two approaches for this purpose: 1) a simultaneous approach and 2) a nested approach. The differential equations and the algebraic equations are solved simultaneously in the simultaneous approach, and the solution of the algebraic equations is nested into the solution of the differential equations in the nested approach.

We implement the dynamic optimization algorithm in Matlab and C, and we present two numerical examples of dynamic optimization of a UV flash separation process. Furthermore, we present a performance study based on the numerical solution of these two dynamic optimization problems. In the performance study, we compare the computational efficiency of using 1) the simultaneous and the nested approach (in the dynamic optimization algorithm), 2) exact and inexact Newton methods (in the simultaneous and the nested approaches), 3) Matlab's `fmincon`, `IPOPT`, `KNITRO`, and `NPSOL`, and 4) GCC compilers together with Netlib's implementation of BLAS/LAPACK (a linear algebra software) and Intel compilers together with Intel's MKL (also a linear algebra software). The performance study demonstrates that 1) the simultaneous approach is faster than the nested approach for both problems, 2) an exact Newton method results in better performance for the nested approach, and for the simultaneous approach, an inexact Newton method is faster for one of the problems, 3) the C implementations of the algorithm are faster than the implementations using Matlab (including a mixed Matlab and C implementation), and 4) the best computational performance is obtained with `NPSOL`, Intel compilers, and Intel's MKL.

NMPC of UV flash processes. We describe an NMPC algorithm based on 1) either of the four state estimation algorithms and 2) the gradient-based dynamic optimization algorithm. NMPC of UV flash processes has not been addressed before. We present a numerical example of economical NMPC for disturbance rejection in a UV flash separation process. In this example, we consider a mixed Matlab and C implementation of the NMPC algorithm which uses the EKF to estimate the states as well as an unknown disturbance. Furthermore, we discuss the computational performance of the algorithm for this example.

Dynamic optimization of reservoir flow models. We consider dynamic optimization of novel and thermodynamically rigorous models of thermal and isothermal compositional reservoir flow. Thermal and compositional reservoir flow models that involve the UV flash have not been presented previously. Isothermal and compositional reservoir flow models based on the VT flash have been presented before. However, the VT flash has not been formulated as an optimization problem in these previous models. Furthermore, dynamic optimization of thermal and isothermal compositional reservoir flow models based on the UV and the VT flash has not been addressed before.

We present numerical examples of optimized thermal and isothermal waterflooding strategies computed using the gradient-based dynamic optimization algorithm mentioned above and the two reservoir flow models. Furthermore, we compare the computational performance of the algorithm for these two examples. This comparison shows that the numerical simulation (and therefore also the dynamic optimization) of the thermal reservoir flow model is more computationally demanding than the numerical simulation (and dynamic optimization) of the isothermal reservoir flow model.

Relations between the UV flash, the VT flash, and the isofugacity condition. As mentioned above, the phase equilibrium problems that are relevant to the thermal and isothermal compositional reservoir flow processes considered in this work are the UV and the VT flash. We formulate the UV and the VT flash as optimization problems and the corresponding phase equilibrium conditions as the first order optimality conditions. However, it is common in the reservoir simulation and optimization literature to formulate the phase equilibrium conditions as the isofugacity condition. The isofugacity condition is derived from the PT flash which can also be formulated as an optimization problem.¹ In the paper in Appendix G, we show that the UV flash optimization problem is equivalent to the combination of 1) the PT flash optimization problem and 2) constraints on the internal energy and the volume. Similarly, in the paper in Appendix L, we show that the VT flash optimization problem is equivalent to the combination of the PT flash optimization problem and a constraint on the volume.

In conclusion, the commonly used isofugacity condition can be derived from the UV and the VT flash.

Perspectives. In this work, we consider three particular phase equilibrium processes: 1) a thermal and compositional reservoir flow process, 2) an isothermal and compositional reservoir flow process, and 3) a flash separation process. As mentioned previously, the models of these processes are comprised of DAEs in the semi-explicit form (1.1). It is natural to model the dynamical behavior of other phase equilibrium processes with DAEs in this form as well, including fluid vessels, distillation columns, two-phase flow in pipes (e.g. in pipelines), CO₂ sequestration, and magmatic hydrothermal flow. Therefore, the algorithms described in this work can be used for 1) simulation, 2) state estimation, 3) dynamic optimization, and 4) NMPC of such processes as well.

¹In the paper in Appendix H and in the technical report in Appendix P, we present 1) expressions for the fugacities based on cubic equations of state and 2) the derivation of the isofugacity condition from the PT flash optimization problem.

Suggestions for future work

In this chapter, we present suggestions for future work based on the work presented in this thesis. The first suggestion is to implement and test the state estimation algorithms and the NMPC algorithm (described in this thesis) for the thermal and the isothermal compositional reservoir flow models (also described in this thesis). The remaining suggestions can be categorized as

- a) modeling of phase appearance and disappearance
- b) application of algorithms to other phase equilibrium processes
- c) general improvements of the computational efficiency of the algorithms

For some phase equilibrium processes (including oil recovery processes), it is relevant to model the appearance or disappearance of phases in order to accurately predict the dynamical behavior of the process. We do not model such appearance and disappearance of phases in this work, i.e. we assume that all phases are nonempty at all times. Furthermore, the algorithms described in this thesis can be applied to other phase equilibrium processes than the waterflooding processes (and the flash separation process) that we consider, e.g. other oil recovery processes. Finally, the computational efficiency of the algorithms that we describe can be improved, e.g. by considering more advanced algorithms for iterative solution of linear systems, numerical simulation, numerical solution of NLPs, gradient computations, etc. Next, we present the specific suggestions and discuss them in detail.

State estimation and NMPC of the reservoir flow models. As mentioned above, the first suggestion is to implement state estimation and NMPC of the thermal and the isothermal compositional reservoir flow models presented in this thesis. We suggest to implement the EnKF for these models because it is computationally tractable for large-scale systems.

In this work, we implement the dynamic optimization algorithm (described in this thesis) in C/C++ and test it for the computation of optimized thermal and isothermal waterflooding strategies. Therefore, we also develop C/C++ code for numerical simulation of the thermal and the isothermal reservoir flow models. This code can be reused in the implementation of state estimation and NMPC of the two reservoir flow models.

Modeling of the appearance and disappearance of phases. The models presented in this work are only valid when all the involved phases are nonempty. However, the appearance and disappearance of phases is common in oil recovery processes (and in other phase equilibrium processes). For instance, the initial pressure in a reservoir can be so high that the gas is completely condensed, and only the liquid oil phase is present. When the reservoir fluid is produced, the pressure drops, and a gas phase may appear. Furthermore, in order to model steam injection processes, it is necessary to model the appearance of the steam phase because it is not initially present in the reservoir.

In this work, we formulate the phase equilibrium problems as optimization problems in the

form (1.2). The decision variables in the optimization problem are the algebraic variables, y , which include composition vectors (i.e. mole numbers) and possibly also the temperature and pressure (depending on the specific phase equilibrium problem). All of these quantities must be nonnegative.¹ This condition can be incorporated explicitly into the phase equilibrium optimization problems:

$$\min_y f(y), \quad (12.1a)$$

$$\text{subject to } g(y) = x, \quad (12.1b)$$

$$h(y) = 0, \quad (12.1c)$$

$$y \geq 0. \quad (12.1d)$$

The only difference between the form (1.2) considered in this work and (12.1) is the inequality constraint (12.1d). The phase equilibrium conditions are the first order optimality conditions of (12.1). Because of the additional inequality constraint, the first order optimality conditions contain both inequalities and complementarity conditions [115] which will be part of the model equations. Consequently, the dynamic optimization problem (which determines the optimal open-loop control strategy) will contain complementarity constraints. Therefore, its numerical solution may require specialized algorithms [13, 19, 137]. Furthermore, the state estimation problem will also involve inequalities. Therefore, algorithms for constrained state estimation are necessary. There exist versions of the state estimation algorithms considered in this work for constrained state estimation [90, 123, 126, 127, 167, 171, 173, 193]. The state estimation problem can also be formulated as a dynamic optimization problem [20] and solved using dynamic optimization algorithms (as in moving horizon estimation [3]).

An alternative approach is to formulate the phase equilibrium conditions as a set of nonsmooth algebraic equations [150, 186]. The numerical solution (and the associated sensitivity computations) of such nonsmooth equations also requires specialized algorithms [10, 163].

Application to steam injection. In this work, we present a model of thermal and compositional waterflooding. However, thermal (enhanced) oil recovery strategies more often involve steam injection or steam-assisted gravity drainage (SAGD) [4, 168]. It is straightforward to extend the phase equilibrium problem in the thermal model presented in this work (i.e. the UV flash) to incorporate a steam phase, and the thermodynamic properties of the steam phase may be computed using the model presented in Section 2.3. Additionally, it is necessary to describe the viscosity and the relative permeability of the steam phase.

Application to distillation columns. In the paper in Appendix F, the dynamic optimization algorithm described in this thesis is used to compute an open-loop optimal control strategy which optimizes the economics of a cryogenic air separation (CAS) process, i.e. of a cryogenic distillation column. Distillation columns involve a number of stages, and in the paper, the phase equilibrium problem for each stage is the UV flash formulated as an optimization problem. This work can be extended to state estimation and NMPC of distillation columns (i.e. for other distillation processes as well).

General improvements of the computational efficiency of the algorithms. Here, we provide a number of suggestions for improving the computational efficiency of the algorithms described in this thesis.

First, in the dynamic optimization algorithm that we describe, we use off-the-shelf optimization software (i.e. NLP solvers) for the numerical solution of the involved NLPs. It may be possible to improve the computational efficiency by implementing a customized optimization algorithm, e.g.

¹Note that mole fractions commonly appear in thermodynamic models (this is also the case for the thermodynamic models described in this thesis). In such cases, the thermodynamic functions are undefined if a phase is completely empty because the expressions for the mole fractions involve division by the total amount of moles in the phase. Consequently, if the phase is empty, a division by zero occurs.

an SQP method [115, Chap. 18]. For instance, it can ease the handling of output constraints, i.e. constraints that involve the dependent decision variables in the dynamic optimization problem.

Next, in this work, we use Euler's implicit method for the numerical solution of the semi-explicit DAEs. However, it can be more efficient to use higher-order ESDIRK methods [31–33, 178, 179]. Furthermore, we use a discrete adjoint method for the gradient computations. It can be more efficient to use a continuous adjoint method [32, 33].

Finally, in the C/C++ implementation of the dynamic optimization algorithm for the thermal and isothermal reservoir flow models, it is necessary to solve large sparse linear systems of equations. The linear systems arise in 1) the numerical solution of the semi-explicit DAEs (i.e. in the involved Newton iterations) and 2) the adjoint equations. We use a GMRES method with a block ILU preconditioner to solve these linear systems. However, constrained pressure residual (CPR) preconditioners may outperform ILU preconditioners [182, 183, 185]. The linear systems in the Newton iterations involve the Jacobian matrices, whereas the linear systems in the adjoint equations involve the transposed Jacobian matrices. Consequently, CPR preconditioners must be formulated specifically for the adjoint equations [68].

Bibliography

- [1] Aanonsen, S. I., Nævdal, G., Oliver, D. S., Reynolds, A. C., and Vallès, B. (2009). The ensemble Kalman filter in reservoir engineering – a review. *SPE Journal*, 14(3):393–412.
- [2] AAPG Wiki (2016). Reservoir simulation study. http://wiki.aapg.org/Reservoir_simulation_study. Accessed: July 1st 2018.
- [3] Alessandri, A., Baglietto, M., Battistelli, G., and Zavala, V. (2010). Advances in moving horizon estimation for nonlinear systems. In *Proceedings of the 49th IEEE Conference on Decision and Control*, Atlanta, Georgia, USA.
- [4] Alvarado, V. and Manrique, E. (2010). Enhanced oil recovery: an update review. *Energies*, 3:1529–1575.
- [5] Arendsen, A. R. J. and Versteeg, G. F. (2009). Dynamic thermodynamics with internal energy, volume, and amount of moles as states: application to liquefied gas tank. *Industrial & Engineering Chemistry Research*, 48(6):3167–3176.
- [6] AwesomeThermodynamics (2016). www.github.com/iurisegtovich/AwesomeThermodynamics. Accessed: June 7th 2018.
- [7] Barros, E. G. D., Jansen, J. D., and Van den Hof, P. M. J. (2015a). Value of information in parameter identification and optimization of hydrocarbon reservoirs. *IFAC-PapersOnLine*, 48(6):229–235.
- [8] Barros, E. G. D., Leeuwenburgh, O., Van den Hof, P. M. J., and Jansen, J. D. (2015b). Value of multiple production measurements and water front tracking in closed-loop reservoir management. In *Proceedings of the 2015 SPE Reservoir Characterisation and Simulation Conference and Exhibition*, Abu Dhabi, United Arab Emirates.
- [9] Barros, E. G. D., Van den Hof, P. M. J., and Jansen, J. D. (2015c). Value of information in closed-loop reservoir management. *Computational Geosciences*, 20(3):737–749.
- [10] Barton, P. I. and Lee, C. K. (2002). Modeling, simulation, sensitivity analysis, and optimization of hybrid systems. *ACM Transactions on Modeling and Computer Simulation*, 12(4):256–289.
- [11] Bastian, P., Blatt, M., Dedner, A., Engwer, C., Klöfkorn, R., Kornhuber, R., Ohlberger, M., and Sander, O. (2008a). A generic grid interface for parallel and adaptive scientific computing. Part II: implementation and tests in DUNE. *Computing*, 82(2-3):121–138.
- [12] Bastian, P., Blatt, M., Dedner, A., Engwer, C., Klöfkorn, R., Ohlberger, M., and Sander, O. (2008b). A generic grid interface for parallel and adaptive scientific computing. Part I: abstract framework. *Computing*, 82(2-3):103–119.
- [13] Baumrucker, B. T., Renfro, J. G., and Biegler, L. T. (2008). MPEC problem formulations and solution strategies with chemical engineering applications. *Computers & Chemical Engineering*, 32(12):2903–2913.
- [14] Becerra, V. M., Roberts, P. D., and Griffiths, G. W. (2001). Applying the extended Kalman filter to systems described by nonlinear differential-algebraic equations. *Control Engineering Practice*, 9(3):267–281.
- [15] Bell, I. H., Wronski, J., Quoilin, S., and Lemort, V. (2014). Pure and pseudo-pure fluid thermophysical property evaluation and the open-source thermophysical property library CoolProp.

- Industrial & Engineering Chemistry Research*, 53(6):2498–2508.
- [16] Benjelloun-Dabaghi, Z., de Hemptinne, J.-C., Jarrin, J., Leroy, J.-M., Aubry, J.-C., Saas, J. N., and Taravel-Condat, C. (2002). MOLDITM: a fluid permeation model to calculate the annulus composition in flexible pipes. *Oil & Gas Science and Technology - Rev. IFP*, 57(2):177–192.
- [17] Bhatnagar, S., Prasad, H. L., and Prashanth, L. A. (2013). *Stochastic recursive algorithms for optimization*, volume 434 of *Lecture Notes in Control and Information Sciences*. Springer-Verlag London.
- [18] Biegler, L. T. (2007). An overview of simultaneous strategies for dynamic optimization. *Chemical Engineering and Processing: Process Intensification*, 46(11):1043–1053.
- [19] Biegler, L. T. (2010). *Nonlinear programming: concepts, algorithms, and applications to chemical processes*. SIAM.
- [20] Binder, T., Blank, L., Bock, H. G., Bulirsch, R., Dahmen, W., Diehl, M., Kronseder, T., Marquardt, W., Schlöder, J. P., and von Stryk, O. (2001). Introduction to model based optimization of chemical processes on moving horizons. In *Online Optimization of Large Scale Systems*, pages 295–339. Springer-Verlag Berlin Heidelberg.
- [21] Blatt, M. and Bastian, P. (2007). The iterative solver template library. In B. Kågström et al., editor, *Applied Parallel Computing. State of the Art in Scientific Computing. PARA 2006*, volume 4699 of *Lecture Notes in Computer Science*, pages 666–675. Springer, Berlin, Heidelberg.
- [22] Bock, H. G. and Plitt, K. J. (1984). A multiple shooting algorithm for direct solution of optimal control problems. *IFAC Proceedings Volumes*, 17(2):1603–1608.
- [23] BP (2017). Statistical review of world energy. <https://www.bp.com/content/dam/bp/en/corporate/pdf/energy-economics/statistical-review-2017/bp-statistical-review-of-world-energy-2017-full-report.pdf>. Accessed: August 6th 2018.
- [24] Brouwer, D. R., Nævdal, G., Jansen, J. D., Vefring, E. H., and van Kruijsdijk, C. P. J. W. (2004). Improved reservoir management through optimal control and continuous model updating. In *Proceedings of the 2004 SPE Annual Technical Conference and Exhibition*, Houston, Texas, USA.
- [25] Bukshtynov, V., Volkov, O., Durlafsky, L. J., and Aziz, K. (2015). Comprehensive framework for gradient-based optimization in closed-loop reservoir management. *Computational Geosciences*, 19(4):877–897.
- [26] Burke, E. K. and Kendall, G. (2014). *Search methodologies: introductory tutorials in optimization and decision support techniques*. Springer Science and Business Media New York, 2nd edition.
- [27] Cabral, V. F., Castier, M., and Tavares, F. W. (2005). Thermodynamic equilibrium in systems with multiple adsorbed and bulk phases. *Chemical Engineering Science*, 60(6):1773–1782.
- [28] Callen, H. B. (1985). *Thermodynamics and an introduction to thermostatistics*. John Wiley & Sons, 2nd edition.
- [29] Cao, Y., Li, S., Petzold, L., and Serban, R. (2003). Adjoint sensitivity analysis for differential-algebraic equations: the adjoint DAE system and its numerical solution. *SIAM Journal on Scientific Computing*, 24(3):1076–1089.
- [30] Capolei, A. (2013). *Nonlinear model predictive control for oil reservoirs management*. PhD thesis, Technical University of Denmark.
- [31] Capolei, A. and Jørgensen, J. B. (2012). Solution of constrained optimal control problems using multiple shooting and ESDIRK methods. In *Proceedings of the 2012 American Control Conference*, pages 295–300, Fairmont Queen Elizabeth, Montréal, Canada.
- [32] Capolei, A., Stenby, E. H., and Jørgensen, J. B. (2012). High order adjoint derivatives using ESDIRK methods for oil reservoir production optimization. In *Proceedings of the 13th European Conference on the Mathematics of Oil Recovery*, Biarritz, France.
- [33] Capolei, A., Völcker, C., Frydendall, J., and Jørgensen, J. B. (2012). Oil reservoir production optimization using single shooting and ESDIRK methods. *IFAC Proceedings Volumes*, 45(8):286–291.

- [34] Castier, M. (2009). Solution of the isochoric-isoenergetic flash problem by direct entropy maximization. *Fluid Phase Equilibria*, 276(1):7–17.
- [35] Castier, M. (2010). Dynamic simulation of fluids in vessels via entropy maximization. *Journal of Industrial and Engineering Chemistry*, 16(1):122–129.
- [36] Castier, M. and Tavares, F. W. (2005). Centrifugation equilibrium of natural gas. *Chemical Engineering Science*, 60(11):2927–2935.
- [37] Chapman, J. W., Lavelle, T. M., May, R. D., Litt, J. S., and Guo, T.-H. (2014). Toolbox for the modeling and analysis of thermodynamic systems (T-MATS) user’s guide. Technical report, NASA’s Glenn Research Center.
- [38] Chavent, G., Dupuy, M., and Lemonnier, P. (1975). History matching by use of optimal theory. *SPE Journal*, 15(1):74–86.
- [39] Chen, Y. and Oliver, D. S. (2009). Ensemble-based closed-loop optimization applied to Brugge field. In *Proceedings of the 2009 SPE Reservoir Simulation Symposium*, The Woodlands, Texas, USA.
- [40] Chen, Y. and Oliver, D. S. (2010). Ensemble-based closed-loop optimization applied to Brugge field. *SPE Reservoir Evaluation & Engineering*, 13(1):56–71.
- [41] Chen, Y., Oliver, D. S., and Zhang, D. (2008). Efficient ensemble-based closed-loop production optimization. In *Proceedings of the 2008 SPE Improved Oil Recovery Symposium*, Tulsa, Oklahoma, USA.
- [42] Chen, Z. (2007). *Reservoir simulation: mathematical techniques in oil recovery*. SIAM.
- [43] Chen, Z., Henson, M. A., Belanger, P., and Megan, L. (2010). Nonlinear model predictive control of high purity distillation columns for cryogenic air separation. *IEEE Transactions on Control Systems Technology*, 18(4):811–821.
- [44] Chen, Z., Huan, G., and Ma, Y. (2006). *Computational methods for multiphase flows in porous media*. Computational Science & Engineering. SIAM.
- [45] Cudas, A., Hanssen, K. G., Foss, B., Capolei, A., and Jørgensen, J. B. (2017). Multiple shooting applied to robust reservoir control optimization including output constraints on coherent risk measures. *Computational Geosciences*, 21(3):479–497.
- [46] Colson, B., Marcotte, P., and Savard, G. (2007). An overview of bilevel optimization. *Annals of Operations Research*, 153(1):235–256.
- [47] Delshad, M. and Pope, G. A. (1989). Comparison of the three-phase oil relative permeability models. *Transport in Porous Media*, 4(1):59–83.
- [48] Do, S. T. and Reynolds, A. C. (2013). Theoretical connections between optimization algorithms based on an approximate gradient. *Computational Geosciences*, 17(6):959–973.
- [49] Douc, R., Cappé, O., and Moulines, E. (2005). Comparison of resampling schemes for particle filtering. In *Proceedings of the 4th International Symposium on Image and Signal Processing and Analysis*, pages 64–69, Zagreb, Croatia.
- [50] DWSIM (2018). dwsim.inforst.com.br. Accessed: June 7th 2018.
- [51] Eppelbaum, L., Kutasov, I., and Pilchin, A. (2014). *Applied geothermics*. Lecture Notes in Earth System Sciences. Springer-Verlag Berlin Heidelberg.
- [52] Espósito, R. O., Castier, M., and Tavares, F. W. (2000). Calculations of thermodynamic equilibrium in systems subject to gravitational fields. *Chemical Engineering Science*, 55(17):3495–3504.
- [53] Eurostat (2017). Oil and petroleum products - a statistical overview. ec.europa.eu/eurostat/statistics-explained/index.php/Oil_and_petroleum_products_-_a_statistical_overview. Accessed: May 9th 2018.
- [54] Evensen, G. (2003). The ensemble Kalman filter: theoretical formulation and practical implementation. *Ocean Dynamics*, 53(4):343–367.
- [55] Evensen, G. (2009a). *Data assimilation: the ensemble Kalman filter*. Springer-Verlag Berlin Heidelberg, 2nd edition.
- [56] Evensen, G. (2009b). The ensemble Kalman filter for combined state and parameter estimation.

- IEEE Control Systems Magazine*, 29(3):83–104.
- [57] Flatby, P., Skogestad, S., and Lundström, P. (1994). Rigorous dynamic simulation of distillation columns based on UV-flash. *IFAC Proceedings Volumes*, 27(2):261–266.
- [58] Forbes, M. G., Patwardhan, R. S., Hamadah, H., and Gopaluni, R. B. (2015). Model predictive control in industry: challenges and opportunities. *IFAC-PapersOnLine*, 48(8):531–538.
- [59] Forouzanfar, F., Rossa, E. D., Russo, R., and Reynolds, A. C. (2013). Life-cycle production optimization of an oil field with an adjoint-based gradient approach. *Journal of Petroleum Science and Engineering*, 112:351–358.
- [60] Foss, B. and Jensen, J. P. (2011). Performance analysis for closed-loop reservoir management. *SPE Journal*, 16(1):183–190.
- [61] Garipov, T. T., Tomin, P., Rin, R., Voskov, D. V., and Tchelepi, H. A. (2018). Unified thermo-compositional-mechanical framework for reservoir simulation. *Computational Geosciences*, 22(4):1039–1057.
- [62] Gildin, E. and Wheeler, M. F. (2008). Control of subsurface flow using model predictive control techniques. In *Proceedings of the 2008 International Conference on Engineering Optimization*, pages 1–10, Rio de Janeiro, Brazil.
- [63] Gillijns, S., Mendoza, O. B., Chandrasekar, J., De Moor, B. L. R., Bernstein, D. S., and Ridley, A. (2006). What is the ensemble Kalman filter and how well does it work? In *Proceedings of the 2006 American Control Conference*, pages 4448–4453, Minneapolis, Minnesota, USA.
- [64] Gmehling, J., Kolbe, B., Kleiber, M., and Rarey, J. (2012). *Chemical thermodynamics for process simulation*. Wiley-VCH.
- [65] Grema, A. S. and Cao, Y. (2017). Receding horizon control for oil reservoir waterflooding process. *Systems Science & Control Engineering*, 5(1):449–461.
- [66] Grema, A. S., Popoola, L. T., Idriss, I. M., and Atiku, K. G. (2016). Application of nonlinear model predictive controller (NMPC) to oil reservoir waterflooding. *International Journal of Modeling and Simulation for the Petroleum Industry*, 9(2):41–46.
- [67] Guevara, J. L., Ortega, A., Canelón, J. I., Nava, E., and Queipo, N. V. (2015). Model-based adaptive-predictive control and optimization of SAGD under uncertainty. In *Proceedings of the 2015 SPE Latin American and Caribbean Petroleum Engineering Conference*, Quito, Ecuador.
- [68] Han, C., Wallis, J., Sarma, P., Li, G., Schrader, M. L., and Chen, W. (2013). Adaptation of the CPR preconditioner for efficient solution of the adjoint equation. *SPE Journal*, 18(2):207–213.
- [69] Haßkerl, D., Arshad, M., Hashemi, R., Subramanian, S., and Engell, S. (2016). Simulation study of the particle filter and the EKF for state estimation of a large-scale DAE-system with multi-rate sampling. *IFAC-PapersOnLine*, 49(7):490–495.
- [70] Haßkerl, D., Subramanian, S., Hashemi, R., Arshad, M., and Engell, S. (2017). State estimation using a multi-rate particle filter for a reactive distillation column described by a DAE model. In *Proceedings of the 25th Mediterranean Conference on Control and Automation*, pages 876–881, Valletta, Malta.
- [71] Heidari, L., Gervais, V., Ravalec, M. L., and Wackernagel, H. (2011). History matching of reservoir models by ensemble Kalman filtering: the state of the art and a sensitivity study. In Ma, Y. Z. and La Pointe, P. R., editors, *Uncertainty analysis and reservoir modeling: AAPG Memoir*, volume 96, chapter 16, pages 249–264. AAPG.
- [72] Heirung, T. A. N., Wartmann, M. R., Jansen, J. D., Ydstie, B. E., and Foss, B. A. (2011). Optimization of the water-flooding process in a small 2D horizontal oil reservoir by direct transcription. *IFAC Proceedings Volumes*, 44(1):10863–10868.
- [73] Hol, J. D., Schön, T. B., and Gustafsson, F. (2006). On resampling algorithms for particle filters. In *Proceedings of the 2006 IEEE Nonlinear Statistical Signal Processing Workshop*, pages 79–82, Cambridge, UK.
- [74] Holman, J. P. (2010). *Heat transfer*. McGraw-Hill, 10th edition.
- [75] Hou, J., Zhou, K., Zhang, X.-S., Kang, X.-D., and Xie, H. (2015). A review of closed-loop reservoir management. *Petroleum Science*, 12(1):114–128.
- [76] Ibragimov, I. I., Markovinović, R., Ermolaev, A. I., and Naevdal, G. (2009). Production opti-

- mization of thin-oil rims: evaluation of a stochastic steepest-ascent approach. *IFAC Proceedings Volumes*, 42(4):1079–1084.
- [77] Jansen, J. D. (2011). Adjoint-based optimization of multi-phase flow through porous media - a review. *Computers & Fluids*, 46(1):40–51.
- [78] Jansen, J.-D., Bosgra, O. H., and Van den Hof, P. M. J. (2008). Model-based control of multiphase flow in subsurface oil reservoirs. *Journal of Process Control*, 18(9):846–855.
- [79] Jansen, J.-D., Brouwer, D. R., Nævdal, G., and van Kruijsdijk, C. (2005). Closed-loop reservoir management. *First Break*, 23(1):43–48.
- [80] Jansen, J. D., Douma, S. D., Brouwer, D. R., Van den Hof, P. M. J., Bosgra, O. H., and Heemink, A. W. (2009). Closed-loop reservoir management. In *Proceedings of the 2009 SPE Reservoir Simulation Symposium*, The Woodlands, Texas, USA.
- [81] Jindrová, T. and Mikyška, J. (2015). Phase equilibria calculation of CO₂-H₂O system at given volume, temperature, and moles in CO₂ sequestration. *IAENG International Journal of Applied Mathematics*, 45(3):20–29.
- [82] Jørgensen, J. B. (2007). Adjoint sensitivity results for predictive control, state- and parameter-estimation with nonlinear models. In *Proceedings of the 2007 European Control Conference*, pages 3649–3656.
- [83] Jørgensen, J. B., Kristensen, M. R., Thomsen, P. G., and Madsen, H. (2007). New extended Kalman filter algorithms for stochastic differential algebraic equations. In *Assessment and Future Directions of Nonlinear Model Predictive Control*, volume 358 of *Lecture Notes in Control and Information Sciences*, pages 359–366. Springer-Verlag Berlin Heidelberg.
- [84] Jossi, J. A., Stiel, L. I., and Thodos, G. (1962). The viscosity of pure substances in the dense gaseous and liquid phases. *AIChE Journal*, 8(1):59–63.
- [85] Julier, S., Uhlmann, J., and Durrant-Whyte, H. F. (2000). A new method for the nonlinear transformation of means and covariances in filters and estimators. *IEEE Transactions on Automatic Control*, 45(3):477–482.
- [86] Julier, S. J. (2002). The scaled unscented transformation. In *Proceedings of the 2002 American Control Conference*, Anchorage, Alaska, USA.
- [87] Julier, S. J. and Uhlmann, J. K. (2004). Unscented filtering and nonlinear estimation. *Proceedings of the IEEE*, 92(3):401–422.
- [88] Jung, S., Lee, K., Park, C., and Choe, J. (2018). Ensemble-based data assimilation in reservoir characterization: a review. *Energies*, 11(2).
- [89] Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82(1):35–45.
- [90] Kandepu, R., Imsland, L., and Foss, B. A. (2008). Constrained state estimation using the unscented Kalman filter. In *Proceedings of the 16th Mediterranean Conference on Control and Automation*, pages 1453–1458, Ajaccio, France.
- [91] Koretsky, M. D. (2013). *Engineering and chemical thermodynamics*. Wiley, 2nd edition.
- [92] Kou, J., Sun, S., and Wang, X. (2016). An energy stable evolution method for simulating two-phase equilibria of multi-component fluids at constant moles, volume and temperature. *Computational Geosciences*, 20(1):283–295.
- [93] Kourounis, D., Durlofsky, L. J., Jansen, J. D., and Aziz, K. (2014). Adjoint formulation and constraint handling for gradient-based optimization of compositional reservoir flow. *Computational Geosciences*, 18(2):117–137.
- [94] Kourounis, D., Voskov, D., and Aziz, K. (2010). Adjoint methods for multicomponent flow simulation. In *Proceedings of the 12th European Conference on the Mathematics of Oil Recovery*, Oxford, UK.
- [95] Kristensen, M. R., Jørgensen, J. B., Thomsen, P. G., and Jørgensen, S. B. (2004). An ESDIRK method with sensitivity analysis capabilities. *Computers & Chemical Engineering*, 28(12):2695–2707.
- [96] Kristensen, M. R., Jørgensen, J. B., Thomsen, P. G., Michelsen, M. L., and Jørgensen, S. B. (2005). Sensitivity analysis in index-1 differential algebraic equations by ESDIRK methods. *IFAC*

- Proceedings Volumes*, 38(1):212–217.
- [97] Kumar, A. S. and Ahmad, Z. (2012). Model predictive control (MPC) and its current issues in chemical engineering. *Chemical Engineering Communications*, 199(4):472–511.
- [98] Lei, Y., Li, S., Zhang, X., and Zhang, Q. (2012a). Optimal control of polymer flooding for high temperature and high salinity reservoir. *International Journal of Advancements in Computing Technology*, 4(12):52–60.
- [99] Lei, Y., Li, S., Zhang, X., Zhang, Q., and Guo, L. (2012b). Optimal control of polymer flooding based on maximum principle. *Journal of Applied Mathematics*, 2012.
- [100] Li, R., Reynolds, A. C., and Oliver, D. S. (2003). History matching of three-phase flow production data. *SPE Journal*, 8(4):328–340.
- [101] Lie, K.-A. (2014). *An introduction to reservoir simulation using MATLAB*. Sintef ICT, Oslo, Norway.
- [102] Lima, E. R. A., Castier, M., and Biscaia Jr., E. C. (2008). Differential-algebraic approach to dynamic simulations of flash drums with rigorous evaluation of physical properties. *Oil & Gas Science and Technology – Rev. IFP*, 63(5):677–686.
- [103] Lohrenz, J., Bray, B. G., and Clark, C. R. (1964). Calculating viscosities of reservoir fluids from their compositions. *Journal of Petroleum Technology*, 16(10):1171–1176.
- [104] Luo, Z.-Q., Pang, J.-S., and Ralph, D. (1996). *Mathematical programs with equilibrium constraints*. Cambridge University Press.
- [105] Maciejowski, J. M. (2000). *Predictive control with constraints*. Prentice Hall.
- [106] Mandela, R. K., Rengaswamy, R., and Narasimhan, S. (2009). Nonlinear state estimation of differential algebraic systems. *IFAC Proceedings Volumes*, 42(11):792–797.
- [107] Mandela, R. K., Rengaswamy, R., Narasimhan, S., and Sridhar, L. N. (2010). Recursive state estimation techniques for nonlinear differential algebraic systems. *Chemical Engineering Science*, 65(16):4548–4556.
- [108] Martín, Á., Bermejo, M. D., Mato, F. A., and Cocero, M. J. (2011). Teaching advanced equations of state in applied thermodynamics courses using open source programs. *Education for Chemical Engineers*, 6(4):e114–e121.
- [109] Meum, P., Tøndel, P., Godhavn, J.-M., and Aamo, O. M. (2008). Optimization of smart well production through nonlinear model predictive control. In *Proceedings of the 2008 SPE Intelligent Energy Conference and Exhibition*, Amsterdam, The Netherlands.
- [110] Michelsen, M. L. (1999). State function based flash specifications. *Fluid Phase Equilibria*, 158-160:617–626.
- [111] Michelsen, M. L. and Mollerup, J. M. (2007). *Thermodynamic models: fundamentals and computational aspects*. Tie-Line Publications, 2nd edition.
- [112] Mobed, P., Munusamy, S., Bhattacharyya, D., and Rengaswamy, R. (2017). State and parameter estimation in distributed constrained systems. 1. extended Kalman filtering of a special class of differential-algebraic equation systems. *Industrial & Engineering Chemistry Research*, 56(1):206–215.
- [113] Morton-Thompson, D. (1993). *Development Geology Reference Manual*. American Association of Petroleum Geologists.
- [114] Nævdal, G., Brouwer, D. R., and Jansen, J.-D. (2006). Waterflooding using closed-loop control. *Computational Geosciences*, 10(1):37–60.
- [115] Nocedal, J. and Wright, S. J. (2006). *Numerical optimization*. Springer Science & Business Media, 2nd edition.
- [116] Onwunali, J. E. and Durlofsky, L. J. (2010). Application of a particle swarm optimization algorithm for determining optimum well location and type. *Computational Geosciences*, 14(1):183–198.
- [117] Organization of the Petroleum Exporting Countries (2017). World oil outlook 2017. www.opec.org/opec_web/en/publications/340.htm. Accessed: May 9th 2018.
- [118] Outrata, J., Kočvara, M., and Zowe, J. (1998). *Nonsmooth approach to optimization problems*

- with equilibrium constraints: theory, applications and numerical results*, volume 28 of *Nonconvex Optimization and Its Applications*. Springer Science & Business Media.
- [119] Pastorino, R., Richiedei, D., Cuadrado, J., and Trevisani, A. (2013). State estimation using multibody models and non-linear Kalman filters. *International Journal of Non-Linear Mechanics*, 53:83–90.
- [120] Peters, E., Arts, R. J., Brouwer, G. K., Geel, C. R., Cullick, S., Lorentzen, R. J., Chen, Y., Dunlop, K. N. B., Vossepoel, F. C., Xu, R., Sarma, P., Alhutali, A. H., and Reynolds, A. C. (2010). Results of the Brugge benchmark study for flooding optimization and history matching. *SPE Reservoir Evaluation & Engineering*, 13(3):391–405.
- [121] PetroWiki (2015a). Compositional simulation of miscible processes. https://petrowiki.org/Compositional_simulation_of_miscible_processes. Accessed: August 1st 2018.
- [122] PetroWiki (2015b). Designing a miscible flood. https://petrowiki.org/Designing_a_miscible_flood. Accessed: August 1st 2018.
- [123] Phale, H. A. and Oliver, D. S. (2011). Data assimilation using the constrained ensemble Kalman filter. *SPE Journal*, 16(2):331–342.
- [124] Pinto, M. A. S., Ghasemi, M., Sorek, N., Gildin, E., and Schiozer, D. J. (2015). Hybrid optimization for closed-loop reservoir management. In *Proceedings of the 2015 SPE Reservoir Simulation Symposium*, Houston, Texas, USA.
- [125] Polívka, O. and Mikyška, J. (2014). Compositional modeling in porous media using constant volume flash and flux computation without the need for phase identification. *Journal of Computational Physics*, 272:149–169.
- [126] Prakash, J., Patwardhan, S. C., and Shah, S. L. (2008a). Constrained state estimation using particle filters. *IFAC Proceedings Volumes*, 41(2):6472–6477.
- [127] Prakash, J., Patwardhan, S. C., and Shah, S. L. (2008b). Constrained state estimation using the ensemble Kalman filter. In *Proceedings of the 2008 American Control Conference*, pages 3542–3547, Seattle, Washington, USA.
- [128] Puranik, Y., Bavdekar, V. A., Patwardhan, S. C., and Shah, S. L. (2012). An ensemble Kalman filter for systems governed by differential algebraic equations (DAEs). *IFAC Proceedings Volumes*, 45(15):531–536.
- [129] Purkayastha, S. N., Gates, I. D., and Trifkovic, M. (2015). Model-predictive-control (MPC) of steam trap subcool in steam-assisted gravity drainage (SAGD). *IFAC-PapersOnLine*, 48(8):539–544.
- [130] Purkayastha, S. N., Gates, I. D., and Trifkovic, M. (2018). Real-time multivariable model predictive control for steam-assisted gravity drainage. *AIChE Journal*, 64(8):3034–3041.
- [131] Purohit, J. L., Patwardhan, S. C., and Mahajani, S. M. (2013). DAE-EKF-based nonlinear predictive control of reactive distillation systems exhibiting input and output multiplicities. *Industrial & Engineering Chemistry Research*, 52(38):13699–13716.
- [132] Purohit, J. L., Patwardhan, S. C., and Mahajani, S. M. (2015). Performance evaluation of bayesian state estimators for nonlinear DAE systems using a moderately high dimensional reactive distillation column model. *Computer Aided Chemical Engineering*, 37:1763–1768.
- [133] Qin, S. J. and Badgwell, T. A. (1997). An overview of industrial model predictive control technology. *AIChE Symposium Series*, 93(316):232–256.
- [134] Qin, S. J. and Badgwell, T. A. (2000). An overview of nonlinear model predictive control applications. In Allgöwer, F. and Zheng, A., editors, *Nonlinear Model Predictive Control*, volume 26 of *Progress in Systems and Control Theory*, pages 369–392. Birkhäuser, Basel.
- [135] Qin, S. J. and Badgwell, T. A. (2003). A survey of industrial model predictive control technology. *Control Engineering Practice*, 11(7):733–764.
- [136] Qiu, L., Wang, Y., and Reitz, R. D. (2014). Multiphase dynamic flash simulations using entropy maximization and application to compressible flow with phase change. *AIChE Journal*, 60(8):3013–3024.
- [137] Raghunathan, A. U., Diaz, M. S., and Biegler, L. T. (2004). An MPEC formulation for dynamic optimization of distillation operations. *Computers & Chemical Engineering*, 28(10):2037–

- 2052.
- [138] Rawlings, J. B., Mayne, D. Q., and Diehl, M. M. (2017). *Model predictive control: theory, computation, and design*. Nob Hill Publishing, 2nd edition.
- [139] Ritschel, T. K. S., Capolei, A., Gaspar, J., and Jørgensen, J. B. (2018). An algorithm for gradient-based dynamic optimization of UV flash processes. *Computers and Chemical Engineering*, 114:281–295.
- [140] Ritschel, T. K. S., Capolei, A., and Jørgensen, J. B. (2017a). The adjoint method for gradient-based dynamic optimization of UV flash processes. *Computer Aided Chemical Engineering*, 40:2071–2076.
- [141] Ritschel, T. K. S., Capolei, A., and Jørgensen, J. B. (2017b). Dynamic optimization of UV flash processes. In *Proceedings of the 2017 Foundations of Computer Aided Process Operations / Chemical Process Control*, Tucson, Arizona, USA.
- [142] Ritschel, T. K. S., Gaspar, J., Capolei, A., and Jørgensen, J. B. (2016). An open-source thermodynamic software library. Technical Report DTU Compute Technical Report-2016-12, Department of Applied Mathematics and Computer Science, Technical University of Denmark.
- [143] Ritschel, T. K. S., Gaspar, J., and Jørgensen, J. B. (2017c). A thermodynamic library for simulation and optimization of dynamic processes. *IFAC-PapersOnLine*, 50(1):3542–3547.
- [144] Ritschel, T. K. S. and Jørgensen, J. B. (2018a). Computation of phase equilibrium in reservoir simulation and optimization. *IFAC-PapersOnLine*, 51(8):94–101.
- [145] Ritschel, T. K. S. and Jørgensen, J. B. (2018b). Dynamic optimization of thermodynamically rigorous models of multiphase flow in porous subsurface oil reservoirs. *Journal of Process Control*. In Submission.
- [146] Ritschel, T. K. S. and Jørgensen, J. B. (2018c). Nonlinear filters for state estimation of UV flash processes. In *Proceedings of the 2nd IEEE Conference on Control Technology and Applications*, Copenhagen, Denmark.
- [147] Ritschel, T. K. S. and Jørgensen, J. B. (2018d). Production optimization of a rigorous thermal and compositional reservoir flow model. *IFAC-PapersOnLine*, 51(8):76–81.
- [148] Ritschel, T. K. S. and Jørgensen, J. B. (2018e). Production optimization of thermodynamically rigorous isothermal and compositional models. In *Proceedings of the 16th European Conference on the Mathematics of Oil Recovery*, Barcelona, Spain.
- [149] Saha, S. and Carroll, J. J. (1997). The isoenergetic-isochoic flash. *Fluid phase equilibria*, 138(1–2):23–41.
- [150] Sahlodin, A. M., Watson, H. A. J., and Barton, P. I. (2016). Nonsmooth model for dynamic simulation of phase changes. *AIChE Journal*, 62(9):3334–3351.
- [151] Santori, G. and Luberti, M. (2016). Thermodynamics of thermally-driven adsorption compression. *Sustainable Materials and Technologies*, 10:1–9.
- [152] Saputelli, L., Malki, H., Canelon, J., and Nikolaou, M. (2002). A critical overview of artificial neural network applications in the context of continuous oil field optimization. In *SPE Annual Technical Conference and Exhibition*, San Antonio, Texas, USA.
- [153] Sarma, P. (2006). *Efficient closed-loop optimal control of petroleum reservoirs under uncertainty*. PhD thesis, Stanford University.
- [154] Sarma, P., Durlofsky, L. J., and Aziz, K. (2005). Efficient closed-loop production optimization under uncertainty. In *Proceedings of the 2005 SPE Europec/EAGE Annual Conference*, Madrid, Spain.
- [155] Sarma, P., Durlofsky, L. J., and Aziz, K. (2008). Computational techniques for closed-loop reservoir modeling with application to a realistic reservoir. *Petroleum Science and Technology*, 26(10–11):1120–1140.
- [156] Sarma, P., Durlofsky, L. J., Aziz, K., and Chen, W. H. (2006). Efficient real-time reservoir management using adjoint-based optimal control and model updating. *Computational Geosciences*, 10(1):3–36.
- [157] Schäfer, A., Kühl, P., Diehl, M., Schlöder, J., and Bock, H. G. (2007). Fast reduced multiple shooting methods for nonlinear model predictive control. *Chemical Engineering and Processing*:

- Process Intensification*, 46(11):1200–1214.
- [158] Shuai, Y. (2012). *Strategies for real time reservoir management*. PhD thesis, Louisiana State University.
- [159] Silva, V. L. S., Emerick, A. A., Couto, P., and Alves, J. L. D. (2017). History matching and production optimization under uncertainties - application of closed-loop reservoir management. *Journal of Petroleum Science and Engineering*, 157:860–874.
- [160] Simon, D. (2006). *Optimal state estimation: Kalman, H infinity, and nonlinear approaches*. John Wiley & Sons.
- [161] Simon, M. and Ulbrich, M. (2015). Adjoint based optimal control of partially miscible two-phase flow in porous media with applications to CO₂ sequestration in underground reservoirs. *Optimization and Engineering*, 16(1):103–130.
- [162] Smith, J. M., Van Ness, H. C., and Abbott, M. M. (2005). *Introduction to chemical engineering thermodynamics*. McGraw-Hill Education, 7th edition.
- [163] Stechliniski, P. G. and Barton, P. I. (2017). Dependence of solutions of nonsmooth differential-algebraic equations on parameters. *Journal of Differential Equations*, 262(3):2254–2285.
- [164] Sundman, B., Kattner, U. R., Palumbo, M., and Fries, S. G. (2015). OpenCalphad - a free thermodynamic software. *Integrating Materials and Manufacturing Innovation*, 4(1).
- [165] Talavera, A. L., Túpac, Y. J., and Vellasco, M. M. B. R. (2010). Controlling oil production in smart wells by MPC strategy with reinforcement learning. In *Proceedings of the 2010 SPE Latin American and Caribbean Petroleum Engineering Conference*, Lima, Peru.
- [166] Talebi, H. A., Abdollahi, F., Patel, R. V., and Khorasani, K. (2010). *Neural network-based state estimation of nonlinear systems: application to fault detection and isolation*, volume 395 of *Lecture Notes in Control and Information Sciences*. Springer-Verlag New York.
- [167] Teixeira, B. O. S., Tôrres, L. A. B., Aguirre, L. A., and Bernstein, D. S. (2008). Unscented filtering for interval-constrained nonlinear systems. In *Proceedings of the 47th IEEE Conference on Decision and Control*, pages 5116–5121, Cancun, Mexico.
- [168] Thomas, S. (2008). Enhanced oil recovery – an overview. *Oil & Gas Science and Technology – Rev. IFP*, 63(1):9–19.
- [169] Tulsyan, A., Gopaluni, R. B., and Khare, S. R. (2016). Particle filtering without tears: a primer for beginners. *Computers and Chemical Engineering*, 95:130–145.
- [170] Udy, J., Hansen, B., Maddux, S., Petersen, D., Heilner, S., Stevens, K., Lignell, D., and Hedengren, J. D. (2017). Review of field development optimization of waterflooding, EOR, and well placement focusing on history matching and optimization algorithms. *Processes*, 5(3).
- [171] Ungarala, S., Dolence, E., and Li, K. (2007). Constrained extended Kalman filter for nonlinear state estimation. *IFAC Proceedings Volumes*, 40(5):63–68.
- [172] U.S. Energy Information Administration (2017). International energy outlook 2017. www.eia.gov/outlooks/ieo/. Accessed: May 9th 2018.
- [173] Vachhani, P., Narasimhan, S., and Rengaswamy, R. (2004). Robust constrained estimation via unscented transformation. *IFAC Proceedings Volumes*, 37(9):317–322.
- [174] Van den Hof, P. M. J., Jansen, J.-D., van Essen, G., and Bosgra, O. H. (2009). Model-based control and optimization of large scale physical systems - challenges in reservoir engineering. In *Proceedings of the 2009 Chinese Control and Decision Conference*, pages xlii–li.
- [175] van Essen, G., Rezapour, A., Van den Hof, P. M. J., and Jansen, J. D. (2010). Integrated dynamic optimization and control in reservoir engineering using locally identified linear models. In *Proceedings of the 49th IEEE Conference on Decision and Control*, pages 7643–7648, Atlanta, Georgia, USA.
- [176] Völcker, C. (2011). *Production optimization of oil reservoirs*. PhD thesis, Technical University of Denmark.
- [177] Völcker, C., Jørgensen, J. B., and Stenby, E. H. (2011). Oil reservoir production optimization using optimal control. In *Proceedings of the 50th IEEE Conference on Decision and Control and European Control Conference*, pages 7937–7943, Orlando, Florida, USA.

- [178] Völcker, C., Jørgensen, J. B., Thomsen, P. G., and Stenby, E. H. (2010a). Adaptive stepsize control in implicit Runge-Kutta methods for reservoir simulation. *IFAC Proceedings Volumes*, 43(5):523–528.
- [179] Völcker, C., Jørgensen, J. B., Thomsen, P. G., and Stenby, E. H. (2010b). Explicit singly diagonally implicit Runge-Kutta methods and adaptive stepsize control for reservoir simulation. In *Proceedings of the 12th European Conference on the Mathematics of Oil Recovery*, Oxford, England.
- [180] Wächter, A. and Biegler, L. T. (2006). On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical Programming*, 106(1):25–57.
- [181] Walas, S. M. (1985). *Phase equilibria in chemical engineering*. Butterworth-Heinemann.
- [182] Wallis, J. R. (1983). Incomplete Gaussian elimination as a preconditioning for generalized conjugate gradient acceleration. In *Proceedings of the 1983 Reservoir Simulation Symposium*, pages 325–334, San Francisco, California, USA.
- [183] Wallis, J. R., Kendall, R. P., and Little, T. E. (1985). Constrained residual acceleration of conjugate residual methods. In *Proceedings of the 1985 Reservoir Simulation Symposium*, pages 415–428, Dallas, Texas, USA.
- [184] Wang, C., Li, G., and Reynolds, A. C. (2009). Production optimization in closed-loop reservoir management. *SPE journal*, 14(3):506–523.
- [185] Wang, K., Liu, H., Luo, J., and Chen, Z. (2018). Efficient CPR-type preconditioner and its adaptive strategies for large-scale parallel reservoir simulations. *Journal of Computational and Applied Mathematics*, 328:443–468.
- [186] Watson, H. A. J., Vikse, M., Gundersen, T., and Barton, P. I. (2017). Reliable flash calculations: part 1. nonsmooth inside-out algorithms. *Industrial & Engineering Chemistry Research*, 56(4):960–973.
- [187] Wen, X.-H. and Chen, W. H. (2007). Some practical issues on real-time reservoir model updating using ensemble Kalman filter. *SPE Journal*, 12(2):156–166.
- [188] Wilhelmsen, Ø., Skaugen, G., Hammer, M., Wahl, P. E., and Morud, J. C. (2013). Time efficient solution of phase equilibria in dynamic and distributed systems with differential algebraic equation solvers. *Industrial & Engineering Chemistry Research*, 52(5):2130–2140.
- [189] Wu, Z., Reynolds, A. C., and Oliver, D. S. (1999). Conditioning geostatistical models to two-phase production data. *SPE Journal*, 4(2):142–155.
- [190] Zaydullin, R., Voskov, D. V., James, S. C., Henley, H., and Lucia, A. (2014). Fully compositional and thermal reservoir simulation. *Computers and Chemical Engineering*, 63:51–65.
- [191] Zhang, X.-D. and Li, S.-R. (2007). Optimal control solving of polymer flooding in enhanced oil recovery with 2-D models. In *Proceedings of the 2007 IEEE International Conference on Control and Automation*, pages 1981–1986, Guangzhou, China.
- [192] Zhao, H., Li, Y., Yao, J., and Zhang, K. (2011). Theoretical research on reservoir closed-loop production management. *Science China Technological Sciences*, 54(10):2815–2824.
- [193] Zhao, Z., Huang, B., and Liu, F. (2014). Constrained particle filtering methods for state estimation of nonlinear process. *AIChE Journal*, 60(6):2072–2082.

Part V

Appendix

First order optimality conditions

The content of this appendix is based on the paper in Appendix G. In this appendix, we derive the first order optimality conditions (also called Karush-Kuhn-Tucker or KKT conditions) of optimization problems in the form

$$\min_y f(y), \tag{A.1a}$$

$$\text{subject to } g(y) = x, \tag{A.1b}$$

$$h(y) = 0. \tag{A.1c}$$

In order to derive the first order optimality conditions of (A.1), we introduce the Lagrange function:

$$\mathcal{L}(y, \eta, \mu; x) = f(y) - \eta'(g(y) - x) - \mu'h(y). \tag{A.2}$$

η and μ are the Lagrange multipliers associated with the constraints (A.1b) and (A.1c), respectively. The first order optimality conditions for the optimization problem (A.1) are

$$\nabla_y \mathcal{L}(y, \eta, \mu; x) = \nabla f(y) - \nabla g(y)\eta - \nabla h(y)\mu = 0, \tag{A.3a}$$

$$\nabla_\eta \mathcal{L}(y, \eta, \mu; x) = -(g(y) - x) = 0, \tag{A.3b}$$

$$\nabla_\mu \mathcal{L}(y, \eta, \mu; x) = -h(y) = 0. \tag{A.3c}$$

We introduce $z = [\eta; \mu]$ such that we can rewrite (A.3) as

$$G(x, y, z) = 0. \tag{A.4}$$

We refer to the paper in Appendix H for more details on flash optimization problems. The Jacobian matrices of the first order optimality conditions (A.4) are

$$\frac{\partial G}{\partial x} = \begin{bmatrix} 0 \\ \mathbf{I} \\ 0 \end{bmatrix}, \tag{A.5a}$$

$$\begin{bmatrix} \frac{\partial G}{\partial y} & \frac{\partial G}{\partial z} \end{bmatrix} = K, \tag{A.5b}$$

where the matrix $K = K(y, z; x) = K(y, \eta, \mu; x)$ is the KKT matrix for (A.1):

$$K(y, \eta, \mu; x) = \begin{bmatrix} \nabla_{yy}^2 \mathcal{L}(y, \eta, \mu; x) & -\nabla g(y) & -\nabla h(y) \\ -\nabla g(y)' & 0 & 0 \\ -\nabla h(y)' & 0 & 0 \end{bmatrix}, \tag{A.6}$$

The Hessian of the Lagrange function, $\nabla_{yy}^2 \mathcal{L} = \nabla_{yy}^2 \mathcal{L}(y, \eta, \mu; x)$, is

$$\nabla_{yy}^2 \mathcal{L} = \nabla^2 f(y) - \sum_i \eta_i \nabla^2 g_i(y) - \sum_i \mu_i \nabla^2 h_i(y). \tag{A.7}$$

The sums in (A.7) span over each element of the functions $g(y)$ and $h(y)$.

Description of workstation

The numerical examples presented in Chapter 9 and Chapter 10 are carried out using a 64-bit workstation with an Ubuntu 16.04 operating system. The workstation has 15.6 GB RAM and four Intel Core i7 3.60 GHz cores. The workstation has 8,192 KB level 3 cache which is shared among all cores. Each core has 256 KB level 2 cache and 64 KB level 1 cache.

Paper I

Dynamic Optimization of UV Flash Processes

Authors:

Tobias K. S. Ritschel, Andrea Capolei, and John Bagterp Jørgensen

Published in:

Proceedings of Foundations of Computer Aided Process Operations / Chemical Processes Control (FOCAPO/CPC), January 8-12, 2017, Tucson, Arizona, USA.

DYNAMIC OPTIMIZATION OF UV FLASH PROCESSES

Tobias K. S. Ritschel, Andrea Capolei and John Bagterp Jørgensen*

Department of Applied Mathematics and Computer Science & Center for Energy Resources Engineering,
Technical University of Denmark, DK-2800 Kgs Lyngby, Denmark

Abstract

UV flash processes, also referred to as isoenergetic-isochoric flash processes, occur for dynamic simulation and optimization of vapor-liquid equilibrium processes. Dynamic optimization and nonlinear model predictive control of distillation columns, certain two-phase flow problems, as well as oil reservoirs with significant compositional and thermal effects may be conducted as dynamic optimization of UV flash processes. The dynamic optimization problem involving a UV flash problem is formulated as a bilevel optimization problem. This problem is solved using a gradient based single-shooting method. The gradients are computed using the adjoint method and different off-the-shelf optimization software (fmincon, IPOPT, KNITRO, NPSOL) are used for the numerical optimization. Computational results are reported for a flash process involving benzene, toluene and diphenyl. The computational experiments demonstrate that the optimization solver, the compiler, and high-performance linear algebra software are all important for efficient dynamic optimization of UV flash processes.

Keywords

Optimization, Optimal Control, Differential Algebraic Equations, Vapor-Liquid Equilibrium

1 Introduction

Dynamic optimization of vapor-liquid equilibrium (VLE) processes are used in operation and control of distillation columns, certain two-phase flow problems, and oil reservoirs with significant thermal and compositional effects. Therefore, such processes require efficient computational methods for dynamic optimization of UV flash processes. The UV flash problem is also known as the isoenergetic-isochoric flash problem or the UV n flash problem. UV n refers to specification of the internal energy, U , the total volume, V , and the total material amount (moles), n . The second law of thermodynamics, i.e. the entropy of a closed system is maximal, is used to determine the equilibrium composition with U , V , and n specified (Michelsen, 1999). The UV flash problem is different from the more common PT flash problem that occurs in steady-state optimization problems. However, it can be demonstrated that the PT flash problem with additional constraints on the internal energy, U , and the volume, V , is equivalent to the UV flash

problem. Algorithmic oriented approaches to dynamic optimization of VLE processes use a nested method in which PT flash problems are solved in the inner loop, and outer loops converge the internal energy, U , and volume, V , to their specified values. Such approaches suffer from computational inefficiency and complicated computations for the gradients. Alternatively, simultaneous methods (Biegler, 2010), multiple- or single-shooting methods (Capolei and Jørgensen, 2012) may be used for dynamic optimization of UV flash processes. In this paper, we present a novel algorithm for dynamic optimization of UV flash processes. The algorithm is based on the single-shooting method and an adjoint method is used for computation of the gradients (Jørgensen, 2007). The numerical integration of the semi-explicit index-1 differential algebraic (DAE) system is the key computational operation in the single-shooting method. We report numerical results for a three-component dynamic UV flash as well as the computational performance for implementations in C and Matlab using different optimization software, different linear algebra software, and different compilers.

*Corresponding author jbjo@dtu.dk. Funded by the Danish Advanced Technology Foundation (OPTION; 63-2013-3).

2 Optimal Control Problem

We consider the following optimal control problem (OCP)

$$\min_{[x(t); y(t); z(t)]_{t_0}^{t_f}, \{u_k\}_{k \in \mathcal{N}}} \phi = \phi \left([y(t); u(t); d(t)]_{t_0}^{t_f} \right) \quad (1a)$$

subject to

$$x(t_0) = \hat{x}_0, \quad (1b)$$

$$G(x(t), y(t), z(t)) = 0, \quad t \in \mathcal{T}, \quad (1c)$$

$$\dot{x}(t) = F(y(t), u(t), d(t)), \quad t \in \mathcal{T}, \quad (1d)$$

$$u(t) = u_k, \quad t \in [t_k, t_{k+1}[, \quad k \in \mathcal{N}, \quad (1e)$$

$$d(t) = \hat{d}_k, \quad t \in [t_k, t_{k+1}[, \quad k \in \mathcal{N}, \quad (1f)$$

$$\{u_k\}_{k \in \mathcal{N}} \in \mathcal{U}, \quad (1g)$$

where the objective function is in Lagrange form

$$\phi = \int_{t_0}^{t_f} \Phi(y(t), u(t), d(t)) dt. \quad (2)$$

$x(t)$ is the state vector, $y(t)$ is a vector of algebraic variables, and $z(t)$ is a vector of adjoint algebraic variables. The estimated initial states, \hat{x}_0 , and the predicted disturbances, $\{\hat{d}_k\}_{k \in \mathcal{N}}$, are parameters in the optimization problem. $[x(t); y(t); z(t)]_{t_0}^{t_f}$ is a vector of dependent decision variables, whereas $\{u_k\}_{k \in \mathcal{N}}$ are independent decision variables. The time interval is $\mathcal{T} = [t_0, t_f]$ and the control indices are $\mathcal{N} = \{0, 1, \dots, N-1\}$.

The OCP (1) includes algebraic constraints (1c) and differential equations (1d). The algebraic constraints (1c) are formulated such that they can be used to model equilibrium processes, e.g. VLE processes. Equilibrium processes can be formulated as optimization problems and (1c) can represent the Karush-Kuhn-Tucker (KKT) conditions of these optimization problems. The differential equations (1d) are obtained from conservation principles and the states, $x(t)$, represent the conserved quantities. The right-hand-side in (1d) depends on the algebraic variables, $y(t)$, which are implicit functions of the states through the algebraic constraints (1c), i.e. $y(t) = y(x(t))$. By this statement, we assume that given $x(t)$, it is possible to compute $y(t) = y(x(t))$ and $z(t) = z(x(t))$ by solving $G(x(t), y(t), z(t)) = 0$. This is true for the VLE processes considered in this work. We define the single shooting objective ψ by

$$\psi = \psi(\{u_k\}_{k \in \mathcal{N}}; \hat{x}_0, \{\hat{d}_k\}_{k \in \mathcal{N}}) = \left\{ \phi : (1b)-(1f) \right\}. \quad (3)$$

Given $\{u_k\}_{k \in \mathcal{N}}$, \hat{x}_0 , and $\{\hat{d}_k\}_{k \in \mathcal{N}}$, ψ is computed as the objective function, ϕ , obtained by integrating (2) using

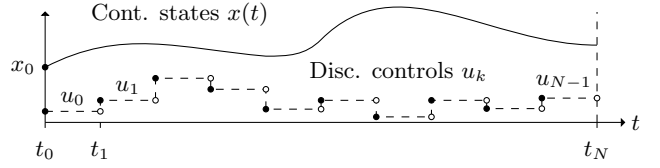


Figure 1. Sketch of the single shooting principle. The controls, $u(t)$, are discretized in time and the continuous states, $x(t)$, are considered functions of the controls. The objective is evaluated by solving the semi-explicit DAEs for a given initial value, x_0 , and a given set of controls, $\{u_k\}_{k=0}^{N-1}$.

the solution of (1c)-(1d) with $x(t_0) = \hat{x}_0$, $u(t) = u_k$ for $t \in [t_k, t_{k+1}[$ and $k \in \mathcal{N}$, and $d(t) = \hat{d}_k$ for $t \in [t_k, t_{k+1}[$ and $k \in \mathcal{N}$, i.e. (1b) and (1e)-(1f). Figure 1 illustrates the discretization of the inputs, u , and the numerical computation of the continuous states, x . This is the principle in the single-shooting method and the principle that is used for computation of ψ . With ψ defined by (3), the OCP (1) with the objective function (2) can be expressed as the finite dimensional constrained optimization problem

$$\min_{\{u_k\}_{k \in \mathcal{N}}} \psi = \psi \left(\{u_k\}_{k \in \mathcal{N}}; \hat{x}_0, \{\hat{d}_k\}_{k \in \mathcal{N}} \right), \quad (4a)$$

$$\text{s.t.} \quad \{u_k\}_{k \in \mathcal{N}} \in \mathcal{U}. \quad (4b)$$

The set \mathcal{U} is often a polyhedron such that the constraints (4b) can be expressed by $u_{\min} \leq u \leq u_{\max}$ and $b_l \leq Au \leq b_u$ where $u = [u_0; u_1; \dots; u_{N-1}]$. Gradient-based optimization algorithms for solution of the nonlinear program (4), and thus the optimal control problem (1), require evaluation of the objective function, ψ , and the gradients, $\{\nabla_{u_k} \psi\}_{k \in \mathcal{N}}$. These computations involve numerical solution of the differential-algebraic equations (DAE), (1c) and (1d), along with computation of the integral (2).

2.1 Equilibrium Constraints

The equilibrium processes that are considered in this work can be described as the solution to a parametric equality constrained optimization problem in the following form

$$\min_y f(y) \quad (5a)$$

$$\text{s.t.} \quad g(y) = x, \quad (5b)$$

$$h(y) = 0. \quad (5c)$$

The Lagrange function associated with the equilibrium optimization problem (5) is

$$\mathcal{L}(y, \eta, \mu; x) = f(y) - \eta^T(g(y) - x) - \mu^T h(y), \quad (6)$$

where η and μ are Lagrange multipliers associated with (5b) and (5c), respectively. The KKT conditions (first order optimality conditions) for a minimizer ($y = y(x), \eta = \eta(x), \mu = \mu(x)$) are

$$\nabla_y \mathcal{L}(y, \eta, \mu; x) = \nabla f(y) - \nabla g(y)\eta - \nabla h(y)\mu = 0, \quad (7a)$$

$$\nabla_\eta \mathcal{L}(y, \eta, \mu; x) = -(g(y) - x) = 0, \quad (7b)$$

$$\nabla_\mu \mathcal{L}(y, \eta, \mu; x) = -h(y) = 0. \quad (7c)$$

By introducing the vector $z = [\eta; \mu]$, we can rewrite the system (7) as the algebraic constraints (1c).

3 Numerical Method

The computation of (3) requires solution of the semi-explicit differential-algebraic initial value problem (1b)-(1d). Subsequently, when $(y(t), u(t), d(t))$ is given, $\psi = \phi$ is computed by quadrature. The system (1b)-(1d) is stiff. Therefore, an implicit method must be used for numerically efficient solution of (1b)-(1d). We exemplify the involved numerical computation using Euler's implicit method. However, the principal numerical methods are also applicable with other implicit solvers such as the ESDIRK methods (Kristensen et al., 2004) and BDF based methods (Barton and Lee, 2002). Furthermore, we describe computation of $\nabla_{u_k} \psi$ for $k \in \mathcal{N}$ by an adjoint method (Jørgensen, 2007; Völcker et al., 2011; Capolei and Jørgensen, 2012). These gradients (sensitivities) may also be computed by a forward method.

3.1 Numerical Integration

Define $w = [x; y; z]$ and define the residual function

$$\begin{aligned} R_{k+1} &= R_{k+1}(w_{k+1}) = R_{k+1}(w_{k+1}; x_k, u_k, d_k) \\ &= R_{k+1}(x_{k+1}, y_{k+1}, z_{k+1}; x_k, u_k, d_k) \\ &= \begin{bmatrix} D_{k+1}(x_{k+1}, x_k, y_{k+1}, u_k, d_k) \\ G(x_{k+1}, y_{k+1}, z_{k+1}) \end{bmatrix} \end{aligned} \quad (8)$$

for $k \in \mathcal{N}$ with $D_{k+1} = x_{k+1} - x_k - \Delta t_k F(y_{k+1}, u_k, d_k)$. Given $x_0 = \hat{x}_0$, $\{u_k\}_{k=0}^{N-1}$, and $\{d_k = \hat{d}_k\}_{k=0}^{N-1}$, the implicit Euler discretization of (1b)-(1d) corresponds to solving

$$R_{k+1} = R_{k+1}(w_{k+1}) = 0, \quad k \in \mathcal{N} \quad (9)$$

sequentially for $\{w_{k+1}\}_{k=0}^{N-1}$ by marching forward. Equation (9) is solved by an inexact Newton method, i.e. by

solving a sequence of linear systems

$$w_{k+1}^{m+1} = w_{k+1}^m - M_{k+1}^{-1} R_{k+1}(w_{k+1}^m), \quad (10)$$

until some convergence criteria is satisfied. The iteration matrix is

$$M_{k+1} = \frac{\partial R_{k+1}}{\partial w_{k+1}} = \begin{bmatrix} I & -\Delta t_k \frac{\partial F}{\partial y} & 0 \\ \frac{\partial G}{\partial x} & \frac{\partial G}{\partial y} & \frac{\partial G}{\partial z} \end{bmatrix}, \quad (11)$$

where

$$\frac{\partial G}{\partial x} = \begin{bmatrix} 0 \\ I \\ 0 \end{bmatrix}, \quad K = \begin{bmatrix} \frac{\partial G}{\partial y} & \frac{\partial G}{\partial z} \end{bmatrix} = \begin{bmatrix} \nabla_{yy}^2 \mathcal{L} & -\nabla g & -\nabla h \\ -\nabla g^T & 0 & 0 \\ -\nabla h^T & 0 & 0 \end{bmatrix}.$$

K denotes the KKT matrix of the equilibrium conditions (7). The second derivative of the Lagrangian, \mathcal{L} , with respect to y is given by

$$\nabla_{yy}^2 \mathcal{L} = \nabla^2 f - \sum_i \eta_i \nabla^2 g_i - \sum_i \mu_i \nabla^2 h_i. \quad (12)$$

3.2 Gradients by the Adjoint Method

We substitute the discrete residual function (8) for the differential-algebraic constraints (1c)-(1d) in the function ψ given by (3) to obtain the following single shooting objective function, in which the zero-order-hold parametrization of the input and disturbances (1e)-(1f) have been applied

$$\psi = \psi(\{u_k\}_{k \in \mathcal{N}}; \hat{x}_0, \{\hat{d}_k\}_{k \in \mathcal{N}}) \quad (13a)$$

$$= \left\{ \phi = \sum_{k \in \mathcal{N}} \Phi_k(y_{k+1}, u_k, \hat{d}_k) : \right. \quad (13b)$$

$$x_0 = \hat{x}_0, \quad (13c)$$

$$R_{k+1}(w_{k+1}, x_k, u_k, \hat{d}_k) = 0, \quad k \in \mathcal{N}, \quad (13d)$$

$$\left. [x_{k+1}; y_{k+1}; z_{k+1}] = w_{k+1}, \quad k \in \mathcal{N} \right\}. \quad (13e)$$

The Lagrange objective (2) is approximated by the sum (13b) in which Φ_k approximates the integral over $[t_k, t_{k+1}[$ using the rectangle rule with y_{k+1} (rather than y_k)

$$\Phi_k = \Phi_k(y_{k+1}, u_k, \hat{d}_k) = \Delta t_k \Phi(y_{k+1}, u_k, \hat{d}_k). \quad (14)$$

The adjoints, $\{\lambda_k\}_{k=1}^N$, are computed by marching backwards in the equations

$$\left(\frac{\partial R_N}{\partial w_N} \right)^T \lambda_N = -\nabla_{w_N} \Phi_{N-1}, \quad (15a)$$

$$\left(\frac{\partial R_k}{\partial w_k} \right)^T \lambda_k = - \left(\frac{\partial R_{k+1}}{\partial w_k} \right)^T \lambda_{k+1} - \nabla_{w_k} \Phi_{k-1}, \quad (15b)$$

for $k \in \{N-1, N-2, \dots, 1\}$. The Jacobian of the discrete residual $\frac{\partial R_{k+1}}{\partial w_{k+1}}(w_{k+1}, w_k, u_k, \hat{d}_k)$ was defined in (11) and the Jacobian with respect to the states and the algebraic variables in the previous timestep is

$$\frac{\partial R_{k+1}}{\partial w_k}(w_{k+1}, w_k, u_k, \hat{d}_k) = - \begin{bmatrix} \mathbf{I} & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad (16)$$

for $k = 1, \dots, N-1$. The gradient of the objective is

$$\nabla_{w_{k+1}} \Phi_k = \begin{bmatrix} 0 \\ \nabla_{y_{k+1}} \Phi_k \\ 0 \end{bmatrix}, \quad k \in \mathcal{N}. \quad (17)$$

The gradients of ψ with respect to the inputs, $\{\nabla_{u_k} \psi\}_{k \in \mathcal{N}}$, are computed by

$$\nabla_{u_k} \psi = \nabla_{u_k} \Phi_k + \left(\frac{\partial R_{k+1}}{\partial u_k} \right)^T \lambda_{k+1}, \quad k \in \mathcal{N}. \quad (18)$$

4 UV Flash Example

Neglecting kinetic and potential energy, the energy- and mass balance of a flash unit may be expressed as

$$\dot{U}(t) = H_F^v(t) + H_F^l(t) - H_V(t) - H_L(t) + Q(t), \quad (19a)$$

$$\dot{n}_i(t) = f_{F,i}^v(t) + f_{F,i}^l(t) - v_i(t) - l_i(t), \quad (19b)$$

for the components $i = 1, \dots, N_C$. U is the internal energy and n_i is the total holdup of component i . H_V and H_L are the enthalpies of the vapor and liquid streams respectively, and H_F^v and H_F^l are the vapor and liquid enthalpies of the feed. $f_{F,i}^v$ and $f_{F,i}^l$ are the vapor and liquid component flow rates of the feed. v_i and l_i are the component flow rates of the vapor and liquid streams. The conservation equations (19) are in the form of the differential equation (1d) where the function F is

$$F(y(t), u(t), d(t)) = \begin{bmatrix} H_F^v(t) + H_F^l(t) - H_V(t) - H_L(t) + Q(t) \\ f_F^v(t) + f_F^l(t) - v(t) - l(t) \end{bmatrix}. \quad (20)$$

Let the state variables, x , the algebraic variables, y , the Lagrange multipliers, z , the manipulated variables, u , and the disturbance variables, d , be defined as:

$$x = [U; n] \in \mathbb{R}^{1+N_C}, \quad (21a)$$

$$y = [T; P; n^v; n^l] \in \mathbb{R}^{2+2N_C}, \quad (21b)$$

$$z = [\mu; \eta] \in \mathbb{R}^{2+N_C}, \quad (21c)$$

$$u = [Q; F_V; F_L] \in \mathbb{R}^3, \quad (21d)$$

$$d = [T_F; P_F; f_F^v; f_F^l] \in \mathbb{R}^{2+2N_C}. \quad (21e)$$

The VLE in the flash tank is governed by

$$\max_{T, P, n^v, n^l} S = S^v(T, P, n^v) + S^l(T, P, n^l) \quad (22a)$$

$$\text{s.t.} \quad U^v(T, P, n^v) + U^l(T, P, n^l) = U, \quad (22b)$$

$$V^v(T, P, n^v) + V^l(T, P, n^l) = V, \quad (22c)$$

$$n_i^v + n_i^l = n_i, \quad i = 1, \dots, N_C. \quad (22d)$$

The VLE problem (22) is in the form of equation (5) where the functions f , g and h are

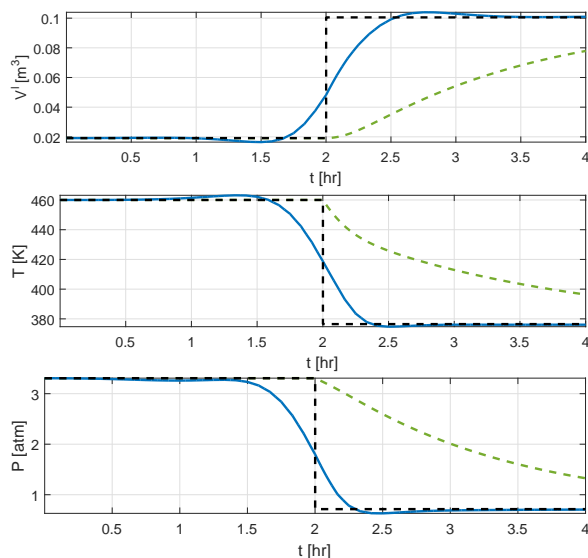
$$\begin{aligned} f(y) &= f(T, P, n^v, n^l) \\ &= - \left(S^v(T, P, n^v) + S^l(T, P, n^l) \right), \end{aligned} \quad (23a)$$

$$\begin{aligned} g(y) &= g(T, P, n^v, n^l) \\ &= \begin{bmatrix} U^v(T, P, n^v) + U^l(T, P, n^l) \\ n^v + n^l \end{bmatrix}, \end{aligned} \quad (23b)$$

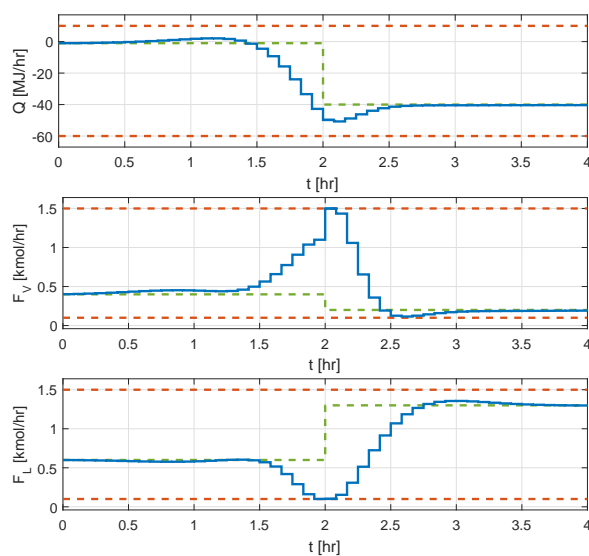
$$\begin{aligned} h(y) &= h(T, P, n^v, n^l) \\ &= V^v(T, P, n^v) + V^l(T, P, n^l) - V, \end{aligned} \quad (23c)$$

We consider a mixture of benzene, toluene and diphenyl that is separated in a flash tank. Figure 2 shows the controlled variables and the manipulated variables for a least-squares optimal transition between two steady states. The optimal transition is computed by dynamic optimization and compared to an open-loop non-optimized transition. The optimized transition is significantly faster than the transition based on the steady state values of the manipulated variables. Figures 3-5 show the composition variables, selected thermodynamic functions (H , S , G), and the state variables for the optimal transition.

Table 1 shows the computation time for solving the OCP (4) with the presented dynamic optimization algorithm using different compilers, optimization libraries, and linear algebra libraries. `fmincon` is at least 10 times faster when used with a compiled C implementation for numerical integration compared to a Matlab implementation for numerical integration. When the compiled C code is called from Matlab, it will in all cases be using Intel MKL. Using IPOPT gives a modest speedup of between 6 and 8, because IPOPT uses a limited-memory BFGS update strategy tailored for large-scale systems. KNITRO results in a speedup of between 41 and 48 compared to a pure Matlab implementation, and NPSOL gives a speedup of between 47 and 66. The Intel compilers and Intel MKL generally have a positive effect on the implementations using KNITRO and NPSOL. The compiler `icc` (Intel) rather than the compiler `gcc` and Intel MKL rather than Netlib's BLAS/LAPACK have a negative effect on the implementation using IPOPT.



(a) Controlled variables (CV).



(b) Manipulated variables (MV).

Figure 2. Transition between two steady states by dynamic optimization (blue) and use of steady state inputs (green dashed).

5 Conclusion

We presented an adjoint single-shooting algorithm for gradient-based dynamic optimization of flash processes. The algorithm simultaneously solves the equilibrium conditions and the differential conservation equations. A simulation example demonstrates that dynamic optimization enables fast transition between steady states. This is an important feature for nonlinear model predictive control applications. The numerical experiments show that using a simultaneous numerical integration scheme in the adjoint single shooting algorithm yields faster solution than with a nested numerical integration scheme. This is primarily due to fewer evaluations of the thermodynamic properties. Furthermore, the computational time of the adjoint single-shooting algorithm is compared for four optimization solvers (KNITRO, NPSOL, IPOPT and fmincon [Matlab]) and illustrates that using a compiled language together with an appropriate NLP solver library is essential to good computational performance. KNITRO and NPSOL give significant speedup compared to a pure Matlab implementation. IPOPT is designed for large-scale problems and less appropriate for the small dense problem considered in this work. Furthermore, the Intel compilers in combination with the Intel MKL are generally more efficient than using GNU compilers and Netlib's BLAS/LAPACK distribution. Using the suggested algorithm, we can solve a dynamic UV flash op-

timization problem with 3 components in less than 0.2 seconds.

References

- Barton, P. I. and Lee, C. K. (2002). Modeling, simulation, sensitivity analysis, and optimization of hybrid systems. *ACM Transactions on Modeling and Computer Simulation*, pages 256–289.
- Biegler, L. T. (2010). *Nonlinear Programming: Concepts, Algorithms, and Applications to Chemical Processes*. SIAM.
- Capolei, A. and Jørgensen, J. (2012). Solution of constrained optimal control problems using multiple shooting and ESDIRK methods. In *Proceedings of the 2012 American Control Conference*, pages 295–300.
- Jørgensen, J. B. (2007). Adjoint sensitivity results for predictive control, state-and parameter-estimation with nonlinear models. In *European Control Conference (ECC), 2007*, pages 3649–3656.
- Kristensen, M. R., Jørgensen, J. B., Thomsen, P. G., and Jørgensen, S. B. (2004). An ESDIRK method with sensitivity analysis capabilities. *Computers & Chemical Engineering*, 28(12):2695–2707.
- Michelsen, M. L. (1999). State function based flash specifications. *Fluid Phase Equilibria*, 158:617–626.
- Völcker, C., Jørgensen, J. B., and Stenby, E. H. (2011). Oil reservoir production optimization using optimal control. In *Conference on Decision and Control*, pages 7937–7943.

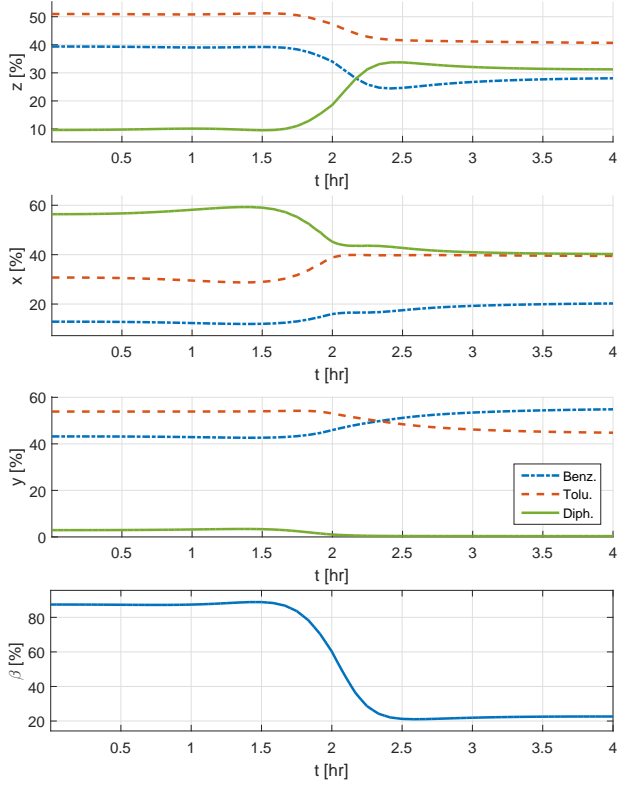


Figure 3. Composition variables of the optimal transition. Overall, z , liquid, x , and vapor, y , mole fractions as well as the vapor fraction, β .

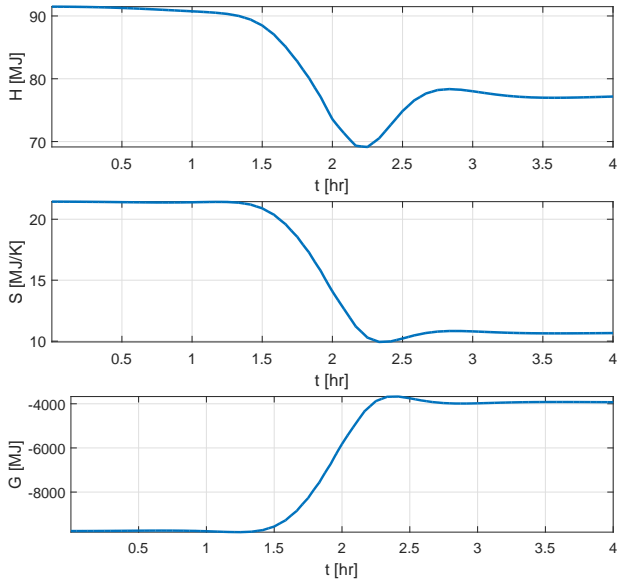


Figure 4. Thermodynamic state functions for the flash tank in the optimal transition. H is the enthalpy, S is the entropy, and G is Gibbs' free energy.

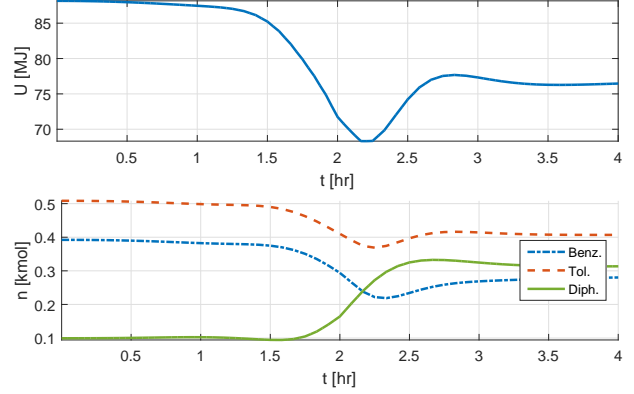


Figure 5. State variables for the optimal transition.

Table 1. Absolute (in seconds) and relative computation time of solving the OCP (4) using simultaneous numerical integration. Average over 10 calls. $fmincon$ (C) uses a C implementation of simultaneous numerical integration and $fmincon$ (M) uses a Matlab implementation. Sp.up is short for speed-up compared to the pure Matlab implementation, i.e. $fmincon$ (M).

	fmincon		IPOPT	KNITRO	NPSOL
	Matlab	C	C	C	C
Iter.	192	192	445	168	158
Func.	195	195	1435	171	159
gcc, gfortran, Netlib BLAS/LAPACK					
Abs.	12.461	1.185	1.663	0.298	0.263
Rel.	1.000	0.095	0.133	0.024	0.021
Sp.up	1.0	10.5	7.5	41.8	47.4
icc, gfortran, Netlib BLAS/LAPACK					
Abs.	12.461	1.081	1.753	0.277	0.246
Rel.	1.000	0.087	0.141	0.022	0.020
Sp.up	1.0	11.5	7.1	45.0	50.7
icc, gfortran, Intel MKL					
Abs.	12.461	1.138	1.876	0.277	0.213
Rel.	1.000	0.091	0.151	0.022	0.017
Sp.up	1.0	10.9	6.6	45.0	58.5
icc, ifort, Intel MKL					
Abs.	12.461	1.149	1.618	0.262	0.189
Rel.	1.000	0.092	0.130	0.021	0.015
Sp.up	1.0	10.8	7.7	47.6	65.9

Paper II

A Thermodynamic Library for Simulation and Optimization of Dynamic Processes

Authors:

Tobias K. S. Ritschel, Jozsef Gaspar, and John Bagterp Jørgensen

Published in:

IFAC-PapersOnLine 50(1), 3542–3547, 2017.

DOI: <https://doi.org/10.1016/j.ifacol.2017.08.951>.

A Thermodynamic Library for Simulation and Optimization of Dynamic Processes^{*}

Tobias K. S. Ritschel, Jozsef Gaspar, John Bagterp Jørgensen

*Department of Applied Mathematics and Computer Science &
Center for Energy Resources Engineering (CERE),
Technical University of Denmark, DK-2800 Kgs. Lyngby, Denmark*

Abstract: Process system tools, such as simulation and optimization of dynamic systems, are widely used in the process industries for development of operational strategies and control for process systems. These tools rely on thermodynamic models and many thermodynamic models have been developed for different compounds and mixtures. However, rigorous thermodynamic models are generally computationally intensive and not available as open-source libraries for process simulation and optimization. In this paper, we describe the application of a novel open-source rigorous thermodynamic library, ThermoLib, which is designed for dynamic simulation and optimization of vapor-liquid processes. ThermoLib is implemented in Matlab and C and uses cubic equations of state to compute vapor and liquid phase thermodynamic properties. The novelty of ThermoLib is that it provides analytical first and second order derivatives. These derivatives are needed for efficient dynamic simulation and optimization. The analytical derivatives improve the computational performance by a factor between 12 and 35 as compared to finite difference approximations. We present two examples that use ThermoLib routines in their implementations: (1) simulation of a vapor-compression cycle, and (2) optimal control of an isoenergetic-isochoric flash separation process. The ThermoLib software used in this paper is distributed as open-source software at www.psetools.org.

© 2017, IFAC (International Federation of Automatic Control) Hosting by Elsevier Ltd. All rights reserved.

Keywords: Thermodynamic library, Process simulation, Dynamic optimization, Vapor compression cycle, Vapor-liquid equilibrium, Flash separation

1. INTRODUCTION

Rigorous thermodynamic computations of vapor and liquid properties (volume, enthalpy, entropy) constitute a significant part of the computations in the dynamic simulation and optimization of many industrial processes governed by vapor-liquid equilibrium. Such processes are ubiquitous and include power cycles, vapor compression cycles, refrigeration systems, vapor-liquid separations in tanks, distillation columns, and oil reservoirs modeled by compositional and thermal models. Thermodynamic models, such as equations of state (EOS) and activity coefficient models, are widely used in the process industries and represent a certain level of complexity. As process simulation and optimization software depends critically on thermodynamic property computations, it is crucial to the computational performance that such thermodynamic models are implemented efficiently. Simulation and gradient-based optimization of dynamic process systems governed by vapor-liquid equilibrium require in addition to the thermodynamic properties themselves also first and in many cases second order derivatives of the thermodynamic properties.

In this paper, we present a performance study of a recently developed open-source thermodynamic library, ThermoLib, which is designed for efficient dynamic simulation and optimization. It is implemented in Matlab and C. The novelty of ThermoLib is that it provides routines for evaluating analytical first and second order derivatives with respect to temperature, pressure, and mole numbers. The derivatives are obtained with symbolic differentiation. The library is based on parameters and correlations from the DIPPR database (Thomson, 1996), the Peng-Robinson (PR) EOS (Peng and Robinson, 1976), the Soave-Redlich-Kwong (SRK) EOS (Soave, 1972), and the van der Waals mixing rules (Shibata and Sandler, 1989). ThermoLib is an open-source library and is available from www.psetools.org (Ritschel et al., 2016). There are already a few open-source libraries such as the simulink toolbox T-MATS by Chapman et al. (2014) and a Matlab library by Martín et al. (2011). Furthermore, the CALPHAD software OpenCalphad developed by Sundman et al. (2015) contains an open-source module for evaluation of thermodynamic properties. However, none of these libraries provide both first and second order derivatives.

We present two examples that use ThermoLib routines in their implementations, namely simulation of a heat pump and dynamic optimization of an isoenergetic-isochoric flash separation process. Heat pumps are promising for the recovery of waste and ground heat and have numerous complex applications in heating and cooling, i.e. air

^{*} This project is funded partly by: 1) Innovation Fund Denmark in the CITIES project (1305-00027B) and in the OPTION project (63-2013-3), 2) the interreg project Smart Cities Accelerator (10606 SCA), and 3) EUDP 64013-0558 in the IEA Annex for energy efficient process control.

conditioning, floor heating, cryogenic air distillation, drying processes, etc. The European Heat Pump Association (2015) has reported that a total heat pump capacity of approximately 6.6 GW was installed in Europe during 2014, producing 13 TWh of energy and reducing CO₂-equivalent emissions by 2.09 Mt. Consequently, efficient simulation of heat pump devices is important to the design of economically attractive control methodologies, and will contribute to energy savings in both household and industrial applications. In the second example, we consider separation of hydrogen sulfide (H₂S) from a gas condensate. This is an example of a multi-component vapor-liquid equilibrium process that appears in e.g. distillation columns and is ubiquitous in the oil and chemical process industries.

This paper is structured as follows. Section 2 gives a brief description of the ThermoLib interface and presents a set of performance tests of selected library routines. Section 3 presents simulations of the heat pump and Section 4 presents an optimal control strategy for the flash separation process. Conclusions are given in Section 5.

2. THERMODYNAMIC LIBRARY

This section briefly discusses the interfaces of the ThermoLib Matlab routines that are used in the heat pump simulations presented in Section 3 and the optimal flash separation presented in Section 4. Furthermore, we present a set of performance tests that illustrate the efficiency of selected library routines. ThermoLib is open-source software available at www.psetools.org and its methods are described by Ritschel et al. (2016).

2.1 ThermoLib Matlab Interface

Fig 1(a) shows a Matlab script that computes molar vapor and liquid enthalpy, entropy, and volume of Freon-12 which is one of the refrigerants that are used in the heat pump simulations. The thermodynamic properties are evaluated at a temperature of $T = 300$ K and a pressure of $P = 1$ MPa. The ThermoLib routine LoadParams must be called before using any other routine. It loads DIPPR parameters and in this case also PR EOS parameters. The two routines PureRealVapHSV and PureRealLiqHSV return the thermodynamic properties together with first and second order temperature and pressure derivatives. Fig. 1(b) shows a Matlab script that computes enthalpy, entropy and volume of a hydrocarbon mixture. ThermoLib requires that the user specifies a set of binary interaction parameters, k_{ij} . These are all zero for hydrocarbons. The properties are evaluated at a temperature of $T = 300$ K and a pressure of $P = 10$ MPa. The two routines MixRealVapHSV and MixRealLiqHSV return the thermodynamic properties, vectors of first order derivatives, and matrices of second order derivatives.

2.2 Computational Performance of ThermoLib

Fig. 2 shows a set of performance tests that evaluate eight of the main routines in ThermoLib. The routines compute vapor and liquid properties of real and ideal mixtures as well as of pure components. Fig. 2(a) compares the efficiency of the library routines to numerical differentiation.

The mixture routines are between 12 and 35 times faster than numerical differentiation, while the pure component routines are around 5 times faster. The speedup is lower for the pure component routines because they only evaluate temperature and pressure derivatives. Fig. 2(b) shows the computation time of the Matlab, C and Mex routines. The C routines are more than a hundred times faster than the Matlab routines and around ten times faster than the Mex routines. The order of magnitude ranges from milliseconds for the Matlab routines to microseconds for the C routines. Fig. 2(c) and 2(d) show the increase in computation time for the Matlab and C routines with respect to the number of components. There is hardly any increase for the Matlab routines, which is due to efficient vectorization, while the increase is close to linear for the C routines.

3. VAPOR-COMPRESSOR CYCLE

In this section we describe a vapor-compression cycle (VCC) for the modeling of heat pumps or refrigerators. Furthermore, we present the effect of ambient conditions on the performance of a heat pump in order to emphasize the need for thermodynamically rigorous computation of the efficiency of heat pumps.

A heat pump is a reversed heat engine that transfers heat from a low temperature zone to one with a higher temperature using mechanical work for compression. Fig. 3 shows a schematic drawing of a VCC with a typical temperature-entropy diagram. The VCC consists of four steps. First the low pressure and low temperature refrigerant (point 1) is evaporated in a heat exchanger producing a saturated vapor (point 2). This saturated low pressure vapor is compressed isentropically by supplying work, W_c , in the compressor to produce a high temperature and high pressure vapor (point 3). Afterwards, heat is released in the condenser at constant pressure, producing a saturated liquid phase (point 4). Finally, the saturated liquid is expanded to its original pressure at isenthalpic conditions in a turbine or in a throttling valve (point 1). The coefficient of performance (COP) provides the overall energy efficiency of a VCC used for either heating, COP_{heat} , or cooling, COP_{cool} . These COPs are defined as

$$COP_{\text{heat}} = \frac{h_3 - h_4}{h_3 - h_2}, \quad (1a)$$

$$COP_{\text{cool}} = \frac{h_2 - h_1}{h_3 - h_2}, \quad (1b)$$

where h_1 , h_2 , h_3 , and h_4 are the enthalpies of the refrigerant in the four points in the cycle. At nominal operating conditions, a COP of a heat pump is often set to 3. This indicates that the extracted heat is 3 times the energy input to the compressor. However, the COP is a nonlinear function of the evaporation temperature, condensation temperature, the active load, and the isentropic efficiency of the compressor. Therefore, thermodynamically correct and accurate modeling of this unit is necessary for reliably determining the performance of a VCC and for realistic optimal control of processes with integrated heat pumps. The dynamics of VCCs are in general much faster than the dynamics of buildings and industrial processes (Halvgaard et al., 2012; Hovgaard et al., 2013; Jensen and Skogestad, 2007; Svensson, 1996; Zhao et al., 2003). Therefore, we implement a static model of a heat pump. This model is based

```

1 %% Thermodynamic properties of Freon-12
2 % Specify components
3 comp = 901; % Freon-12
4
5 % Load parameters (Peng-Robinson EOS)
6 params = LoadParams(comp, 'PR');
7
8 % Specify temperature and pressure
9 T = 300; % Kelvin
10 P = 1e6; % Pascal
11
12 % Compute vapor properties
13 [hv, sv, vv, ...
14 dhvT, dsvT, dvvT, dhvP, dsvP, dvvP, ...
15 d2hvT, d2svT, d2vvt, d2hvP, d2svP, d2vvp, ...
16 d2hvTP, d2svTP, d2vvtP] = ...
17 PureRealVapHSV(T, P, params);
18
19 % Compute liquid properties
20 [hl, sl, vl, ...
21 dhlT, dslT, dvlT, dhlP, dslP, dvlP, ...
22 d2hlT, d2slT, d2vlT, d2hlP, d2slP, d2vlP, ...
23 d2hlTP, d2slTP, d2vlTP] = ...
24 PureRealLiqHSV(T, P, params);
25

```

(a) Computation of thermodynamic properties of Freon-12.

```

1 %% Thermodynamic properties of a hydrocarbon mixture
2 % Specify components
3 comp = [1 % n-C1
4         2 % n-C2
5         7]; % n-C7
6
7 % Load binary interaction parameters
8 kij = zeros(3, 3);
9
10 % Load parameters (Peng-Robinson EOS)
11 params = LoadParams(comp, 'PR', kij);
12
13 % Specify temperature, pressure and mole numbers
14 T = 300; % Kelvin
15 P = 10e6; % Pascal
16 nv = [0.72; 0.07; 0.01]; % kmol
17 nl = [0.08; 0.03; 0.09]; % kmol
18
19 % Compute vapor properties
20 [Hv, Sv, Vv, dHv, dSv, dVv, d2Hv, d2Sv, d2Vv] = ...
21 MixRealVapHSV(T, P, nv, params);
22
23 % Compute liquid properties
24 [Hl, Vl, Sl, dHl, dSl, dVl, d2Hl, d2Sl, d2Vl] = ...
25 MixRealLiqHSV(T, P, nl, params);

```

(b) Computation of thermodynamic properties of a hydrocarbon mixture.

Fig. 1. Examples of Matlab scripts that use ThermoLib routines to compute thermodynamic properties.

on energy conservation equations, and ThermoLib routines are used for the computation of enthalpy, entropy, and saturation pressure as well as their temperature and pressure derivatives. Fig. 4 shows the solution methodology that is used to compute the COP of this static VCC model. For a single component refrigerant, this implementation relies on the ThermoLib routines `PureRealVapHSV` and `PureRealLiqHSV`, whose interfaces were briefly presented in Section 2, and the `PureRealSatTemp` routine.

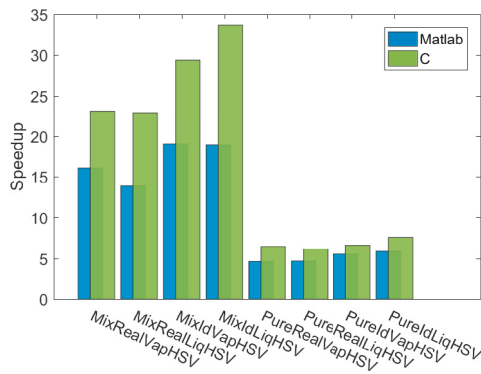
3.1 Simulation Results

This section presents simulations of two frequently encountered operational scenarios and it illustrates that the efficiency of a heat pump is strongly dependent on ambient conditions and the type of the refrigerant.

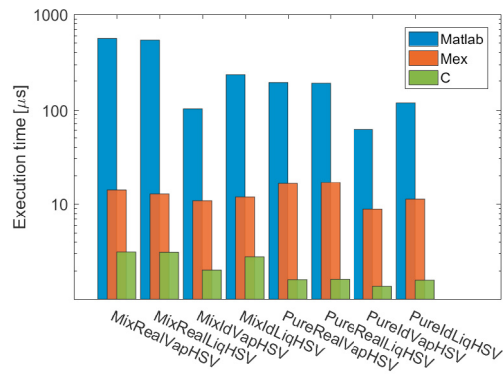
Fig. 5 shows the simulation results for Freon-12 and CO_2 . Fig. 5(a) shows the effect of changes in the evaporation temperature on the performance (COP) of a heat pump. The condenser operates at a fixed 20°C . This scenario frequently occurs in the case of heat pumps for indoor climate control aiming at maintaining a constant room temperature during daily and seasonal changes of the ground temperature. Fig. 5(b) shows the COP as function of the condensation temperature at a fixed evaporation temperature of -10°C . This scenario occurs when heat pumps absorb energy from a constant heat source while the released heat is a manipulated variable. Such scenarios are encountered in many industrial applications, for instance in cryogenic air distillation. The results shown in Fig. 5 illustrate that the COP is significantly affected by changes in the operation parameters and that it also depends on the type of refrigerant. Consequently, the assumption of a constant COP, often used in control applications, is not realistic. Realistic simulations and conclusions require a rigorous thermodynamic library such as ThermoLib.

4. FLASH SEPARATION PROCESS

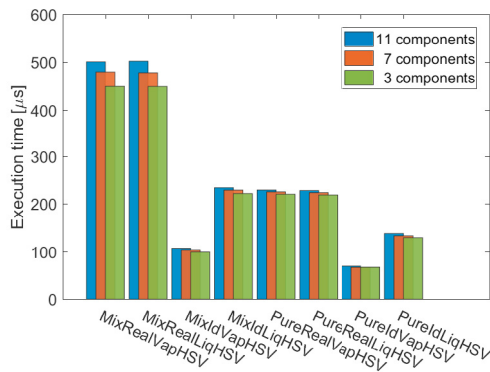
In this section, we present an optimal control strategy for a flash separation process in which H_2S is removed from a gas condensate. The condensate consists of eleven hydrocarbons as well as CO_2 , N_2 , and H_2S . The separation process is assumed to be at vapor-liquid equilibrium. The dynamic optimization of the isoenergetic-isochoric (UV) flash process can be formulated as a bilevel optimization problem (Ritschel et al., 2017). The inner optimization problem is the 2nd law of thermodynamics that says that equilibrium of a closed systems is obtained at maximum entropy of the system. The outer dynamic optimization problem has the differential mass and energy balances in addition to the equilibrium conditions as constraints. The mass and energy balances are affected by a feed stream that supplies the gas condensate to the separator unit, a vapor stream that extracts gas, a liquid stream that extracts liquid, and a heat input. The separator unit is sketched in Fig. 6. We apply a model-based approach to find an optimal control strategy with respect to a given performance measure, while satisfying a set of constraints. We use a single shooting approach which combines dynamic simulation with an iterative optimization algorithm (Binder et al., 2001). Fig. 7 illustrates the solution methodology. Model-based control is advantageous because it can be used for both setpoint tracking and economical control, and because it can handle constraints. However, it is also computationally expensive and therefore it is important to accelerate the computations by applying a gradient-based optimization method. This requires the first and second order derivatives of thermodynamic properties because the first order derivatives appear in the vapor-liquid equilibrium conditions in the model. They are provided by ThermoLib. The gradients of input bound constraints are derived analytically while output bound constraints are incorporated into the objective function using a logarithmic



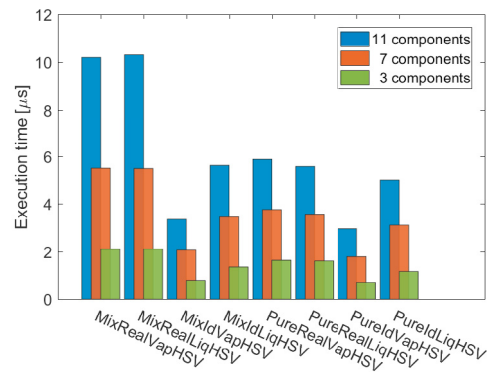
(a) Speedup in computation time compared to numerical forward difference approximations of first and second order derivatives for benzene, toluene and diphenyl.



(b) Computation times of the Matlab, C and Mex routines. Pure component routines evaluate benzene properties and mixture routines evaluate properties of a mixture of benzene, toluene and diphenyl.



(c) Computation times of the Matlab routines for 3 components: n-C₁ to n-C₃, 7 components: n-C₁ to n-C₇, and 11 components: n-C₁ to n-C₁₁.



(d) Computation times of the C routines for 3 components: n-C₁ to n-C₃, 7 components: n-C₁ to n-C₇, and 11 components: n-C₁ to n-C₁₁.

Fig. 2. Performance tests of eight Matlab, C, and Mex routines from ThermoLib. The functionality is related to the routine names as follows. Mix: mixture properties, Pure: pure component properties, Real: real properties, Id: ideal properties, Vap: vapor properties and Liq: liquid properties.

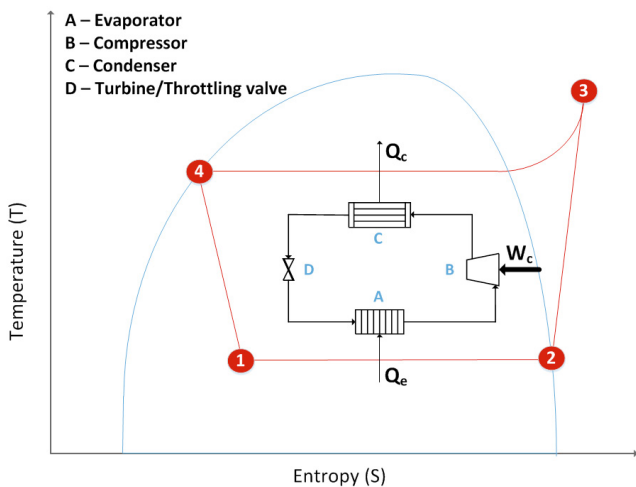


Fig. 3. Schematic representation and temperature-entropy diagram of a vapor-compression cycle.

barrier function. An adjoint method is used for computing the gradients of the objective function (Jørgensen, 2007).

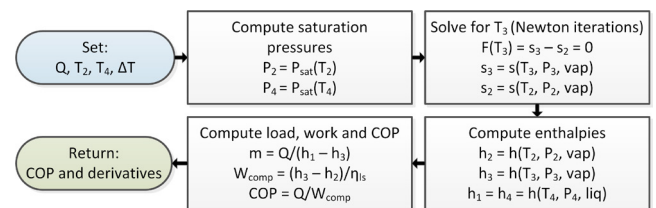


Fig. 4. Solution methodology for simulation of the static model of the vapor-compression cycle. T₁ through T₄ are temperatures, and P₁ through P₄ are pressures.

4.1 Optimal Control Example

We present an example where the gas condensate is processed over a time interval of six hours, i.e. $t \in [t_0, t_f] = [0 \text{ hr}, 6 \text{ hr}]$. During the middle two hours the H₂S content of the feed increases significantly as shown in Fig. 8(a). The manipulated variables are the vapor stream flow rate, F_V , and the heat input, Q . We consider the heat input to be a cooling device such that $Q \leq 0$. The objective is to minimize the amount of energy spent on cooling, i.e. to maximize the functional

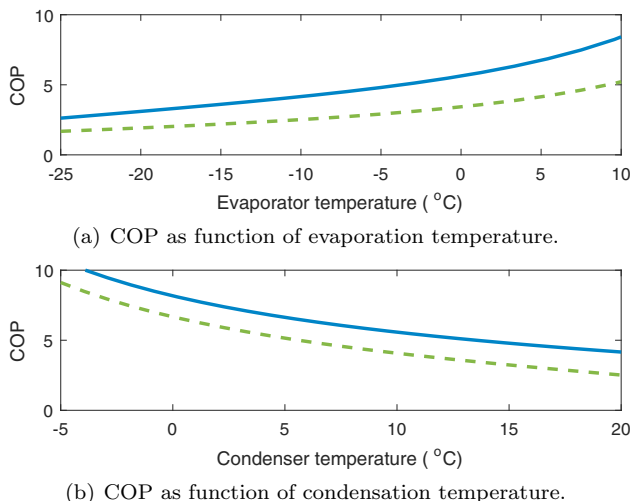


Fig. 5. Heat pump simulations of common operational scenarios. Blue solid: Freon-12. Green dashed: CO₂.

$$\phi = \int_{t_0}^{t_f} Q(t) dt \quad (2)$$

subject to the following constraint on the H₂S vapor mole fraction, $y_{\text{H}_2\text{S}}$

$$y_{\text{H}_2\text{S}}(t) \leq y_{\text{H}_2\text{S}}^{\max}, \quad t \in [t_0, t_f], \quad (3)$$

where $y_{\text{H}_2\text{S}}^{\max} = 0.02$ is an upper bound on the level of impurity of the gas condensate. The liquid stream flow rate is controlled such that the total mass is constant, i.e. $F_L = F_F - F_V$ where $F_F = 12$ kmol/hr is the feed flow rate. We compare the optimal strategy with a reference strategy, which also satisfies the constraint (3). The vapor mole fraction of H₂S is shown in Fig. 8(a), and Fig. 8(b) shows the optimal control strategy (solid blue) and the reference strategy (green dashed). The optimal control solution only applies cooling during the step in the H₂S feed content where it is used to ensure that the constraint (3) is satisfied. The optimal strategy makes active use of the vapor flow rate. However, it processes the same amount of gas as the reference strategy. The energy spent on cooling is decreased from 600 MJ for the reference strategy to 370 MJ for the optimal strategy. This is an improvement of 38%. The single-shooting method requires 165 dynamic simulations to converge to the optimal control strategy shown in Fig. 8(b). The dynamic simulations are the main computational part of the single shooting approach. If the gradients of the objective function were approximated with finite differences it would not be necessary to compute second order derivatives of the thermodynamic properties. However, that would require at least 19,724 dynamic simulations. This is nearly 120 times as many simulations and it is likely that the computation time would increase by two orders of magnitude. This illustrates the potential of using the ThermoLib routines for efficient gradient-based solution of dynamic optimization problems.

5. CONCLUSIONS

We have presented a performance study of the recently developed thermodynamic library, ThermoLib, which is implemented in Matlab and C. The study shows that the computation time is of the order milliseconds for the Matlab routines and microseconds for the C routines.

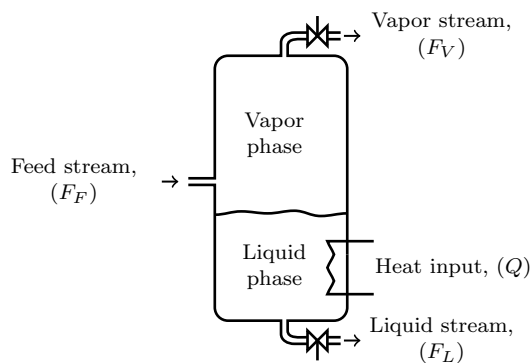


Fig. 6. Sketch of vapor-liquid flash separation unit.

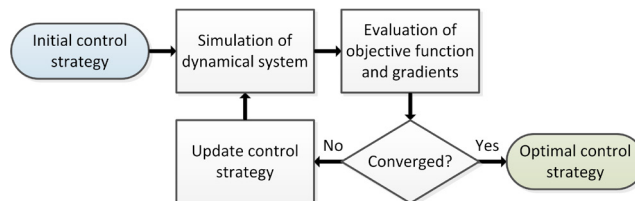


Fig. 7. Flow diagram of a gradient-based single shooting approach for dynamic optimization.

ThermoLib routines evaluate first and second order derivatives analytically which is an order of magnitude faster than numerical differentiation for mixture properties and around 5 times faster for pure component properties. The computation time is nearly constant with respect to the number of components for the Matlab routines and close to linear for the C routines.

Furthermore, we have used ThermoLib in the simulation of a vapor-compression cycle and for optimal control of a flash separation process. The study of the vapor-compression cycle outlined the importance of a rigorous thermodynamic approach for reliable predictions of process performance and economics with respect to different operating conditions and different types of refrigerants. The study demonstrated that assuming a constant coefficient of performance for a compression cycle, as is generally considered in control studies, may lead to significant prediction errors. Furthermore, we have solved a dynamic optimization problem involving a flash process using efficient gradient-based methods based on ThermoLib routines. The dynamic optimization problem involved nonlinear output constraints that were incorporated into the objective function using a barrier function, and the gradients of the objective were computed using an adjoint method. In both examples the availability of first and second order gradients is crucial to the good computational performance.

REFERENCES

- Binder, T., Blank, L., Bock, H. G., Bulirsch, R., Dahmen, W., Diehl, M., Kronseder, T., Marquardt, W., Schlöder, J. P., von Stryk, O., 2001. Introduction to model based optimization of chemical processes on moving horizons. In: Online optimization of large scale systems. Springer, pp. 295–339.
- Chapman, J. W., Lavelle, T. M., May, R. D., Litt, J. S., Guo, T.-H., 2014. Toolbox for the modeling and analy-

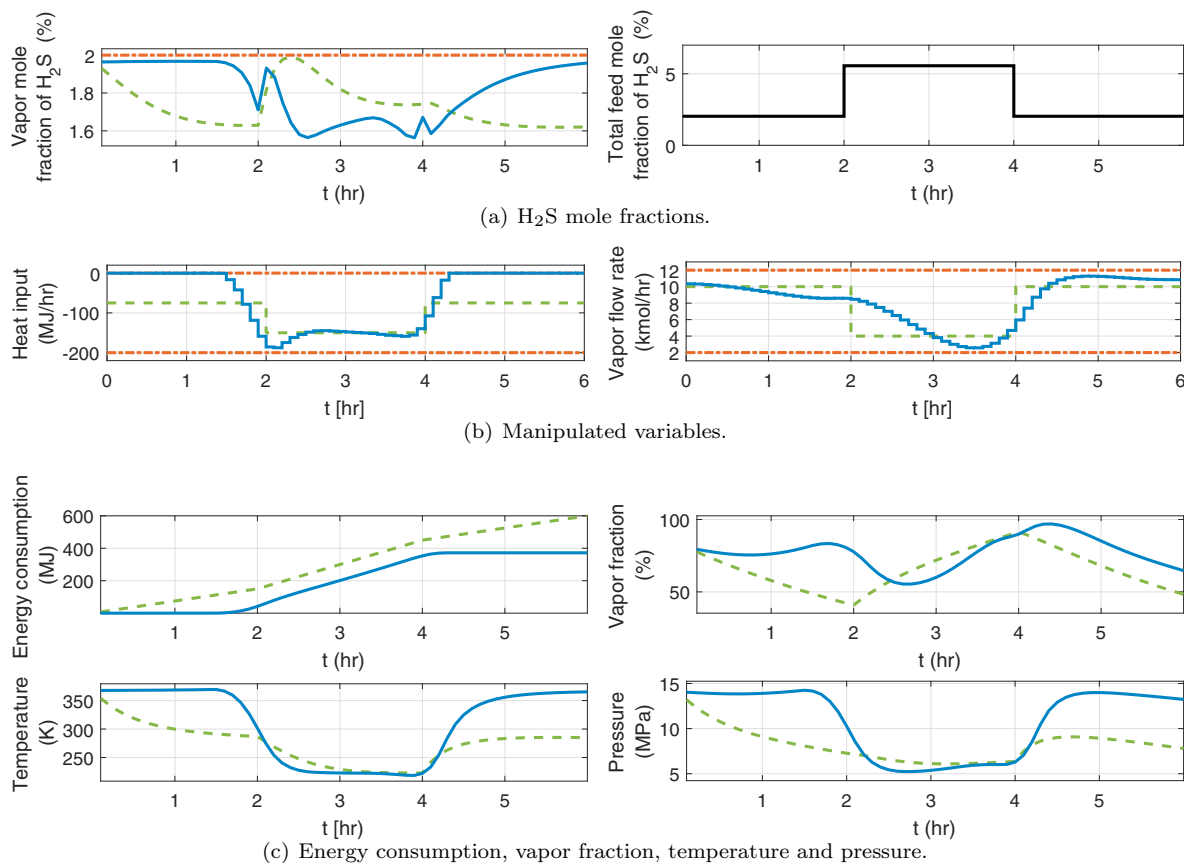


Fig. 8. Optimal flash separation of a gas condensate with a high content of H₂S. Blue solid: Optimal strategy. Green dashed: Reference strategy. Red dash-dotted: Bounds.

- sis of thermodynamic systems (T-MATS) user's guide. Tech. rep., NASA's Glenn Research Center.
- European Heat Pump Association, 2015. European heat pump market and statistics report 2015. Tech. rep., EHPA, Brussels.
- Halvgaard, R., Poulsen, N. K., Madsen, H., Jørgensen, J. B., Jan 2012. Economic model predictive control for building climate control in a smart grid. In: 2012 IEEE PES Innovative Smart Grid Technologies (ISGT). pp. 1–6.
- Hovgaard, T. G., Boyd, S., Larsen, L. F., Jørgensen, J. B., 2013. Nonconvex model predictive control for commercial refrigeration. *International Journal of Control* 86 (8), 1349–1366.
- Jensen, J. B., Skogestad, S., 2007. Optimal operation of simple refrigeration cycles part ii: Selection of controlled variables. *Computers and Chemical Engineering* 31 (12), 1590–1601.
- Jørgensen, J. B., 2007. Adjoint sensitivity results for predictive control, state- and parameter-estimation with nonlinear models. In: *European Control Conference (ECC)*, 2007. pp. 3649–3656.
- Martín, Á., Bermejo, M. D., Mato, F. A., Cocero, M. J., 2011. Teaching advanced equations of state in applied thermodynamics courses using open source programs. *Education for Chemical Engineers* 6 (4), e114–e121.
- Peng, D.-Y., Robinson, D. B., 1976. A new two-constant equation of state. *Industrial & Engineering Chemistry Fundamentals* 15 (1), 59–64.
- Ritschel, T. K. S., Capolei, A., Jørgensen, J. B., 2017. Dynamic optimization of UV flash processes. In: *FOCAPO / CPC* 2017.
- Ritschel, T. K. S., Gaspar, J., Capolei, A., Jørgensen, J. B., 2016. An open-source thermodynamic software library. Tech. Rep. DTU Compute Technical Report-2016-12, Department of Applied Mathematics and Computer Science, Technical University of Denmark.
- Shibata, S. K., Sandler, S. I., 1989. Critical evaluation of equation of state mixing rules for the prediction of high-pressure phase equilibria. *Industrial & engineering chemistry research* 28 (12), 1893–1898.
- Soave, G., 1972. Equilibrium constants from a modified Redlich-Kwong equation of state. *Chemical Engineering Science* 27 (6), 1197–1203.
- Sundman, B., Kattner, U. R., Palumbo, M., Fries, S. G., 2015. OpenCalphad - a free thermodynamic software. *Integrating Materials and Manufacturing Innovation* 4 (1).
- Svensson, M. C., 1996. Model-based optimizing control of a water-to-water heat pump unit. *Modeling, Identification and Control* 17 (4), 279–295.
- Thomson, G. H., 1996. The DIPPR® databases. *International Journal of Thermophysics* 17 (1), 223–232.
- Zhao, L., Zhao, L. L., Zhang, Q., Ding, G. L., 2003. Theoretical and basic experimental analysis on load adjustment of geothermal heat pump systems. *Energy Conversion and Management* 44 (1), 1–9.

Paper III

The Adjoint Method for Gradient-based Dynamic Optimization of UV Flash Processes

Authors:

Tobias K. S. Ritschel, Andrea Capolei, and John Bagterp Jørgensen

Published in:

Computer Aided Chemical Engineering 40, 2071–2076, 2017.

DOI: <https://doi.org/10.1016/B978-0-444-63965-3.50347-0>.

The Adjoint Method for Gradient-based Dynamic Optimization of UV Flash Processes

Tobias K. S. Ritschel^a, Andrea Capolei^a and John B. Jørgensen^{a*}

^a*Department of Applied Mathematics & Center for Energy Resources Engineering, Technical University of Denmark, DK-2800 Kgs. Lyngby, Denmark
jbjo@dtu.dk*

Abstract

This paper presents a novel single-shooting algorithm for gradient-based solution of optimal control problems with vapor-liquid equilibrium constraints. Dynamic optimization of UV flash processes is relevant in nonlinear model predictive control of distillation columns, certain two-phase flow problems, and oil reservoir production with significant compositional and thermal effects. Gradients are computed with the adjoint method and we use various optimization software (fmincon, IPOPT, KNITRO, and NPSOL) for the numerical optimization. We present computational results for a non-ideal five-component flash process which demonstrate the importance of the optimization solver, the compiler, and the linear algebra software for the efficiency of dynamic optimization of UV flash processes.

Keywords: Optimization, Optimal Control, Differential-Algebraic Equations, Vapor-Liquid Equilibrium

1. Introduction

Dynamic optimization of vapor-liquid equilibrium (VLE) processes is used in control algorithms for distillation columns, certain two-phase flow problems, and oil reservoir production with significant thermal and compositional effects. Computationally efficient algorithms for dynamic optimization of UV flash processes are thus relevant for such applications. The UV flash problem is sometimes referred to as the isoenergetic-isochoric flash problem or as the UV_n flash problem since the internal energy, U , the volume, V , and the total amount of moles, n , are specified. The equilibrium composition is determined using the second law of thermodynamics (Michelsen, 1999) which states that the entropy of a closed system is maximal. The UV flash problem is different from the more commonly known PT flash problem. However, the UV flash can be shown to be equivalent to a UV-constrained PT flash problem, i.e. with constraints on the internal energy, U , and the volume, V . Nested approaches are often used in the simulation and dynamic optimization of VLE processes. In such approaches a PT flash problem is solved in an inner loop while the specified properties, internal energy, U , and volume, V , are converged in an outer loop. Nested approaches are, however, computationally inefficient. In this paper we present an algorithm for dynamic optimization which is based on the single-shooting method and the adjoint method for the computation of gradients (Jørgensen, 2007). The algorithm has previously been considered by Ritschel et al. (2017) for ideal

mixtures, whereas we consider non-ideal mixtures in this work. There are alternatives to the single-shooting method such as the multiple-shooting method (Capolei and Jørgensen, 2012) and the simultaneous method (Biegler, 2010). We present numerical results for a five-component dynamical UV flash process as well as the computational performance of Matlab and C implementations using different optimization software, compilers, and linear algebra software. The thermodynamic library ThermoLib (www.psetools.org) is used for the evaluation of the thermodynamic functions.

2. Optimal control problem

We consider an optimal control problem (OCP) in the form

$$\min_{[x(t);y(t);z(t)]_{t_0}^{t_f}, \{u_k\}_{k \in \mathcal{N}}} \phi = \int_{t_0}^{t_f} \Phi(y(t), u(t), d(t)) dt, \quad (1a)$$

$$\text{subject to } x(t_0) = \hat{x}_0, \quad (1b)$$

$$G(x(t), y(t), z(t)) = 0, \quad t \in \mathcal{T}, \quad (1c)$$

$$\dot{x}(t) = F(y(t), u(t), d(t)), \quad t \in \mathcal{T}, \quad (1d)$$

$$u(t) = u_k, t \in [t_k, t_{k+1}[, \quad k \in \mathcal{N}, \quad (1e)$$

$$d(t) = \hat{d}_k, t \in [t_k, t_{k+1}[, \quad k \in \mathcal{N}, \quad (1f)$$

$$\{u_k\}_{k \in \mathcal{N}} \in \mathcal{U}, \quad (1g)$$

where $x(t)$ is the state vector, $y(t)$ is a vector of algebraic variables, and $z(t)$ is a vector of adjoint algebraic variables. In this optimal control problem the estimated initial state, \hat{x}_0 , and the estimated disturbance variables, $\{\hat{d}_k\}_{k \in \mathcal{N}}$, are considered parameters. The inputs, $\{u_k\}_{k \in \mathcal{N}}$, are independent decision variables, and $[x(t); y(t); z(t)]_{t_0}^{t_f}$ is a vector of dependent decision variables. The indices of the control intervals are $\mathcal{N} = \{0, 1, \dots, N-1\}$, and the time horizon is $\mathcal{T} = [t_0, t_f]$. The algebraic equations in Eq. (1c) are used to model equilibrium conditions for VLE processes, and the differential equations in Eq. (1d) are based on conservation principles. We assume that it is possible to solve the equilibrium conditions $G(x(t), y(t), z(t)) = 0$ for $y(t) = y(x(t))$ and $z(t) = z(x(t))$ when $x(t)$ is given. We define the objective function

$$\psi = \psi(\{u_k\}_{k \in \mathcal{N}}; \hat{x}_0, \{\hat{d}_k\}_{k \in \mathcal{N}}) = \left\{ \phi : (1b)-(1f) \right\}, \quad (2)$$

which is computed as the objective function ϕ in Eq. (1a) using the solution of the differential-algebraic equations (DAEs) in Eq. (1c)-(1d) with the initial condition $x(t_0) = \hat{x}_0$, $u(t) = u_k$ for $t \in [t_k, t_{k+1}[$ and $k \in \mathcal{N}$, and $d(t) = \hat{d}_k$ for $t \in [t_k, t_{k+1}[$ and $k \in \mathcal{N}$, i.e. Eq. (1b) and Eq. (1e)-(1f). The OCP in Eq. (1) can thus be expressed as the following finite-dimensional constrained optimization problem

$$\min_{\{u_k\}_{k \in \mathcal{N}}} \psi = \psi(\{u_k\}_{k \in \mathcal{N}}; \hat{x}_0, \{\hat{d}_k\}_{k \in \mathcal{N}}), \quad (3a)$$

$$\text{s.t. } \{u_k\}_{k \in \mathcal{N}} \in \mathcal{U}. \quad (3b)$$

The constraints in Eq. (3b) are often bound constraints, $u_{\min} \leq u \leq u_{\max}$, and linear constraints, $b_l \leq Au \leq b_u$, where $u = [u_0; u_1; \dots; u_{N-1}]$. Efficient gradient-based optimization

algorithms for solving the optimization problem in Eq. (3) require the evaluation of both the objective function, ψ , and its gradients, $\{\nabla_{u_k} \psi\}_{k \in \mathcal{N}}$. Such computations are based on numerical integration of the DAE system in Eq. (1c)-(1d) and of the integral in the objective function ϕ in Eq. (1a).

2.1. The equilibrium constraints

In this work we consider equilibrium processes that can be described as the solution to the following parametric equality constrained optimization problem

$$\min_y f(y), \tag{4a}$$

$$\text{s.t. } g(y) = x, \tag{4b}$$

$$h(y) = 0. \tag{4c}$$

The Lagrange function associated with the optimization problem in Eq. (4) is

$$\mathcal{L}(y, \eta, \mu; x) = f(y) - \eta^T (g(y) - x) - \mu^T h(y), \tag{5}$$

where η and μ are Lagrange multipliers associated with the constraints in Eq. (4b) and (4c), respectively. The Karush-Kuhn-Tucker (KKT) conditions (or first order optimality conditions) for a minimizer ($y = y(x), \eta = \eta(x), \mu = \mu(x)$) are

$$\nabla_y \mathcal{L}(y, \eta, \mu; x) = \nabla f(y) - \nabla g(y) \eta - \nabla h(y) \mu = 0, \tag{6a}$$

$$\nabla_\eta \mathcal{L}(y, \eta, \mu; x) = -(g(y) - x) = 0, \tag{6b}$$

$$\nabla_\mu \mathcal{L}(y, \eta, \mu; x) = -h(y) = 0. \tag{6c}$$

We introduce the vector of adjoint algebraic variables $z = [\eta; \mu]$ such that the KKT system in Eq. (6) can be expressed as the algebraic equations in Eq. (1c).

3. Numerical algorithm

The DAE system in Eq. (1c)-(1d) is stiff and we therefore discretize the differential equations with Euler's implicit method. Alternatively, ESDIRK methods (Kristensen et al., 2004) or BDF based methods (Barton and Lee, 2002) can be used. Furthermore, we describe the computation of $\{\nabla_{u_k} \psi\}_{k \in \mathcal{N}}$ by an adjoint method. These gradients may also be computed with a forward method. We define $w = [x; y; z]$ and the residual equations

$$R_{k+1} = R_{k+1}(w_{k+1}; x_k, u_k, \hat{d}_k) = \begin{bmatrix} D_{k+1}(x_{k+1}, x_k, y_{k+1}, u_k, \hat{d}_k) \\ G(x_{k+1}, y_{k+1}, z_{k+1}) \end{bmatrix} = 0, \quad k \in \mathcal{N}, \tag{7}$$

where $D_{k+1} = x_{k+1} - x_k - \Delta t_k F(y_{k+1}, u_k, \hat{d}_k)$ are the discretized differential equations. The residual equations, $R_{k+1} = 0$, are solved by an inexact Newton method, i.e. by solving a sequence of linear systems

$$w_{k+1}^{m+1} = w_{k+1}^m - (M_{k+1}^m)^{-1} R_{k+1}(w_{k+1}^m), \quad M_{k+1}^m \approx \frac{\partial R_{k+1}}{\partial w_{k+1}}(w_{k+1}^m) \tag{8}$$

until a given convergence criteria is satisfied. The objective function, ϕ , in Eq. (1a) is approximated by a discrete sum, and the residual equations in Eq. (7) are substituted for the DAE system in the objective function, ψ , in Eq. (2) in order to obtain

$$\psi = \psi(\{u_k\}_{k \in \mathcal{N}}; \hat{x}_0, \{\hat{d}_k\}_{k \in \mathcal{N}}) = \left\{ \begin{aligned} \phi = \sum_{k \in \mathcal{N}} \Delta t_k \Phi(y_{k+1}, u_k, \hat{d}_k) : \end{aligned} \right. \quad (9a)$$

$$x_0 = \hat{x}_0, \quad (9b)$$

$$R_{k+1}(w_{k+1}; x_k, u_k, \hat{d}_k) = 0, \quad k \in \mathcal{N} \quad \left. \right\}. \quad (9c)$$

The adjoints, $\{\lambda_k\}_{k=1}^N$, are computed in a backwards manner by solving the equations

$$\left(\frac{\partial R_N}{\partial w_N} \right)^T \lambda_N = -\Delta t_N \nabla_{w_N} \Phi_{N-1}, \quad (10a)$$

$$\left(\frac{\partial R_k}{\partial w_k} \right)^T \lambda_k = - \left(\frac{\partial R_{k+1}}{\partial w_k} \right)^T \lambda_{k+1} - \Delta t_k \nabla_{w_k} \Phi_{k-1}, \quad k = N-1, \dots, 1. \quad (10b)$$

The gradients of ψ with respect to the inputs, $\{\nabla_{u_k} \psi\}_{k \in \mathcal{N}}$, are then computed by

$$\nabla_{u_k} \psi = \Delta t_k \nabla_{u_k} \Phi_k + \left(\frac{\partial R_{k+1}}{\partial u_k} \right)^T \lambda_{k+1}, \quad k \in \mathcal{N}. \quad (11)$$

4. Dynamic UV flash example

The energy and mass balance of a flash unit may be expressed by the following differential equations where kinetic and potential energy are neglected

$$\dot{U}(t) = H_F^v(t) + H_F^l(t) - H_V(t) - H_L(t) + Q(t), \quad (12a)$$

$$\dot{n}_i(t) = f_{F,i}^v(t) + f_{F,i}^l(t) - v_i(t) - l_i(t), \quad i = 1, \dots, N_C. \quad (12b)$$

U is the internal energy, and n_i is the total holdup of the i 'th component. H_F^v and H_F^l are the vapor and liquid enthalpies of the feed, and H_V and H_L are the enthalpies of the vapor and liquid streams. $f_{F,i}^v$ and $f_{F,i}^l$ are the vapor and liquid component flow rates of the feed, and v_i and l_i are the component flow rates of the vapor and liquid streams. Q is a heat input. The conservation equations in Eq. (12) are in the form of the differential equations in Eq. (1d). The UV flash problem is a mathematical statement of the second law of thermodynamics, i.e. the entropy of a closed system, S , is maximal, and it is formulated as the following optimization problem (Michelsen, 1999)

$$\max_{T, P, n^v, n^l} \quad S = S^v(T, P, n^v) + S^l(T, P, n^l), \quad (13a)$$

$$\text{s.t.} \quad U^v(T, P, n^v) + U^l(T, P, n^l) = U, \quad (13b)$$

$$V^v(T, P, n^v) + V^l(T, P, n^l) = V, \quad (13c)$$

$$n_i^v + n_i^l = n_i, \quad i = 1, \dots, N_C. \quad (13d)$$

Superscripts v and l denote properties of the vapor and liquid phase, respectively. The optimization problem in Eq. (13) is in the form of the parametric optimization problem

Table 1: Absolute computation time (in seconds) and speedup compared to the pure Matlab implementation of the solution of the OCP. The C code has been compiled with GCC compilers (and Netlibs BLAS/LAPACK) and with Intel compilers (and Intel MKL).

	fmincon	fmincon	IPOPT	KNITRO	NPSOL	fmincon	IPOPT	KNITRO	NPSOL
Compilers		GCC	GCC	GCC	GCC	Intel	Intel	Intel	Intel
Prog. lang.	Matlab	C	C	C	C	C	C	C	C
Iterations	156	156	328	117	150	156	328	117	150
Func. eval.	173	173	1364	121	152	173	1364	121	152
Absolute	137.803	37.525	6.868	6.654	7.002	37.929	6.437	8.149	2.481
Speedup	1.0	3.7	20.1	20.7	19.7	3.6	21.4	16.9	55.5

in Eq. (4). The thermodynamic functions are computed using the Peng-Robinson equation of state. We consider the flash separation of methane, ethane, propane, heptane, and hydrogen sulfide (H_2S). The flash unit is cooled, i.e. $Q \leq 0$, and the objective is to minimize the energy consumption, i.e. to maximize Q , while satisfying an upper bound on the H_2S vapor mole fraction. Fig. 1 shows the optimal strategy (solid) together with a reference strategy (dashed) which also satisfies the constraints (dash-dotted). The manipulated inputs are the vapor flow rate, the liquid flow rate, and the heat input, Q . The optimal strategy reduces the energy consumption by 26 % as compared to the open-loop reference strategy. The constraints on the H_2S vapor mole fraction and on the overall vapor fraction are incorporated into the objective function with logarithmic barrier functions.

Table 1 shows several performance indicators for the solution of the OCP by means of the single-shooting algorithm using various optimization software, compilers, and linear algebra software. We use the efficiency of a pure Matlab implementation, which uses fmincon, as a benchmark for the remaining implementations. The pure Matlab implementation solves the OCP in 137.8 s. A mixed implementation, which uses fmincon and C routines for the simulation and gradient computations, is only four times faster than the pure Matlab implementation. This indicates that the efficiency of the optimization software is important to the overall performance of the algorithm. IPOPT, KNITRO, and NPSOL perform equally well when GCC compilers and Netlibs BLAS/LAPACK are used. The Intel compilers and Intels linear algebra library (Intel MKL) only improve the performance when NPSOL is used. This suggests that Intels Fortran compiler (ifort) is significantly more efficient than GCCs Fortran compiler (gfortran) and that Intels C compiler (icc) is as efficient as GCCs C compiler (gcc) for the present implementations.

5. Conclusions

We have presented a single-shooting algorithm for gradient-based dynamic optimization of flash processes. The algorithm solves the differential equations and the equilibrium conditions simultaneously. An example demonstrates the capabilities of the algorithm for output-constrained economical control of a flash unit. Furthermore, a performance study has shown that C implementations of the single-shooting algorithm are significantly more efficient than a pure Matlab implementation. There is little or no improvement in performance when Intels C compiler is used for the implementations using fmincon, IPOPT, and KNITRO. Intels C and Fortran compilers do, however, improve the performance of the implementation using NPSOL by a factor of almost 3. The most efficient implementation solved the OCP in 2.5 s which is 55.5 times faster than the pure Matlab implementation.

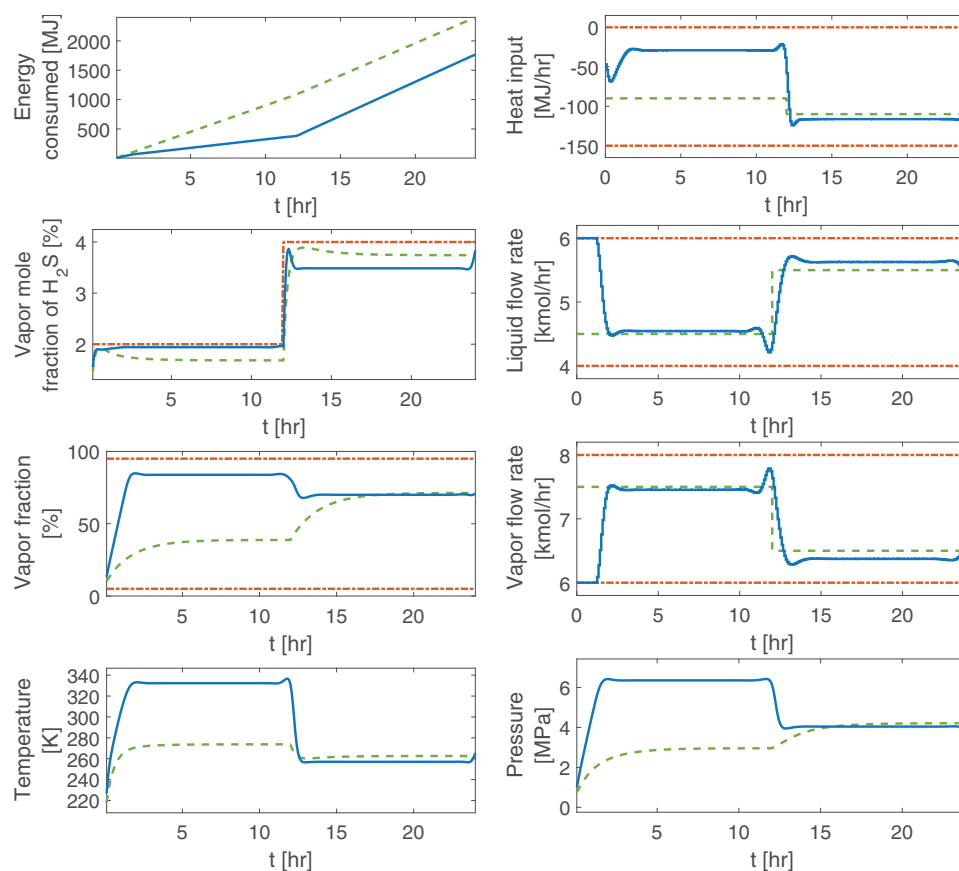


Figure 1: Minimal cooling strategy. Solid: Optimal strategy. Dashed: Reference strategy. Dash-dotted: Bounds.

6. Acknowledgments

The authors acknowledge the financial support from the Danish Advanced Technology Foundation (OPTION; 63-2013-3).

References

- Barton, P. I., Lee, C. K., 2002. Modeling, simulation, sensitivity analysis, and optimization of hybrid systems. *ACM Transactions on Modeling and Computer Simulation*, 256–289.
- Biegler, L. T., 2010. *Nonlinear programming: concepts, algorithms, and applications to chemical processes*. Vol. 10. SIAM.
- Capolei, A., Jørgensen, J., 2012. Solution of constrained optimal control problems using multiple shooting and ESDIRK methods. In: *Proceedings of the 2012 American Control Conference*. American Control Conference. pp. 295–300.
- Jørgensen, J. B., 2007. Adjoint sensitivity results for predictive control, state- and parameter-estimation with nonlinear models. In: *European Control Conference (ECC)*, 2007. pp. 3649–3656.
- Kristensen, M. R., Jørgensen, J. B., Thomsen, P. G., Jørgensen, S. B., 2004. An ESDIRK method with sensitivity analysis capabilities. *Computers & Chemical Engineering* 28 (12), 2695–2707.
- Michelsen, M. L., 1999. State function based flash specifications. *Fluid Phase Equilibria* 158–160, 617–626.
- Ritschel, T. K. S., Capolei, A., Jørgensen, J. B., 2017. Dynamic optimization of UV flash processes. In: *FOCAPO / CPC 2017*.

Paper IV

An Efficient and Rigorous Thermodynamic Library and Optimal-control of a Cryogenic Air Separation Unit

Authors:

Jozsef Gaspar, Tobias K. S. Ritschel, and John Bagterp Jørgensen

Published in:

Computer Aided Chemical Engineering 40, 1543–1548, 2017.

DOI: <https://doi.org/10.1016/B978-0-444-63965-3.50259-2>.

An efficient and rigorous thermodynamic library and optimal-control of a cryogenic air separation unit

Jozsef Gaspar^{a*}, Tobias K. S. Ritschel^a and John Bagterp Jørgensen^a

^a*Department of Applied Mathematics and Computer Science, Technical University of Denmark, DK-2800 Kgs. Lyngby, Denmark*
joca@dtu.dk

Abstract

Cryogenic air separation (CAS) is the leading technology for large scale production of pure N₂, O₂ and Ar. This process is very electric-energy intensive; thus it is a likely candidate for load balancing of power stations in a smart grid. This type of intermittent operation of CAS, requires a non-linear model based control to achieve optimal techno-economic performance. Accordingly, this work presents a computationally efficient and novel approach for solving a tray-by-tray equilibrium model and its implementation for open-loop optimal-control of a cryogenic distillation column. Here, the optimisation objective is to reduce the cost of compression in a volatile electricity market while meeting the production requirements, i.e. product flow rate and purity. This model is implemented in Matlab and uses the ThermoLib rigorous thermodynamic library. The present work represents a first step towards plant-wide dynamic modelling and smart control of a cryogenic distillation plant.

Keywords: Cryogenic air separation, load balancing, distillation column, dynamic modeling, optimal-control.

1. Introduction

Cryogenic air separation (CAS) is the leading and the most mature technology to produce gaseous and liquid O₂, N₂, and Ar for various industries, e.g. steel, petrochemical, fertilizer, coal, liquefaction, etc. (Smith and Klosek, 2001). This process is very energy intensive with electricity being the major operating cost. However, the electricity price varies significantly from hour to hour during a day due to the increasing share of renewable energy and due to changes in industrial and domestic demands (Miller et al., 2008). Therefore, intermittent operation with large changes in the production rate and quick shut-down/start-up can drastically reduce the operating cost of cryogenic air separation. Process non-linearities are highly relevant for these operating conditions, and some type of non-linear control will be essential to achieve optimal techno-economic performance.

Linear and non-linear model predictive control (MPC) had widely spread as state-of-the-art technology for improving the economic benefits and controllability of CAS. Linear MPC has proven to be successful for achieving high product purities and stable control but only over a limited range of production rates (Roffel et al., 2000; Mezou, 2000). Non-linear MPC (NMPC) has also been investigated for operation of cryogenic distillation during load changes. Chen et al. (2010) implemented a NMPC and evaluated its perfor-

mance for $\pm 30\%$ changes in the production rate demand. They demonstrated that NMPC has better performance compared to linear MPC. Huang and Biegler (2012) presented an economic NMPC that minimizes the cost of air separation for a day ahead respectively for a real-time-pricing electricity markets. They showed an approximately 10% cost reduction when using economic NMPC compared to a set-point tracking NMPC. These studies generally rely on some kind of reduced order/simplified model since NMPC is computationally intensive, limiting its potential for on-line control of CAS.

Several studies used models of moderate complexity in the dynamic design and performance analysis of air separation units. White et al. (1996) minimized the transition time between two steady-state operating points using compartmental model for dynamic optimization. Schenk et al. (2002) considered integrated design and control of a cryogenic plant by formulating a dynamic optimization problem to improve the cost of separation. Sirdeshpande et al. (2005) formulated a mixed-integer problem to determine cost-optimal process configuration of CAS and they analysed the flexibility of the optimal design for different production rates. This study is based on regressed algebraic model from Aspen HYSYS. Later, Cao et al., 2015 improved the agility of a cryogenic N_2 plant to successfully respond to $\pm 20\%$ demand changes by solving a open-loop optimal-control problem. However, there is still a great need to understand and further improve the agility of a cryogenic plant in presence of large fluctuations.

This study presents a dynamic distillation column model and the open-loop optimal-control of a cryogenic air separation column. The control objective is to minimize the cost of air separation while assuring the desired purity in a volatile energy market, i.e. we use an hour-by-hour pricing scheme with an economic objective function, to minimize the future daily cost while meeting the production demands. This study is performed in Matlab using an open-source thermodynamic library (ThermoLib, www.psetools.org) for thermal properties and phase equilibrium. In addition, we show that ThermoLib greatly improves the computational load by using analytical derivatives of the thermal properties.

2. Cryogenic air separation

The development of a dynamic model for a cryogenic distillation column is the focus of this section. This low-temperature distillation column is the main unit of a cryogenic air separation plant (CAS) and it involves: (a) compression and cooling of the air feed, (b) boiling of the O_2 rich phase at the bottom and (c) condensation of the N_2 rich phase at the top. The development of a dynamic model for this column is the focus of this section.

2.1. Dynamic model for cryogenic air distillation

The distillation column is a tray-by-tray equilibrium model, consisting of differential equations for the mass and energy balances around each tray and a set of algebraic equations for phase equilibrium, physical properties and the boundary conditions. In the development of the model we assumed: (a) perfect mixing in both phases and on all stages, (b) thermal and thermodynamic equilibrium between phases and (c) constant molar liquid/gas flows. Here, we apply a UV-flash instead of the more common PT-flash that occurs in steady-state optimization problems. The UV-flash corresponds to the rigorous description of the second law of thermodynamics, i.e. the entropy of a closed system is maximal at equilibrium (Ritschel, 2017).

The model consists of N stages (trays) where the first stage represents the reboiler and the last stage corresponds to the condenser. Therefore, the mass and energy conservation equations for stages $j = 2, \dots, N - 1$ are:

$$\dot{U}_j(t) = H_{j+1}^L(t) + H_{j-1}^V(t) - H_j^L(t) - H_j^V(t) + H_{Feed}(t), \quad (1a)$$

$$\dot{n}_{i,j}(t) = l_{i,j+1}(t) + v_{i,j-1}(t) - l_{i,j}(t) - v_{i,j}(t) + f_{i,Feed}(t). \quad (1b)$$

U is internal energy and n_i is the total holdup of component i . H^L and H^V are the enthalpies of the liquid respectively vapor streams. l_i and v_i represent the liquid respectively the vapor flow rates. $f_{i,Feed}$ is the feed flow rate of component i and H_{Feed} is the enthalpy of the feed. Note, $f_{i,Feed}$ and H_{Feed} are zero for all stages excluding feed stages.

The reboiler and the condenser behave like normal trays with the addition/removal of heat. The mass and energy conservation equations for the reboiler ($j = 1$) are:

$$\dot{U}_1(t) = H_2^L(t) - H_1^L(t) - H_B^V(t) + Q_R(t), \quad (2a)$$

$$\dot{n}_{i,j}(t) = l_{i,2}(t) - l_{i,1}(t) - B_i(t), \quad (2b)$$

The conservation equations for the condenser ($j = N$) are:

$$\dot{U}_1(t) = H_{N-1}^V(t) - H_j^V(t) - H_R^L(t) + Q_C(t), \quad (3a)$$

$$\dot{n}_{i,j}(t) = v_{i,N-1}(t) - v_{i,N}(t) - R_i(t). \quad (3b)$$

B_i is the boil-up flow rate of component i ; R_i is the reflux flow rate of component i ; Q_R and Q_C are the reboiler respectively condenser heat duties. These are manipulated variables. For more details regarding the development of a dynamic column model, we refer to Kröner et al. (2001) and Roffel et al. (2000).

3. Optimal-control formulation

The objective of this study is to find a set of manipulated variables for the cryogenic distillation unit over a given scheduling horizon that minimizes the cost of compression while meeting production demands, i.e. producing a minimum amount of N_2 and O_2 per day with a predefined purity of 90% N_2 respectively 75% O_2 . This optimal-control problem (OCP) is formulated as follows:

$$\min_{u(t)} \phi = \int_{t_0}^{t_f} c(t)W_c(u(t), d(t))dt \quad (4a)$$

$$\text{s.t. } x(t_0) = x_0, \quad (4b)$$

$$G(x(t), y(t)) = 0, \quad t \in [t_0, t_f] \quad (4c)$$

$$\dot{x}(t) = F(y(t), u(t), d(t)), \quad t \in [t_0, t_f] \quad (4d)$$

$$h(x(t), y(t), u(t), d(t)) \leq 0, \quad t \in [t_0, t_f] \quad (4e)$$

$$h_f(x(t_f), y(t_f), u(t_f), d(t_f)) \leq 0, \quad (4f)$$

where $x(t)$ is a vector of the total moles and the internal energy at each stage; x_0 is the initial state vector; $y(t)$ consists of the temperature, pressure, vapor mole numbers, and liquid mole numbers on each stage. $u(t)$ represent the manipulated variables: the feed

flow rate, reflux flow rate, boil-up flow rate and reboiler/condenser duty. $d(t)$ are the disturbance variables, i.e. feed composition and feed vapor fraction. The objective function Φ in Eq. (4a) describes the cost of feed compression over the entire control horizon $[t_0, t_f]$. The electricity price enters this optimization as the cost coefficient, $c(t)$. The compression work, W_c depends on the feed flow rate and discharge pressure (Smith and Van Ness, 1987).

The algebraic equation (4c) represents the phase equilibrium conditions on each stage, and the differential equations in eq. (4d) represents the conservation of mass and energy on each stage, eqs. (1) - (3). Equation (4e) represents a set of constraints to regulate the system behaviour during transition. These are: lower bound constraint on the N_2 composition in the condenser; lower bound constraint on the O_2 composition in the bottom stage respectively lower and upper bound constraints on the vapor and liquid holdup on each stage. Furthermore, we include a set of terminal constraints, eq. (4f) in order to assure a minimum daily production of N_2 and O_2 .

3.1. Solution methodology

The optimal-control problem in eq. (4) is solved with a single shooting algorithm which uses an adjoint method for gradient computations. The single shooting algorithm requires the numerical solution of the stiff differential-algebraic initial value problem in eq. (4b)-(4f). The differential equation (4d) are discretized using the Euler's implicit method. The discretized differential equations and the algebraic equations are solved simultaneously with an inexact Newton method. This is an important feature for dynamic optimization applications as it yields faster solution than with a nested numerical integration scheme (Ritschel, 2017).

The thermal properties and the liquid-vapor phase equilibrium, together with analytical temperature and pressure derivatives are calculated with the ThermoLib open-source thermodynamic library (www.psetools.org). The use of analytical derivatives is essential for dynamic simulation and gradient based optimization. We compared the efficiency of the implemented analytical first and second order derivatives to numerical forward difference approximations for: (a) 79% N_2 + 21% O_2 and (b) 78.09% N_2 + 20.93% O_2 + 0.93% Ar + 0.04% CO_2 . The results outlined that the analytical derivatives are roughly 5 times faster than numerical differentiation for pure component calculations and they are approximately 10 times faster for binary mixture, i.e. N_2 + O_2 . The results also showed that the gain in computational efficiency increases greatly with the number of components: the analytical derivatives are 17 to 25 times more efficient than numerical forward differentiation for a quaternary mixture.

4. Case study

To illustrate the potential of the economic optimal-control formulation for a cryogenic distillation unit, we simulate a two scenarios using hour-by-hour electricity prices from Nordpool, the Nordic power exchange market. We use a 24 hours scheduling horizon with an optimal control signal every 12 minutes. The simulated unit corresponds to a distillation column with a capacity of 3 t/hr air. In this study, the air feed consists of 79% N_2 + 21% O_2 and it is compressed to 324 kPa at $-180^\circ C$. The column model has 15 stages, the 8th stage being the feed-stage. It is important to note that we consider a

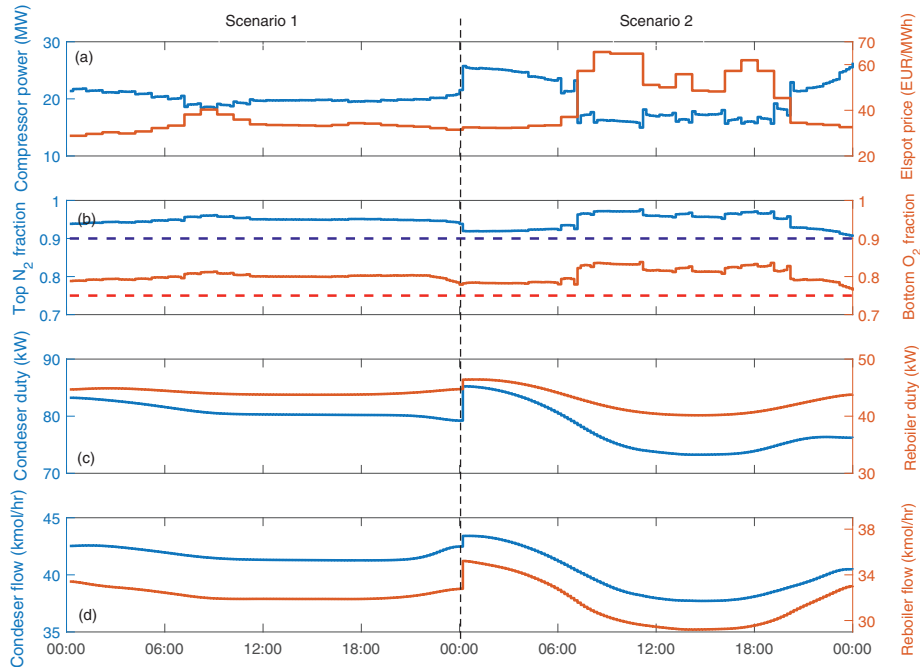


Figure 1: Optimal open-loop performance of CAS with fluctuating electricity price

nitrogen-limited distillation column, i.e. the production is driven by the nitrogen demand.

Figure 1 shows the market electricity price, the compressor power, and the optimal open-loop schedule for the manipulated variables. Scenario 1 corresponds to a case with excess of wind energy when the electricity price is relatively low and stable, 28-40 EUR/MWh. Scenario 2 reflects a case with largely fluctuating energy demand (32-65 EUR/MWh). This scenario may occur during significant increase in the domestic and industrial electricity demand or when there is not enough wind for electricity generation.

Figure 1a shows the variation of the electricity price and the corresponding optimal compressor power. This figure shows how the compressor power (thus the feed flow rate) increases during periods with lower electricity price and vice-versa. In addition, Figure 1b shows that the N₂ and O₂ product purity are always above their lower bounds. Figure 1c and 1d present the trajectory for the other manipulated variables, i.e. reflux flow, boil-up flow and condenser/reboiler duty. These figures show how the plant smoothly changes the process inputs during the 24 hours simulation window, without large fluctuations. This adds to the robustness of the numerical optimization routine and is important from an operational point of view. Note, the abrupt change between the two scenarios is because a new open-loop optimization period is started.

Therefore, the volatility of the energy market can be used to reduce the cost of air separation by shifting the energy consumption to periods with low electricity prices and still maintaining product requirements. Furthermore, intermittent operation of CAS is a viable option for balancing of demand and production of electricity in a smart grid.

5. Conclusions

In this paper, we presented a dynamic model for cryogenic air distillation and we demonstrated the potential of open-loop optimal-control for reducing the cost of air separation in a volatile electricity market. The dynamic model and the optimization algorithm uses the ThermoLib thermodynamic library to compute thermal properties and their analytical derivatives. We showed that analytical derivatives lead to large decrease of the simulation time, i.e. for a binary mixture the analytical derivatives are up to 12 times faster than numerical differentiation. Using real market electricity prices, we demonstrated that open-loop dynamic optimization shifts the compressor power consumption to periods with lower electricity prices. This control approach results in lower production of N_2 and O_2 during peaks in the electricity price and high production rate during excess of electricity. This work represents a first step towards accurate modelling and robust control of a cryogenic air separation plant. In the future, we aim to improve the model and to include the additional units of a CAS plant. We believe that real time implementation of a smart control strategy will lead to significant operational cost savings.

6. Acknowledgement

We acknowledge Innovation Fund Denmark in the project "CITIES: Centre for IT-Intelligent Energy Systems in Cities" (1305-00027B) and in the project "OPTION: Optimizing Oil Production by Novel Technology Integration" (63-2013-3) for the funding of this work.

References

- Cao, Y., Swartz, C. L., Baldea, M., Blouin, S., 2015. Optimization-based assessment of design limitations to air separation plant agility in demand response scenarios. *Journal of Process Control* 33, 37–48.
- Chen, Z., Henson, M. A., Belanger, P., Megan, L., jul 2010. Nonlinear Model Predictive Control of High Purity Distillation Columns for Cryogenic Air Separation. *IEEE Transactions on Control Systems Technology* 18 (4), 811–821.
- Huang, R., Biegler, L. T., 2012. Economic NMPC for energy intensive applications with electricity price prediction. *Computer Aided Chemical Engineering* 31, 1612 – 1616.
- Kröner, A., Kronseder, T., Engl, G., Stryk, O., 2001. Dynamic optimization for air separation plants. Vol. 9 of *Computer Aided Chemical Engineering*. Elsevier, pp. 433 – 438.
- Meziou, A., 2000. The application of multivariable constrained control to cryogenic air separation units. *ISA Expo/2000*.
- Miller, J., Luyben, W. L., Blouin, S., feb 2008. Economic Incentive for Intermittent Operation of Air Separation Plants with Variable Power Costs. *Industrial & Engineering Chemistry Research* 47 (4), 1132–1139.
- Ritschel, T. K. S., 2017. Dynamic Optimization of UV Flash Processes. *Proceedings of FOCAPO / CPC 2017*.
- Roffel, B., Betlem, B., de Ruijter, J., 2000. First principles dynamic modeling and multivariable control of a cryogenic distillation process. *Computers & Chemical Engineering* 24 (1), 111–123.
- Schenk, M., Sakizlis, V., Perkins, J., Pistikopoulos, E., 2002. Optimization-Based Methodologies for Integrating Design and Control in Cryogenic Plants. In: *Computer Aided Chemical Engineering*. Vol. 10. pp. 331–336.
- Sirdeshpande, A. R., Ierapetritou, M. G., Andreovich, M. J., Naumovitz, J. P., apr 2005. Process synthesis optimization and flexibility evaluation of air separation cycles. *AIChE Journal* 51 (4), 1190–1200.
- Smith, A., Klosek, J., 2001. A review of air separation technologies and their integration with energy conversion processes. *Fuel Processing Technology* 70 (2), 115 – 134.
- Smith, J. M. J. M., Van Ness, H. C. H. C., 1987. *Introduction to chemical engineering thermodynamics*. McGraw-Hill.
- White, V., Perkins, J., Espie, D., apr 1996. Switchability analysis. *Computers & Chemical Engineering* 20 (4), 469–474.

Paper V

An Algorithm for Gradient-based Dynamic Optimization of UV Flash Processes

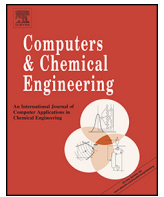
Authors:

Tobias K. S. Ritschel, Andrea Capolei, Jozsef Gaspar, and John Bagterp Jørgensen

Published in:

Computers and Chemical Engineering 114, 281–295, 2018.

DOI: <https://doi.org/10.1016/j.compchemeng.2017.10.007>.



An algorithm for gradient-based dynamic optimization of UV flash processes

Tobias K.S. Ritschel, Andrea Capolei, Jozsef Gaspar, John Bagterp Jørgensen*

Department of Applied Mathematics and Computer Science & Center for Energy Resources Engineering (CERE), Technical University of Denmark, DK-2800 Kgs. Lyngby, Denmark

ARTICLE INFO

Article history:

Received 31 August 2017

Accepted 10 October 2017

Available online 19 October 2017

Keywords:

Dynamic optimization

Optimal control

Adjoint algorithm

Single-shooting

UV flash

Vapor–liquid equilibrium

ABSTRACT

This paper presents a novel single-shooting algorithm for gradient-based solution of optimal control problems with vapor–liquid equilibrium constraints. Such optimal control problems are important in several engineering applications, for instance in control of distillation columns, in certain two-phase flow problems, and in operation of oil reservoirs. The single-shooting algorithm uses an adjoint method for the computation of gradients. Furthermore, the algorithm uses either a simultaneous or a nested approach for the numerical solution of the dynamic vapor–liquid equilibrium model equations. Two numerical examples illustrate that the simultaneous approach is faster than the nested approach and that the efficiency of the underlying thermodynamic computations is important for the overall performance of the single-shooting algorithm. We compare the performance of different optimization software as well as the performance of different compilers in a Linux operating system. These tests indicate that real-time nonlinear model predictive control of UV flash processes is computationally feasible.

© 2017 Elsevier Ltd. All rights reserved.

1. Introduction

Dynamic optimization, also called optimal control, is concerned with computing an open-loop control strategy that manipulates a dynamical system in such a way that it optimizes some performance measure, e.g. expected profit or deviation of a product quality from a target (Binder et al., 2001; Betts, 2001; Zavala and Biegler, 2009; Diehl et al., 2009; Bryson, 1999). Systems modeled by mass and energy balances and subject to vapor–liquid equilibrium constraints occur in a number of important process engineering applications, e.g. distillation (Bisgaard et al., 2017, 2015; Diehl et al., 2002; Biegler, 2010; Luyben, 1992; Stichlmair and Fair, 1998), cryogenic distillation (Laiglecia et al., 2012), and two-phase flow in pipelines (Hammer and Morin, 2014; Qiu et al., 2014). Vapor–liquid equilibrium constrained mass and energy conservation models also occur for a number of subsurface flow processes related to CO₂ sequestration (Stauffer et al., 2009), magmatic hydrothermal flow (Ingebritsen et al., 2010), and production of oil from an oil reservoir (Li and Johns, 2006; Lucia et al., 2012; Zaydullin et al., 2014). The UV flash is a single stage vapor–liquid equilibrium process and is thus

a key component in rigorous modeling of fluid vessels and flash drums (Castier, 2010; Lima et al., 2008; Arendsen and Versteeg, 2009), distillation columns (Flatby et al., 1994), two-phase computational fluid dynamical problems (Qiu et al., 2014), and thermal and compositional oil reservoir flow (Zaydullin et al., 2014). While advanced algorithms for robust simulation of the UV flash problem exist (Saha and Carroll, 1997; Castier, 2009), no algorithm for dynamic optimization of UV flash processes seems to exist in the open literature. Dynamic optimization of UV flash processes was first explained by Ritschel et al. (2017a,b). The UV flash problem is also known as the isoenergetic-isochoric flash problem or the UVn flash problem. UVn refers to specification of the internal energy, U , the total volume, V , and the total material amount (moles), n . The second law of thermodynamics, i.e. the entropy of a closed system is maximal, is used to determine the equilibrium composition with U , V , and n specified (Michelsen, 1999). The UV flash problem is different from the more common PT flash problem that occurs in steady-state optimization problems. However, it can be demonstrated that the PT flash problem with additional constraints on the internal energy, U , and the volume, V , is equivalent to the UV flash problem. Algorithmic oriented approaches to dynamic optimization of vapor–liquid equilibrium processes use a nested method in which PT flash problems are solved in the inner loop, and outer loops converge the internal energy, U , and volume, V , to their specified values.

* Corresponding author.

E-mail addresses: tobk@dtu.dk (T.K.S. Ritschel), acap@dtu.dk (A. Capolei), joca@dtu.dk (J. Gaspar), jbjo@dtu.dk (J.B. Jørgensen).

In this paper, we develop a single-shooting algorithm for solution of dynamic optimization problems with vapor–liquid equilibrium constraints, i.e. an algorithm for dynamic optimization of the UV flash problem. The key novelties of our algorithm are that an adjoint method is used for the computation of gradients (Jørgensen, 2007) and that the problem is formulated as a bilevel optimization problem. The numerical integration of the semi-explicit index-1 differential algebraic (DAE) system is the key computational operation in the single-shooting method. The DAE systems can be solved numerically with either a simultaneous approach or with a nested approach. The discretized differential equations and the equilibrium conditions are solved simultaneously in the simultaneous approach. In the nested approach, the equilibrium conditions are solved in an inner loop for each evaluation of the discretized differential equations. The key potential advantage of a nested approach is that well established algorithms for the PT-flash may be used in the inner loop. We report numerical results as well as the computational performance for implementations in C and Matlab using different optimization software, different linear algebra software, and different compilers. The computations confirm previous results in which the simultaneous approach is faster than a nested approach (Wilhelmsen et al., 2013). Hence, the performance of the single-shooting algorithm depends on the efficiency of the thermodynamic function evaluations and the numerical linear algebra, but also on whether the equilibrium equations are solved simultaneously with the differential equations or in a loop nested to the differential equations.

Previously, Kourounis et al. (2014) developed an adjoint method for gradient-based optimization of compositional reservoir flow. Their model included isothermal and isobaric (constant temperature and pressure) vapor–liquid equilibrium processes, but not the isoenergetic–isochoric situation considered in this paper. Alternatives to the single-shooting method exist. They are the multiple-shooting method (Bock and Plitt, 1984; Capolei and Jørgensen, 2012) and the simultaneous method (Biegler, 2007). Both methods have been applied to optimal control problems with vapor–liquid equilibrium constraints (Schäfer et al., 2007; Raghunathan et al., 2004). The key insight in our formulation of the vapor–liquid equilibrium is that it may be formulated as an equality constrained optimization problem (Michelsen, 1999), which in the UV-flash case is a natural and intuitive representation of the second law of thermodynamics; i.e. the entropy is maximal of a closed system with fixed energy, volume and mass. Accordingly, optimal control problems with vapor–liquid equilibrium constraints belong to a class of bilevel optimization problems (Colson et al., 2007) and also to the closely related class of mathematical programs with equilibrium constraints (Luo et al., 1996; Outrata et al., 2013). This structure is exploited in the efficient computation of the resulting index-1 differential-algebraic system and its adjoints. However, it should be noted that we only consider the situation with both phases (vapor and liquid) present, but not the more complicated situation in which phases can appear and disappear (Biegler, 2010; Sahlodin et al., 2016; Watson et al., 2017).

This paper is organized as follows. Section 2 presents the optimal control problem in consideration. Section 3 presents the single-shooting algorithm based on both the simultaneous approach and the nested approach. Section 4 describes a dynamical UV flash model, and Section 5 demonstrates the equivalence between the UV flash problem and a PT flash problem with additional constraints on the internal energy and the volume. Section 6 discusses the implementation of the single-shooting algorithm. Section 7 presents numerical solutions to a tracking-type control problem and an economical control problem. Section 8 presents a numerical performance study. Conclusions are given in Section 9.

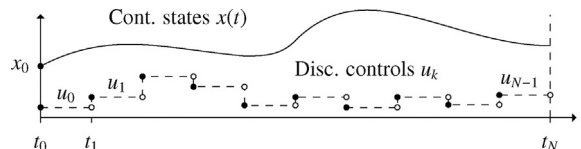


Fig. 1. Sketch of the principle in the single-shooting method. The controls, $u(t)$, are discretized in time and the continuous states, $x(t)$, are considered functions of the controls. The objective function is evaluated by solving the dynamic equations for a given set of controls.

2. Optimal control problem

We consider the following optimal control problem (OCP)

$$\min_{\{x(t); y(t); z(t)\}_{t_0}^{t_f}, \{u_k\}_{k \in \mathcal{N}}} \phi = \phi([y(t); u(t); d(t)]_{t_0}^{t_f}) \quad (1a)$$

subject to

$$x(t_0) = \hat{x}_0, \quad (1b)$$

$$G(x(t), y(t), z(t)) = 0, \quad t \in \mathcal{T}, \quad (1c)$$

$$\dot{x}(t) = F(y(t), u(t), d(t)), \quad t \in \mathcal{T}, \quad (1d)$$

$$u(t) = u_k, \quad t \in [t_k, t_{k+1}[, \quad k \in \mathcal{N}, \quad (1e)$$

$$d(t) = \hat{d}_k, \quad t \in [t_k, t_{k+1}[, \quad k \in \mathcal{N}, \quad (1f)$$

$$\{u_k\}_{k \in \mathcal{N}} \in \mathcal{U}, \quad (1g)$$

where the objective function, ϕ , is in Lagrange form

$$\phi = \int_{t_0}^{t_f} \Phi(y(t), u(t), d(t)) dt. \quad (2)$$

$x(t)$ is the state vector, $y(t)$ is a vector of algebraic variables, and $z(t)$ is a vector of adjoint algebraic variables. The estimated initial state, \hat{x}_0 , and the predicted disturbances, $\{\hat{d}_k\}_{k \in \mathcal{N}}$, are parameters in the optimization problem. $[x(t); y(t); z(t)]_{t_0}^{t_f}$ is a vector of dependent decision variables, and $\{u_k\}_{k \in \mathcal{N}}$ are independent decision variables. The time horizon is $\mathcal{T} = [t_0, t_f]$, and the indices of the control intervals are $\mathcal{N} = \{0, 1, \dots, N-1\}$.

The OCP (1) includes algebraic constraints (1c) and differential equations (1d). The equilibrium conditions for an equilibrium process can be formulated as the Karush–Kuhn–Tucker (KKT) conditions of an optimization problem. The algebraic constraints (1c) are formulated such that they can represent such KKT conditions. The differential equations (1d) are obtained from conservation principles, and the states, $x(t)$, represent the conserved quantities. The right-hand side in (1d) depends on the algebraic variables, $y(t)$, which are implicit functions of the states through the algebraic constraints (1c), i.e. $y(t) = y(x(t))$. We assume that it is possible to solve $G(x(t), y(t), z(t)) = 0$ for $y(t) = y(x(t))$ and $z(t) = z(x(t))$ when $x(t)$ is given. This is true for the vapor–liquid equilibrium processes considered in this work. We define the objective function, ψ , as

$$\psi = \psi(\{u_k\}_{k \in \mathcal{N}}; \hat{x}_0, \{\hat{d}_k\}_{k \in \mathcal{N}}) = \{\phi : (1b) - (1f)\}. \quad (3)$$

Given $\{u_k\}_{k \in \mathcal{N}}$, \hat{x}_0 , and $\{\hat{d}_k\}_{k \in \mathcal{N}}$ this defines ψ as the objective function, ϕ , obtained from (2) using the solution of (1c) and (1d) with $x(t_0) = \hat{x}_0$, $u(t) = u_k$ for $t \in [t_k, t_{k+1}[$ and $k \in \mathcal{N}$, and $d(t) = \hat{d}_k$ for $t \in [t_k, t_{k+1}[$ and $k \in \mathcal{N}$, i.e. (1b) and (1e)–(1f). Fig. 1 illustrates the discretization of the inputs and the numerical computation of the continuous states. This is the principle that is used to compute ψ in the single-shooting algorithm. With ψ defined by (3), the OCP (1) with the objective function (2) can be expressed as the finite-dimensional constrained optimization problem

$$\min_{\{u_k\}_{k \in \mathcal{N}}} \psi = \psi(\{u_k\}_{k \in \mathcal{N}}; \hat{x}_0, \{\hat{d}_k\}_{k \in \mathcal{N}}) \quad (4a)$$

$$\text{s.t. } \{u_k\}_{k \in \mathcal{N}} \in \mathcal{U}. \quad (4b)$$

The set \mathcal{U} is often a polyhedron such that the constraints (4b) can be expressed as $u_{\min} \leq u \leq u_{\max}$ and $b_l \leq A u \leq b_u$ where $u = [u_0; u_1; \dots; u_{N-1}]$. Gradient-based optimization algorithms for solving the nonlinear program (4), and thus the optimal control problem (1), require evaluation of the objective function, ψ , and the gradients, $\{\nabla_{u_k} \psi\}_{k \in \mathcal{N}}$. This involves the numerical solution of the differential-algebraic equations (1c) and (1d) and the computation of the integral (2).

2.1. Equilibrium constraints

The equilibrium processes that we consider can be described as the solution to the following parametric optimization problem

$$\min_y f(y) \quad (5a)$$

$$\text{s.t. } g(y) = x, \quad (5b)$$

$$h(y) = 0. \quad (5c)$$

The Lagrange function associated with the equilibrium optimization problem (5) is

$$\mathcal{L}(y, \eta, \mu; x) = f(y) - \eta^T(g(y) - x) - \mu^T h(y), \quad (6)$$

where η and μ are Lagrange multipliers associated with (5b) and (5c), respectively. The KKT conditions (first order optimality conditions) for a minimizer ($y = y(x)$, $\eta = \eta(x)$, $\mu = \mu(x)$) are

$$\nabla_y \mathcal{L}(y, \eta, \mu; x) = \nabla f(y) - \nabla g(y)\eta - \nabla h(y)\mu = 0, \quad (7a)$$

$$\nabla_\eta \mathcal{L}(y, \eta, \mu; x) = -(g(y) - x) = 0, \quad (7b)$$

$$\nabla_\mu \mathcal{L}(y, \eta, \mu; x) = -h(y) = 0. \quad (7c)$$

By introducing the vector $z = [\eta; \mu]$, we can rewrite the system (7) as

$$G(x, y, z) = 0, \quad (8)$$

which is equivalent to the algebraic constraints (1c).

3. Numerical solution

This section describes the numerical algorithms for the computation of the objective function, ψ , defined in (3) as well as the gradients with respect to the controls, $\{\nabla_{u_k} \psi\}_{k \in \mathcal{N}}$. The evaluation of ψ requires the solution of the semi-explicit differential-algebraic initial value problem

$$x(t_0) = \hat{x}_0, \quad (9a)$$

$$G(x(t), y(t), z(t)) = 0, \quad t \in \mathcal{T}, \quad (9b)$$

$$\dot{x}(t) = F(y(t), u(t), d(t)), \quad t \in \mathcal{T}. \quad (9c)$$

When $[y(t); u(t); d(t)]_0^{t_f}$ is given, $\psi = \phi$ is computed by quadrature. An implicit method must be used for efficient numerical solution of the system (9) because it is stiff. There exists several implicit methods such as ESDIRK methods (Kristensen et al., 2004; Völcker et al., 2010) and BDF based methods (Tolsma and Barton, 2000; Barton and Lee, 2002). In this work, we use Euler's implicit method. Furthermore, we describe the computation of the gradients, $\{\nabla_{u_k} \psi\}_{k \in \mathcal{N}}$, by an adjoint method (Jørgensen, 2007; Völcker et al., 2011; Capolei et al., 2012; Capolei and Jørgensen, 2012). These gradients (or sensitivities) may also be computed by a forward method (Kristensen et al., 2004, 2005).

As described in Section 2, the time horizon, $[t_0, t_f]$, is divided into N control intervals. There can be several time steps in each control interval, but for ease of notation, we assume that there is only one time step for each control interval. We consider both a

simultaneous approach and a nested approach for the numerical solution of the differential-algebraic system (9). In the simultaneous approach the discretized differential equations, $D_{k+1} = 0$, and the algebraic equations, $G(x_{k+1}, y_{k+1}, z_{k+1}) = 0$, are solved simultaneously for the state variables, x_{k+1} , the algebraic variables, y_{k+1} , and the adjoint algebraic variables, z_{k+1} . In the nested approach, the discretized differential equations, $D_{k+1} = 0$, are solved by iterating on the state variables, x_{k+1} , in an outer loop. For each iterate of x_{k+1} the algebraic variables, $y_{k+1} = y(x_{k+1})$, and the adjoint algebraic variables, $z_{k+1} = z(x_{k+1})$, are computed by solving the algebraic equations, $G(x_{k+1}, y_{k+1}, z_{k+1}) = 0$, in an inner loop. The two approaches are briefly described in the following subsections.

3.1. The simultaneous approach

Define $w = [x; y; z]$ and the residual function

$$\begin{aligned} R_{k+1} &= R_{k+1}(w_{k+1}) = R_{k+1}(w_{k+1}; x_k, u_k, \hat{d}_k) \\ &= R_{k+1}(x_{k+1}, y_{k+1}, z_{k+1}; x_k, u_k, \hat{d}_k) \\ &= \begin{bmatrix} D_{k+1}(x_{k+1}, x_k, y_{k+1}, u_k, \hat{d}_k) \\ G(x_{k+1}, y_{k+1}, z_{k+1}) \end{bmatrix}, \quad k \in \mathcal{N}, \end{aligned} \quad (10)$$

where the function $D_{k+1} = D_{k+1}(x_{k+1}, x_k, y_{k+1}, u_k, \hat{d}_k)$ is

$$D_{k+1} = x_{k+1} - x_k - \Delta t_k F(y_{k+1}, u_k, \hat{d}_k). \quad (11)$$

Euler's implicit method for the system (9) corresponds to solving the residual equations

$$R_{k+1} = R_{k+1}(w_{k+1}) = 0, \quad k \in \mathcal{N}, \quad (12)$$

for $\{w_{k+1}\}_{k \in \mathcal{N}}$ by sequentially marching forward when $x_0 = \hat{x}_0$, $\{u_k\}_{k \in \mathcal{N}}$, and $\{\hat{d}_k\}_{k \in \mathcal{N}}$ are given. We solve the residual equations (12) with an inexact Newton method, i.e. by solving the following sequence of linear systems

$$w_{k+1}^{m+1} = w_{k+1}^m - (M_R^m)^{-1} R_{k+1}(w_{k+1}^m). \quad (13)$$

The initial guess for the Newton iterations, w_{k+1}^0 , is the states, the algebraic variables, and the adjoint algebraic variables in the previous time step, i.e. $w_k = [x_k; y_k; z_k]$. The initial iteration matrix, M_R^0 , is the Jacobian of the residual function evaluated at the initial guess

$$M_R^0 = \frac{\partial R_{k+1}}{\partial w_{k+1}}(w_{k+1}^0). \quad (14)$$

We update the iteration matrix:

$$M_R^{m+1} = \frac{\partial R_{k+1}}{\partial w_{k+1}}(w_{k+1}^{m+1}), \quad (15)$$

if the estimate, w_{k+1}^{m+1} , does not sufficiently reduce the norm of the residual function as compared to the previous estimate, w_{k+1}^m :

$$\|R_{k+1}(w_{k+1}^{m+1})\| > \tau_R \|R_{k+1}(w_{k+1}^m)\|, \quad (16)$$

where $\tau_R \in [0, 1]$. The iteration matrix is not updated (and $M_R^{m+1} = M_R^m$) if (16) is not satisfied. The Newton iterations (13) are terminated when the norm of the residual function is smaller than the specified tolerance, ϵ_R , i.e.

$$\|R_{k+1}(w_{k+1}^{m+1})\| < \epsilon_R. \quad (17)$$

The Jacobian of the residual function is

$$\frac{\partial R_{k+1}}{\partial w_{k+1}} = \begin{bmatrix} I & -\Delta t_k \frac{\partial F}{\partial y} & 0 \\ \frac{\partial G}{\partial x} & \frac{\partial G}{\partial y} & \frac{\partial G}{\partial z} \end{bmatrix}, \quad (18)$$

where the Jacobians of the algebraic function, G , are

$$\frac{\partial G}{\partial x} = \frac{\partial G}{\partial x}(w_{k+1}) = [0; \quad 1; \quad 0], \tag{19a}$$

$$\left[\frac{\partial G}{\partial y} \quad \frac{\partial G}{\partial z} \right] = \left[\frac{\partial G}{\partial y}(w_{k+1}) \quad \frac{\partial G}{\partial z}(w_{k+1}) \right] = K(w_{k+1}). \tag{19b}$$

The matrix K is the KKT matrix for the equilibrium optimization problem (5). With the adjoint algebraic variables defined as $z_{k+1} = [\eta_{k+1}; \mu_{k+1}]$, the KKT matrix may be expressed as $K(w_{k+1}) = K(x_{k+1}, y_{k+1}, z_{k+1}) = K(y_{k+1}, \eta_{k+1}, \mu_{k+1}; x_{k+1})$, where

$$K(y, \eta, \mu; x) = \begin{bmatrix} \nabla_{yy}^2 \mathcal{L}(y, \eta, \mu; x) & -\nabla g(y) & -\nabla h(y) \\ -\nabla g(y)^T & 0 & 0 \\ -\nabla h(y)^T & 0 & 0 \end{bmatrix}. \tag{20}$$

The Hessian of the Lagrangian with respect to the algebraic variables, y , is

$$\begin{aligned} \nabla_{yy}^2 \mathcal{L} &= \nabla_{yy}^2 \mathcal{L}(y, \eta, \mu; x) \\ &= \nabla^2 f(y) - \sum_i \eta_i \nabla^2 g_i(y) - \sum_i \mu_i \nabla^2 h_i(y). \end{aligned} \tag{21}$$

The two sums in (21) span over each component of the functions g and h .

3.1.1. The adjoint method for the simultaneous approach

We substitute the residual equations (12) into the objective function, ψ , in (3):

$$\psi = \psi(\{u_k\}_{k \in \mathcal{N}}; \hat{x}_0, \{\hat{d}_k\}_{k \in \mathcal{N}}) \tag{22a}$$

$$= \left\{ \phi = \sum_{k \in \mathcal{N}} \Phi_k(y_{k+1}, u_k, \hat{d}_k) : \right. \tag{22b}$$

$$x_0 = \hat{x}_0, \tag{22c}$$

$$R_{k+1}(w_{k+1}; x_k, u_k, \hat{d}_k) = 0, \quad k \in \mathcal{N}, \tag{22d}$$

$$[x_{k+1}; y_{k+1}; z_{k+1}] = w_{k+1}, \quad k \in \mathcal{N} \}. \tag{22e}$$

The sum in (22b) approximates the integral in (2). Φ_k approximates the integral of Φ over $[t_k, t_{k+1}]$ using the rectangle rule with y_{k+1} (instead of y_k):

$$\Phi_k = \Phi_k(y_{k+1}, u_k, \hat{d}_k) = \Delta t_k \Phi(y_{k+1}, u_k, \hat{d}_k). \tag{23}$$

The integral may be approximated more accurately with other quadrature methods. However, the rectangle rule is used because of its simplicity and because it is consistent with the implicit Euler method used for integration of the differential equations. The gradients of ψ with respect to the inputs, $\{\nabla_{u_k} \psi\}_{k \in \mathcal{N}}$ are computed by solving the following equation for the adjoints, $\lambda_{\mathcal{N}}$,

$$\left(\frac{\partial R_{\mathcal{N}}}{\partial w_{\mathcal{N}}} \right)^T \lambda_{\mathcal{N}} = -\nabla_{w_{\mathcal{N}}} \Phi_{\mathcal{N}-1}, \tag{24}$$

and inserting into the following expression for the gradients

$$\nabla_{u_k} \psi = \nabla_{u_k} \Phi_k + \left(\frac{\partial R_{k+1}}{\partial u_k} \right)^T \lambda_{k+1}, \quad k \in \mathcal{N}. \tag{25}$$

Each subsequent set of adjoints, λ_k , is computed by marching backwards in the equations

$$\left(\frac{\partial R_k}{\partial w_k} \right)^T \lambda_k = - \left(\frac{\partial R_{k+1}}{\partial w_k} \right)^T \lambda_{k+1} - \nabla_{w_k} \Phi_{k-1}, \quad k = \mathcal{N} - 1, \dots, 1. \tag{26}$$

The Jacobian of the residual function, $\frac{\partial R_k}{\partial w_k}$, is given in (18) and the Jacobian of the residual function with respect to the state variables, the algebraic variables, and the adjoint algebraic variables in the previous time step is

$$\frac{\partial R_{k+1}}{\partial w_k}(w_{k+1}; x_k, u_k, \hat{d}_k) = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \tag{27}$$

for $k = 1, \dots, \mathcal{N} - 1$. The gradient of Φ_k is

$$\nabla_{w_{k+1}} \Phi_k = [0; \Delta t_k \nabla_y \Phi(y_{k+1}, u_k, \hat{d}_k); 0], \quad k \in \mathcal{N}. \tag{28}$$

3.2. The nested approach

We solve the algebraic equations (9b) for the algebraic variables, y_{k+1}^m , and the adjoint algebraic variables, z_{k+1}^m , for each estimate of the states, x_{k+1}^m . The algebraic equations are solved with an inexact Newton method:

$$[y_{k+1}^{l+1}; z_{k+1}^{l+1}] = [y_{k+1}^l; z_{k+1}^l] - (M_G^l)^{-1} G(x_{k+1}^m, y_{k+1}^l, z_{k+1}^l). \tag{29}$$

If $m=0$, the initial guess for the Newton iterations, $[y_{k+1}^0; z_{k+1}^0]$, is the algebraic variables and the adjoint algebraic variables in the previous time step, $[y_k; z_k]$. Otherwise, the initial guess is the variables from the previous outer Newton iteration, $[y_{k+1}^{m-1}; z_{k+1}^{m-1}]$. The initial iteration matrix, M_G^0 , is the KKT matrix (20) evaluated at the initial guess

$$M_G^0 = K(x_{k+1}^m, y_{k+1}^0, z_{k+1}^0). \tag{30}$$

The iteration matrix,

$$M_G^{l+1} = K(x_{k+1}^m, y_{k+1}^{l+1}, z_{k+1}^{l+1}), \tag{31}$$

is updated if the estimates, y_{k+1}^{l+1} and z_{k+1}^{l+1} , do not sufficiently reduce the norm of the algebraic function, G , as compared to the previous estimates, y_{k+1}^l and z_{k+1}^l . This condition can be expressed as

$$\|G(x_{k+1}^m, y_{k+1}^{l+1}, z_{k+1}^{l+1})\| > \tau_G \|G(x_{k+1}^m, y_{k+1}^l, z_{k+1}^l)\|, \tag{32}$$

where $\tau_G \in [0, 1]$. The iteration matrix is not updated (and $M_G^{l+1} = M_G^l$) if (32) is not satisfied. The Newton iterations (29) are terminated when the norm of the algebraic function, G , is smaller than a specified tolerance, ϵ_G :

$$\|G(x_{k+1}^m, y_{k+1}^{l+1}, z_{k+1}^{l+1})\| < \epsilon_G. \tag{33}$$

y_{k+1}^m and z_{k+1}^m denote the estimates that satisfy the stopping criteria (33). The outer Newton iterations are terminated if x_{k+1}^m and y_{k+1}^m furthermore satisfies the discretized differential equations within a tolerance of ϵ_D :

$$\|D_{k+1}(x_{k+1}^m, x_k, y_{k+1}^m, u_k, \hat{d}_k)\| < \epsilon_D. \tag{34}$$

If the stopping criteria (34) is not satisfied, the next estimate of the states, x_{k+1}^{m+1} , is computed:

$$x_{k+1}^{m+1} = x_{k+1}^m - (M_D^m)^{-1} D_{k+1}(x_{k+1}^m, x_k, y_{k+1}^m, u_k, \hat{d}_k). \tag{35}$$

The initial guess for the outer Newton iterations, x_{k+1}^0 , is the states in the previous time step, x_k . The initial iteration matrix, M_D^0 , is the Jacobian of D_{k+1} evaluated at the initial guess

$$M_D^0 = \frac{\partial D_{k+1}}{\partial x_{k+1}}(x_{k+1}^0, x_k, y_{k+1}^0, u_k, \hat{d}_k). \tag{36}$$

y_{k+1}^0 is the solution to the algebraic equations, with x_{k+1}^0 as parameter, which is found with the inner Newton iterations (29). Once the outer Newton step (35) has been computed, the inner Newton iterations (29) are repeated in order to update the estimate for the

algebraic variables, y_{k+1}^{m+1} , and the adjoint algebraic variables, z_{k+1}^{m+1} . The iteration matrix,

$$M_D^{m+1} = \frac{\partial D_{k+1}}{\partial x_{k+1}}(x_{k+1}^{m+1}, x_k, y_{k+1}^{m+1}, u_k, \hat{d}_k), \quad (37)$$

is updated if the estimates, x_{k+1}^{m+1} and y_{k+1}^{m+1} , do not sufficiently reduce the norm of the function D_{k+1} as compared to the previous estimates, x_{k+1}^m and y_{k+1}^m . This is expressed as the condition

$$\|D_{k+1}(x_{k+1}^{m+1}, x_k, y_{k+1}^{m+1}, u_k, \hat{d}_k)\| > \tau_D \|D_{k+1}(x_{k+1}^m, x_k, y_{k+1}^m, u_k, \hat{d}_k)\|, \quad (38)$$

where $\tau_D \in [0, 1]$. The iteration matrix is not updated (and $M_D^{m+1} = M_D^m$) if (38) is not satisfied. The inner and outer Newton iterations are terminated when an estimate has been reached that satisfies both the discretized differential equations and the algebraic equations, i.e. that satisfies both (33) and (34). The Jacobian of the function D_{k+1} is

$$\frac{\partial D_{k+1}}{\partial x_{k+1}} = I - \Delta t_k \frac{\partial F}{\partial y}(y_{k+1}, u_k, \hat{d}_k) \frac{\partial y_{k+1}}{\partial x_{k+1}}. \quad (39)$$

The sensitivities of the algebraic variables and the adjoint algebraic variables are

$$\begin{bmatrix} \frac{\partial y_{k+1}}{\partial x_{k+1}}; \frac{\partial z_{k+1}}{\partial x_{k+1}} \end{bmatrix} = -K(w_{k+1})^{-1} \frac{\partial G}{\partial x}(w_{k+1}). \quad (40)$$

3.2.1. The adjoint method for the nested approach

The discretized differential equations, $D_{k+1} = 0$, are substituted into the objective function, ψ , in (3):

$$\psi = \psi(\{u_k\}_{k \in \mathcal{N}}, \hat{x}_0, \{\hat{d}_k\}_{k \in \mathcal{N}}) \quad (41a)$$

$$= \begin{cases} \phi = \sum_{k \in \mathcal{N}} \Phi_k(y_{k+1}, u_k, \hat{d}_k) : \end{cases} \quad (41b)$$

$$x_0 = \hat{x}_0, \quad (41c)$$

$$D_{k+1}(x_{k+1}, x_k, y_{k+1}, u_k, \hat{d}_k) = 0, \quad k \in \mathcal{N}, \quad (41d)$$

$$G(x_{k+1}, y_{k+1}, z_{k+1}) = 0, \quad k \in \mathcal{N}, \quad (41e)$$

where Φ_k is defined in (23). The gradients of ψ in (41) with respect to the inputs, $\{\nabla_{u_k} \psi\}_{k \in \mathcal{N}}$ are computed by solving for the adjoints, $\lambda_{\mathcal{N}}$,

$$\left(\frac{\partial D_{\mathcal{N}}}{\partial x_{\mathcal{N}}} \right)^T \lambda_{\mathcal{N}} = - \left(\frac{\partial y_{\mathcal{N}}}{\partial x_{\mathcal{N}}} \right)^T \nabla_{y_{\mathcal{N}}} \Phi_{N-1}. \quad (42)$$

and inserting into the following expression for the gradients

$$\nabla_{u_k} \psi = \nabla_{u_k} \Phi_k + \left(\frac{\partial D_{k+1}}{\partial u_k} \right)^T \lambda_{k+1}, \quad k \in \mathcal{N}. \quad (43)$$

Each subsequent set of adjoints, λ_k , satisfies

$$\left(\frac{\partial D_k}{\partial x_k} \right)^T \lambda_k = - \left(\frac{\partial D_{k+1}}{\partial x_k} \right)^T \lambda_{k+1} - \left(\frac{\partial y_k}{\partial x_k} \right)^T \nabla_{y_k} \Phi_{k-1} \quad (44)$$

for $k = N-1, \dots, 1$. The Jacobian of the function D_k , $\frac{\partial D_k}{\partial x_k}$, is defined in (39), and the Jacobian with respect to the states in the previous time step is

$$\frac{\partial D_{k+1}}{\partial x_k}(x_{k+1}, x_k, y_{k+1}, u_k, \hat{d}_k) = -I, \quad k = 1, \dots, N-1. \quad (45)$$

This concludes the description of the numerical methods for the computation of the objective function, ψ , defined in (3) as well as the gradients with respect to the inputs, $\{\nabla_{u_k} \psi\}_{k \in \mathcal{N}}$

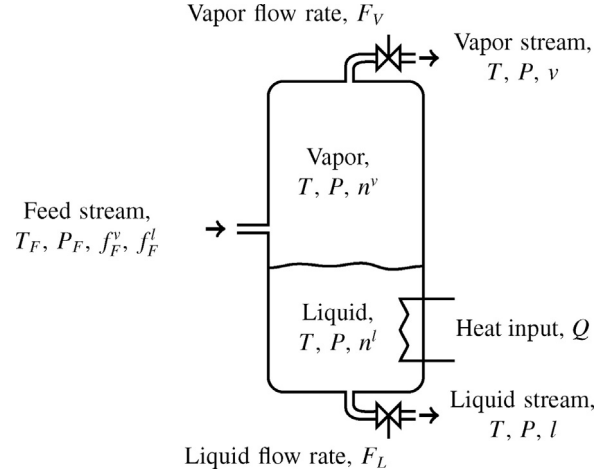


Fig. 2. Sketch of a flash unit. The unit is supplied by a vapor–liquid feed stream. A vapor stream and a liquid stream extract mass from the vapor phase and the liquid phase of the mixture in the unit. The mixture in the flash unit is furthermore subject to external heating or cooling.

4. The dynamic UV flash model

This section describes the dynamic UV flash model, i.e. the differential-algebraic constraints (1c) and (1d) in the optimal control problem (1). Fig. 2 illustrates the flash unit. The mixture in the unit contains N_C components. We assume that two phases (vapor and liquid) always exist and that they are in chemical, thermal and mechanical equilibrium. The flash unit is continuously supplied by a feed stream at a given temperature, pressure, and composition. The mixture in the flash unit is extracted continuously through a vapor stream and a liquid stream. The total flow rate of the vapor stream, F_V , and the total flow rate of the liquid stream, F_L , are manipulated. The mixture in the unit is subject to either heating or cooling with heat flux, Q . The heat flux is also manipulated.

The following subsections describe the properties of the mixture in the unit, the feed stream, the vapor stream, and the liquid stream. We furthermore formulate the conservation equations and the equilibrium conditions. We illustrate how all needed thermodynamic properties can be computed by evaluation of the enthalpy, $H = H(T, P, n)$, the entropy, $S = S(T, P, n)$, and the volume, $V = V(T, P, n)$, as function of temperature, T , pressure, P , and composition (mole numbers), n .

4.1. The mixture in the flash unit

The flash unit contains a mixture in thermal, mechanical, and chemical vapor–liquid equilibrium. The thermal equilibrium implies that the temperature of the vapor phase, T^v , and the liquid phase, T^l , are identical, i.e. $T = T^v = T^l$. The mechanical equilibrium implies that the pressure in the vapor phase, P^v , is identical to the pressure in the liquid phase, P^l , i.e. $P = P^v = P^l$. The vapor holdup is n^v and the liquid holdup is n^l . The total composition (in moles) of the mixture is $n_i = n_i^v + n_i^l$ for $i = 1, 2, \dots, N_C$. The properties of the mixture in the flash unit are

$$N^v = \sum_{i=1}^{N_C} n_i^v, \quad N^l = \sum_{i=1}^{N_C} n_i^l, \quad N = \sum_{i=1}^{N_C} n_i,$$

$$y_i = \frac{n_i^v}{N^v}, \quad x_i = \frac{n_i^l}{N^l}, \quad z_i = \frac{n_i}{N}.$$

The vapor fraction is $\beta = N^v/N$. The specification of (T, P, n^v) and (T, P, n^l) also allows for the computation of the enthalpy, entropy, and volume:

$$H^v = H^v(T, P, n^v), H^l = H^l(T, P, n^l), H = H^v + H^l,$$

$$S^v = S^v(T, P, n^v), S^l = S^l(T, P, n^l), S = S^v + S^l,$$

$$V^v = V^v(T, P, n^v), V^l = V^l(T, P, n^l), V = V^v + V^l.$$

Given (H, S, V) all other thermodynamic state functions may be computed: the internal energy is $U = H - PV$, Gibbs free energy is $G = H - TS$, and Helmholtz free energy is $A = U - TS$.

4.2. The feed

The feed is a vapor–liquid mixture at temperature T_F and pressure P_F . $f_{F,i}^v$ and $f_{F,i}^l$ denote the molar vapor and liquid feed flow rates of component i . The total feed flow rate of the i 'th component is $f_{F,i} = f_{F,i}^v + f_{F,i}^l$ for $i = 1, 2, \dots, N_C$, and the total flow rate is $F_F = F_F^v + F_F^l$. F_F^v and F_F^l are the total flow rates of the vapor and liquid phases of the feed. The properties of the feed stream are

$$F_F^v = \sum_{i=1}^{N_C} f_{F,i}^v, F_F^l = \sum_{i=1}^{N_C} f_{F,i}^l, F_F = \sum_{i=1}^{N_C} f_{F,i},$$

$$y_{F,i} = \frac{f_{F,i}^v}{F_F^v}, x_{F,i} = \frac{f_{F,i}^l}{F_F^l}, z_{F,i} = \frac{f_{F,i}}{F_F}.$$

The fraction of vapor in the feed is $\beta_F = F_F^v/F_F$. The specification of $(T_F, P_F, f_{F,i}^v, f_{F,i}^l)$ also allows for the computation of the enthalpy, entropy, and volume of the feed stream:

$$H_F^v = H^v(T_F, P_F, f_{F,i}^v), H_F^l = H^l(T_F, P_F, f_{F,i}^l), H_F = H_F^v + H_F^l,$$

$$S_F^v = S^v(T_F, P_F, f_{F,i}^v), S_F^l = S^l(T_F, P_F, f_{F,i}^l), S_F = S_F^v + S_F^l,$$

$$V_F^v = V^v(T_F, P_F, f_{F,i}^v), V_F^l = V^l(T_F, P_F, f_{F,i}^l), V_F = V_F^v + V_F^l.$$

The feed is completely specified by $(T_F, P_F, f_{F,i}^v, f_{F,i}^l)$ as other thermodynamic functions concerned with the feed may be computed using (H_F, S_F, V_F) .

4.3. The vapor stream

The vapor outlet stream has the same temperature, T , the same pressure, P , and the same composition, y , as the vapor phase in the unit. The total flow rate of the vapor stream is F_V . This implies that the compositional flow rates are $v_i = y_i F_V$ for $i = 1, 2, \dots, N_C$. Let h^v denote the molar enthalpy, s^v denote the molar entropy, and v^v denote the molar volume of the vapor phase in the flash unit. Then the total properties of the vapor stream are $H_V = F_V h^v$, $S_V = F_V s^v$, and $V_V = F_V v^v$.

4.4. The liquid stream

The liquid outlet stream has the same temperature, T , the same pressure, P , and the same composition, x , as the liquid phase in the flash unit. F_L denotes the total flow rate of the liquid stream, and the compositional flow rates are $l_i = x_i F_L$ for $i = 1, 2, \dots, N_C$. Let h^l denote the molar enthalpy, s^l denote the molar entropy, and v^l denote the molar volume of the liquid phase in the flash unit. The total properties of the liquid stream are then given by $H_L = F_L h^l$, $S_L = F_L s^l$, and $V_L = F_L v^l$.

4.5. The thermodynamic model and database

The total thermodynamic properties of the mixture, the feed, the vapor stream, and the liquid stream are computed from molar

thermodynamic properties. The molar thermodynamic properties (h^v, s^v, v^v) of a vapor phase mixture are the functions $h^v = h^v(T, P, y)$, $s^v = s^v(T, P, y)$, and $v^v = v^v(T, P, y)$. The corresponding total properties of the vapor phase are $H^v = N^v h^v$, $S^v = N^v s^v$, and $V^v = N^v v^v$. Similarly, the molar thermodynamic properties (h^l, s^l, v^l) of a liquid phase mixture are the functions $h^l = h^l(T, P, x)$, $s^l = s^l(T, P, x)$, and $v^l = v^l(T, P, x)$. The corresponding total properties of the liquid phase are $H^l = N^l h^l$, $S^l = N^l s^l$, and $V^l = N^l v^l$. We compute the molar vapor–liquid properties (h^v, s^v, v^v) and (h^l, s^l, v^l) with a recently developed thermodynamic library, ThermoLib (Ritschel et al., 2016, 2017c). ThermoLib uses data and correlations from the DIPPR database (Thomson, 1996) together with either the SRK EOS or the PR EOS. It is implemented in both Matlab and C, and it is distributed as open-source software at www.psetools.org. It provides routines for evaluation of enthalpy, entropy, and volume. The routines furthermore provide first and second order derivatives with respect to temperature, pressure, and composition (in moles).

4.6. The equilibrium constraint – vapor–liquid equilibrium

Vapor–liquid equilibrium processes may be formulated as optimization problems (Michelsen, 1999). When the internal energy, U , and the volume, V , are specified, the optimization problem determining the equilibrium temperature, pressure, and vapor–liquid compositions is called the UV flash problem. The UV flash optimization problem is

$$\max_{T, P, n^v, n^l} S = S^v(T, P, n^v) + S^l(T, P, n^l) \quad (48a)$$

$$\text{s.t. } U^v(T, P, n^v) + U^l(T, P, n^l) = U, \quad (48b)$$

$$V^v(T, P, n^v) + V^l(T, P, n^l) = V, \quad (48c)$$

$$n_i^v + n_i^l = n_i, \quad i = 1, \dots, N_C. \quad (48d)$$

U is the specified internal energy, and V is the total fixed volume of the flash unit. n_i is the specified total amount (in moles) of component i in the flash unit. The optimization problem (48) is a mathematical statement of the second law of thermodynamics which states that the entropy of a closed system at equilibrium is maximal. The UV flash optimization problem (48) is in the form of the optimization problem (5) where

$$f(y) = f(T, P, n^v, n^l) = - (S^v(T, P, n^v) + S^l(T, P, n^l)), \quad (49a)$$

$$g(y) = g(T, P, n^v, n^l) = \begin{bmatrix} U^v(T, P, n^v) + U^l(T, P, n^l) \\ n^v + n^l \end{bmatrix}, \quad (49b)$$

$$h(y) = h(T, P, n^v, n^l) = V^v(T, P, n^v) + V^l(T, P, n^l) - V. \quad (49c)$$

The state variables, x , the algebraic variables, y , and the adjoint algebraic variables, z , are

$$x = [U; n] \in \mathbb{R}^{1+N_C}, \quad (50a)$$

$$y = [T; P; n^v; n^l] \in \mathbb{R}^{2+2N_C}, \quad (50b)$$

$$z = [\mu; \eta] \in \mathbb{R}^{2+N_C}. \quad (50c)$$

The UV flash problem is sometimes referred to as the UVn flash problem to indicate that U, V , and n are specified. It is also known as the isoenergetic–isochoric flash (constant energy–constant volume).

4.7. The differential equations – conservation equations

The mass and energy conservation equations that describe the temporal evolution of the internal energy and the total holdup are

$$\dot{U}(t) = H_F^v(t) + H_F^l(t) - H_V(t) - H_L(t) + Q(t), \quad (51a)$$

$$\dot{n}_i(t) = f_{F,i}^v(t) + f_{F,i}^l(t) - v_i(t) - l_i(t), \quad i = 1, \dots, N_C. \quad (51b)$$

U is the internal energy, and n_i is the total holdup of component i . H_V and H_L are the enthalpies of the vapor and liquid streams, respectively. H_F^v and H_F^l are the vapor and liquid enthalpies of the feed. $f_{F,i}^v$ and $f_{F,i}^l$ are the molar vapor and liquid feed flow rates. v_i and l_i are the molar vapor and liquid stream flow rates. The conservation equations (51) are in the form of the differential equations (1d) where

$$F(y(t), u(t), d(t)) = \begin{bmatrix} H_F^v(t) + H_F^l(t) - H_V(t) - H_L(t) + Q(t) \\ f_{F,i}^v(t) + f_{F,i}^l(t) - v(t) - l(t) \end{bmatrix}. \quad (52)$$

The algebraic variables, y , are defined in (50), and the controls (manipulated variables), u , and the disturbance variables, d , are

$$u = [Q; F_V; F_L] \in \mathbb{R}^3, \quad (53a)$$

$$d = [T_F; P_F; f_{F,i}^v; f_{F,i}^l] \in \mathbb{R}^{2+2N_C}. \quad (53b)$$

5. The relation between the UV flash and the PT flash

In this section, we demonstrate that the solution of the UV flash is identical to the solution of a PT flash with constraints on the internal energy, U , and the volume, V . The solution to the PT flash problem is the vapor–liquid composition that minimizes Gibbs energy at a given temperature and pressure while satisfying a mass balance constraint. The PT flash is the most common type of flash because it can be formulated as an unconstrained minimization problem and therefore solved efficiently.

5.1. The UV flash

The UV flash maximization problem (48) is equivalent to the following minimization problem

$$\min_{T, P, n^v, n^l} -S = - (S^v(T, P, n^v) + S^l(T, P, n^l)), \quad (54a)$$

$$\text{s.t. } U^v(T, P, n^v) + U^l(T, P, n^l) = U, \quad (54b)$$

$$V^v(T, P, n^v) + V^l(T, P, n^l) = V, \quad (54c)$$

$$n_i^v + n_i^l = n_i, \quad i = 1, \dots, N_C. \quad (54d)$$

The solution to the UV flash problem is the temperature, pressure, and vapor–liquid composition of a closed system at equilibrium, (T, P, n^v, n^l) , with a given internal energy, volume, and total composition. The Lagrange function of the UV flash problem (54) is

$$\begin{aligned} \mathcal{L} = & - (S^v(T, P, n^v) + S^l(T, P, n^l)) \\ & - \kappa (U^v(T, P, n^v) + U^l(T, P, n^l) - U) \\ & - \lambda (V^v(T, P, n^v) + V^l(T, P, n^l) - V) \\ & - \sum_{i=1}^{N_C} \bar{\mu}_i (n_i^v + n_i^l - n_i). \end{aligned} \quad (55)$$

κ is the Lagrange multiplier associated with the internal energy constraint (54b), λ is the Lagrange multiplier associated with the volume constraint (54c), and $\{\bar{\mu}_i\}_{i=1}^{N_C}$ are the Lagrange multipliers associated with the mole balances (54d). In this section, we use the notation $S^{vl} = S^v + S^l$, $U^{vl} = U^v + U^l$, and $V^{vl} = V^v + V^l$ to distinguish these properties from the specified values U and V . The first

order optimality conditions for the UV flash minimization problem (54) are

$$\frac{\partial \mathcal{L}}{\partial T} = - \frac{\partial S^{vl}}{\partial T} - \kappa \frac{\partial U^{vl}}{\partial T} - \lambda \frac{\partial V^{vl}}{\partial T} = 0, \quad (56a)$$

$$\frac{\partial \mathcal{L}}{\partial P} = - \frac{\partial S^{vl}}{\partial P} - \kappa \frac{\partial U^{vl}}{\partial P} - \lambda \frac{\partial V^{vl}}{\partial P} = 0, \quad (56b)$$

$$\frac{\partial \mathcal{L}}{\partial n_i^v} = - \frac{\partial S^v}{\partial n_i^v} - \kappa \frac{\partial U^v}{\partial n_i^v} - \lambda \frac{\partial V^v}{\partial n_i^v} - \bar{\mu}_i = 0, \quad i = 1, \dots, N_C, \quad (56c)$$

$$\frac{\partial \mathcal{L}}{\partial n_i^l} = - \frac{\partial S^l}{\partial n_i^l} - \kappa \frac{\partial U^l}{\partial n_i^l} - \lambda \frac{\partial V^l}{\partial n_i^l} - \bar{\mu}_i = 0, \quad i = 1, \dots, N_C, \quad (56d)$$

$$U^{vl}(T, P, n^v, n^l) = U^v(T, P, n^v) + U^l(T, P, n^l) = U, \quad (56e)$$

$$V^{vl}(T, P, n^v, n^l) = V^v(T, P, n^v) + V^l(T, P, n^l) = V, \quad (56f)$$

$$n_i^v + n_i^l = n_i, \quad i = 1, \dots, N_C. \quad (56g)$$

(56a) and (56b) are the stationarity conditions associated with the temperature and pressure gradients. (56c) and (56d) are the stationarity conditions associated with the component derivatives, and (56e)–(56g) are the feasibility conditions. The first order optimality conditions (56) of the equality constrained optimization problem (54) are necessary conditions for a minimizer of (54).

5.2. The UV constrained PT flash

The solution to the PT flash problem, (n^v, n^l) , minimizes Gibbs free energy, G , subject to a mass balance constraint

$$\min_{n^v, n^l} G = G^v(T, P, n^v) + G^l(T, P, n^l) \quad (57a)$$

$$\text{s.t. } n_i^v + n_i^l = n_i, \quad i = 1, \dots, N_C. \quad (57b)$$

The Lagrange function of the PT flash problem (57) is

$$\mathcal{L} = (G^v(T, P, n^v) + G^l(T, P, n^l)) - \sum_{i=1}^{N_C} \mu_i (n_i^v + n_i^l - n_i), \quad (58)$$

where $\{\mu_i\}_{i=1}^{N_C}$ denote the Lagrange multipliers associated with the mole balances (57b). The first order optimality conditions for the PT flash are

$$\frac{\partial \mathcal{L}}{\partial n_i^v} = \frac{\partial G^v}{\partial n_i^v} - \mu_i = 0, \quad i = 1, \dots, N_C, \quad (59a)$$

$$\frac{\partial \mathcal{L}}{\partial n_i^l} = \frac{\partial G^l}{\partial n_i^l} - \mu_i = 0, \quad i = 1, \dots, N_C, \quad (59b)$$

$$n_i^v + n_i^l = n_i, \quad i = 1, \dots, N_C. \quad (59c)$$

(59a) and (59b) are the stationarity conditions, and (59c) are the feasibility conditions. The first order optimality conditions (59) are necessary conditions for a minimizer of the PT flash (57). The PT flash provides the equilibrium composition of the vapor phase, n^v , and the liquid phase, n^l , at given temperature, T , and pressure, P . We combine the PT flash problem (57) with the following constraints on the internal energy, U , and the volume, V ,

$$U^{vl}(T, P, n^v, n^l) = U^v(T, P, n^v) + U^l(T, P, n^l) = U, \quad (60a)$$

$$V^{vl}(T, P, n^v, n^l) = V^v(T, P, n^v) + V^l(T, P, n^l) = V. \quad (60b)$$

The constraints (60) are identical to the constraints (54b) and (54c) in the UV optimization problem. A point (T, P, n^v, n^l) satisfying the PT flash (57) and the UV constraints (60) must satisfy the conditions (59) and (60).

5.3. Equivalence of the UV flash and the UV-constrained PT flash

We show that the first order optimality conditions (56) of the UV flash (54) are equivalent to the first order optimality conditions (59) of the PT flash (57) in combination with the UV constraints (60). This shows that the solution of the UV flash problem (54) is equivalent to the solution of the PT flash problem (57) combined with the specification of internal energy, U , and volume, V , in (60). The feasibility conditions (56e)–(56g) for the UV flash (54) are equivalent to the feasibility conditions (59c) for the PT flash (57) and the UV constraints (60). The thermodynamic relations $G = H - TS$ and $H = U + PV$ imply that $G = U + PV - TS$ such that

$$\frac{\partial G^v}{\partial n_i^v} = \frac{\partial U^v}{\partial n_i^v} + P \frac{\partial V^v}{\partial n_i^v} - T \frac{\partial S^v}{\partial n_i^v}, \quad i = 1, \dots, N_C, \quad (61a)$$

$$\frac{\partial G^l}{\partial n_i^l} = \frac{\partial U^l}{\partial n_i^l} + P \frac{\partial V^l}{\partial n_i^l} - T \frac{\partial S^l}{\partial n_i^l}, \quad i = 1, \dots, N_C. \quad (61b)$$

Therefore, the stationarity conditions (59a) and (59b) of the PT flash (57) may be reformulated using (61):

$$\frac{\mu_i}{T} = -\frac{\partial S^v}{\partial n_i^v} - \left(-\frac{1}{T}\right) \frac{\partial U^v}{\partial n_i^v} - \left(-\frac{P}{T}\right) \frac{\partial V^v}{\partial n_i^v}, \quad i = 1, \dots, N_C, \quad (62a)$$

$$\frac{\mu_i}{T} = -\frac{\partial S^l}{\partial n_i^l} - \left(-\frac{1}{T}\right) \frac{\partial U^l}{\partial n_i^l} - \left(-\frac{P}{T}\right) \frac{\partial V^l}{\partial n_i^l}, \quad i = 1, \dots, N_C. \quad (62b)$$

The stationarity conditions (62) for the PT flash (57) are equivalent to the stationarity conditions (56c) and (56d) for the UV flash (54) provided that the Lagrange multipliers (κ , λ , $\{\bar{\mu}_i\}_{i=1}^{N_C}$) are

$$\kappa = \frac{-1}{T}, \quad (63a)$$

$$\lambda = \frac{-P}{T}, \quad (63b)$$

$$\bar{\mu}_i = \frac{\mu_i}{T}, \quad i = 1, \dots, N_C. \quad (63c)$$

The fundamental thermodynamic relations (Smith et al., 2005, Chap. 6)

$$\frac{\partial S}{\partial T} = \frac{C_p}{T}, \quad (64a)$$

$$\frac{\partial U}{\partial T} = C_p - P \frac{\partial V}{\partial T}, \quad (64b)$$

may be used to demonstrate that (56a) is satisfied when $\kappa = -1/T$ and $\lambda = -P/T$. Similarly, the fundamental thermodynamic relations (Smith et al., 2005, Chap. 6)

$$\frac{\partial S}{\partial P} = -\frac{\partial V}{\partial T}, \quad (65a)$$

$$\frac{\partial U}{\partial P} = -T \frac{\partial V}{\partial T} - P \frac{\partial V}{\partial P}, \quad (65b)$$

may be used to demonstrate that (56b) is satisfied when $\kappa = -1/T$ and $\lambda = -P/T$. Alternatively, the relations (64)–(65) may be substituted into (56a) and (56b) and the resulting 2×2 linear system of equations may be solved for κ and λ to obtain $\kappa = -1/T$ and $\lambda = -P/T$.

5.4. Significance

We have demonstrated that the solution of the PT flash (57) in combination with the UV constraints (60) is identical to the solution of the UV flash (54). This implies that existing efficient algorithms for the PT flash may be combined with the UV constraints to obtain a solution that is identical to the solution of the UV flash problem (54). Similarly, the Lagrange multipliers of the UV flash problem may be computed from the UV constraints and the PT flash using

(63). The KKT matrix (20) of the UV flash problem (54) can thus be obtained without directly solving the UV flash problem.

6. Implementation

This section briefly describes the transformations that are made to the vapor–liquid equilibrium constraints in the implementation of the single-shooting algorithm. We furthermore discuss the details of the different software libraries and compilers that are used in the implementation.

6.1. Transformation of the UV flash

In the implementation, the optimality conditions (56) for the UV flash are replaced by the optimality conditions (59) for the PT flash combined with the UV constraints (60). We furthermore eliminate variables and scale equations. The resulting algebraic equations are similar to the equations that are solved in the Newton approach by (Michelsen and Mollerup, 2007, Chap. 14). We subtract (59b) from (59a) in order to eliminate the Lagrange multipliers, $\{\mu_i\}_{i=1}^{N_C}$, from the first order optimality conditions (59) for the PT flash. Furthermore, we eliminate the vapor mole numbers, n^v , from the linear mass balance constraints (57b):

$$n_i^v = n_i^v(n_i, n_i^l) = n_i - n_i^l, \quad i = 1, \dots, N_C. \quad (66)$$

The first order optimality conditions (59) are thus transformed to

$$\frac{\partial G^l}{\partial n_i^l}(T, P, n^l) - \frac{\partial G^v}{\partial n_i^v}(T, P, n^v) = 0, \quad i = 1, \dots, N_C, \quad (67)$$

which can be solved for the liquid mole numbers n^l . We rewrite the internal energy constraint (60a) using the thermodynamic relations $U^v = H^v - PV^v$ and $U^l = H^l - PV^l$:

$$\begin{aligned} U - H^v(T, P, n^v) + PV^v(T, P, n^v) \\ - H^l(T, P, n^l) + PV^l(T, P, n^l) = 0. \end{aligned} \quad (68)$$

We furthermore substitute the volume constraint (60b):

$$U + PV - H^v(T, P, n^v) - H^l(T, P, n^l) = 0. \quad (69)$$

The optimality conditions (67) and the internal energy constraint (69) are scaled with $1/(RT)$. The volume constraint (60b) is scaled with $P/(RT)$:

$$\frac{1}{RT} \left(\frac{\partial G^l}{\partial n_i^l}(T, P, n^l) - \frac{\partial G^v}{\partial n_i^v}(T, P, n^v) \right) = 0, \quad (70a)$$

$$\frac{1}{RT} (U + PV - H^l(T, P, n^l) - H^v(T, P, n^v)) = 0, \quad (70b)$$

$$\frac{P}{RT} (V^l(T, P, n^l) + V^v(T, P, n^v) - V) = 0. \quad (70c)$$

The internal energy, U , and the volume, V , are specified, and the algebraic equations (70) can be solved for the temperature, T , the pressure, P , and the liquid mole numbers, n^l . The temperature and pressure are, however, large in magnitude as compared to the mole numbers. The implementation therefore uses logarithmic values of the temperature and pressure as algebraic variables:

$$y = [\ln T; \ln P; n^l] \in \mathbb{R}^{2+N_C}. \quad (71)$$

The algebraic equations (70) are written compactly

$$G(x, y) = 0, \quad (72)$$

where x is defined in (50a).

6.2. Units of variables and scaling

Large differences in the elements of Jacobian matrices can lead to ill-conditioned linear systems of equations in the numerical solution schemes. Such ill-conditioning can furthermore lead to imprecise results. Several quantities are therefore not in SI-units in the implementation. The unit of time, t , is h, and the flow rates, F_F , F_V , and F_L , are given in kmol/h. The compositional flow rates, $f_{F,i}^v$, $f_{F,i}^l$, v_i , and l_i for $i = 1, \dots, N_C$, are also in kmol/h. Mole numbers, i.e. n_i , n_i^v , and n_i^l for $i = 1, \dots, N_C$, are given in kmol. The internal energy, U , and the enthalpies are in MJ, and the heat input, Q , is in MJ/h.

6.3. Software libraries, compilers, and hardware

The single-shooting algorithm is implemented in Matlab. The algorithm is implemented with both a simultaneous approach and a nested approach for the solution of the dynamic UV flash model equations. The Matlab implementation uses Matlab routines from ThermoLib as well as Matlab's function for constrained nonlinear optimization, `fmincon`. The single-shooting algorithm, with a simultaneous approach, is also implemented in C. The C implementation uses C routines from ThermoLib as well as either of four solvers for nonlinear constrained optimization: Matlab's `fmincon`, the open-source IPOPT 3.12.4 (Wächter and Biegler, 2006), the commercial NPSOL 5.0, or the commercial KNITRO 10.2. We use `fmincon`'s and KNITRO's interior point (IP) barrier methods. IPOPT also uses an IP barrier method, and NPSOL uses an active set sequential quadratic programming (SQP) method. We compare the performance obtained with GCC compilers and with Intel compilers. We use Netlib's linear algebra libraries (BLAS and LAPACK) together with the GCC compilers, and we use Intel's linear algebra library (Intel MKL) with the Intel compilers. ThermoLib and IPOPT are compiled with a C compiler, and NPSOL is compiled with a Fortran compiler. KNITRO is compiled prior to purchase. The performance study in Section 8 is carried out on a 64-bit workstation with 15.6 GB memory and four Intel Core i7 3.60 GHz cores. Each core has 64 kB of level 1 cache and 256 kB of level 2 cache. The workstation has a shared level 3 cache of 8192 kB.

7. Optimal control examples

This section presents two optimal control problems. The first is a tracking-type control problem where an ideal thermodynamic model is used. This problem has 144 decision variables. The second is an economical optimal control problem where a nonideal thermodynamic model is used. This problem has 864 decision variables.

7.1. Optimal tracking example

In this example, we consider a flash unit with a volume of 10 m^3 . The unit processes a mixture of benzene, toluene, and diphenyl. We use an ideal thermodynamic model for the computation of thermodynamic properties. This is reasonable because these three components are very similar and because we consider high ranges of temperature and low ranges of pressure. The time horizon is $[t_0, t_f] = [0 \text{ h}, 4 \text{ h}]$. The initial condition is a steady state which is described in Appendix A. The feed rate, F_F , increases by 50% at time $t = 2 \text{ h}$. The length of the control intervals is $\Delta t_k = 5 \text{ min}$ for $k \in \mathcal{N}$. That is a total of 48 control intervals during the 4 h time horizon. The controls are: the heat input, Q , the total vapor flow rate, F_V , and the total liquid flow rate, F_L . The objective is to keep the temperature, pressure, and liquid volume at desired setpoints.

7.1.1. Objective function

The objective function, ϕ , is the integral of the squared differences between the controlled variables and the setpoints:

$$\phi = \int_{t_0}^{t_f} \left[\alpha_T (\ln T - \ln T^{\text{set}})^2 + \alpha_P (\ln P - \ln P^{\text{set}})^2 + \alpha_{Vl} (V^l - V^{l,\text{set}})^2 \right] dt. \quad (73)$$

The controlled variables include logarithmic temperature and pressure because of the choice of algebraic variables described in Section 6. The weights ($\alpha_T, \alpha_P, \alpha_{Vl}$) determine the relative importance of following each setpoint. The objective function (73) is supplemented with a regularization term that penalizes: (a) the temporal change in the controls and (b) the difference between the controls in the first control interval and a set of initial reference controls: $Q_{-1} = -1 \text{ MJ/h}$, $F_{L,-1} = 0.6 \text{ kmol/h}$ and $F_{V,-1} = 0.4 \text{ kmol/h}$. The discrete objective is augmented with this regularization term:

$$\hat{\Phi}_k = \Phi_k + \left[\alpha_Q (Q_k - Q_{k-1})^2 + \alpha_{F_V} (F_{V,k} - F_{V,k-1})^2 + \alpha_{F_L} (F_{L,k} - F_{L,k-1})^2 \right] \Delta t_k, k \in \mathcal{N}. \quad (74)$$

The weights ($\alpha_Q, \alpha_{F_V}, \alpha_{F_L}$) determine the size of the penalty of temporal changes in the controls. The weights are $(\alpha_T, \alpha_P, \alpha_{Vl}) = (2000, 20, 2000)$ and $(\alpha_Q, \alpha_{F_V}, \alpha_{F_L}) = (0.05, 10, 10)$. The derivatives of the additional regularization terms are derived analytically and added to the derivatives that are computed with the adjoint algorithm.

7.1.2. Constraints

The sum of the vapor and liquid stream flow rates is bounded from above by a factor times the feed flow rate:

$$F_{V,k} + F_{L,k} \leq \alpha_F F_{F,k}, k \in \mathcal{N}. \quad (75)$$

The factor is $\alpha_F = 1.2$. Furthermore, the controls are subject to the following bound constraints

$$Q_k \in [-60 \text{ MJ/h}, 10 \text{ MJ/h}], \quad (76a)$$

$$F_{L,k} \in [0.1 \text{ kmol/h}, 1.5 \text{ kmol/h}], \quad (76b)$$

$$F_{V,k} \in [0.1 \text{ kmol/h}, 1.5 \text{ kmol/h}], \quad (76c)$$

for $k \in \mathcal{N}$. The derivatives of the linear constraints (75) and the bound constraints (76) are derived analytically.

7.1.3. Optimal control strategy

Fig. 3(a) shows the controlled variables and the vapor fraction obtained with the optimal strategy (blue solid) and with a reference strategy (green dashed). A step occurs in the setpoints (black dashed) at time $t = 2 \text{ h}$. The controlled variables successfully follow the setpoints both before and after the step. The large variations in the temperature and pressure significantly affect the vapor–liquid equilibrium. This is evident from the large variations in the vapor fraction and in the vapor–liquid composition shown in Fig. 4. Fig. 3(b) shows the optimal strategy, the reference strategy, and the bound constraints (red dashed). The reference strategy is constructed such that the controlled variables eventually reach the setpoints after the step. However, the transition is significantly faster with the optimal strategy. This is because the optimal strategy uses the information about the step in the setpoints to initiate the transition already around $t = 1 \text{ h}$.

7.2. Minimal cooling example

In this example, we consider a flash unit with a volume of 1 m^3 . The unit processes a mixture of methane, ethane, propane, n-heptane, and hydrogen sulfide (H_2S). We use a nonideal thermodynamic model based on the PR EOS to compute thermodynamic properties. The time horizon is $[t_0, t_f] = [0 \text{ h}, 24 \text{ h}]$. The initial condition is a steady state which is described in Appendix A. The feed

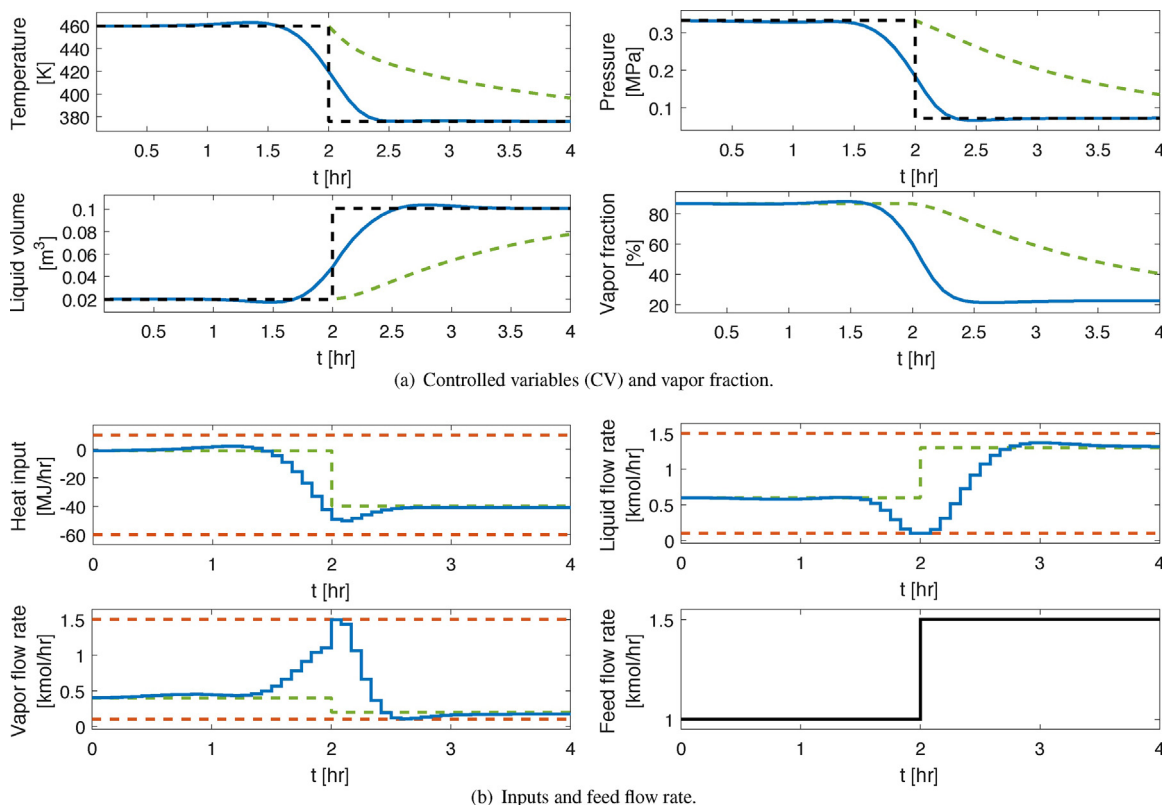


Fig. 3. The optimal tracking of temperature, pressure, and liquid volume setpoints for a ternary mixture containing benzene, toluene, and diphenyl. Blue solid: optimal strategy. Green dashed: reference strategy. Red dash-dotted: bounds. Black solid: feed. Black dashed: setpoints. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of the article.)

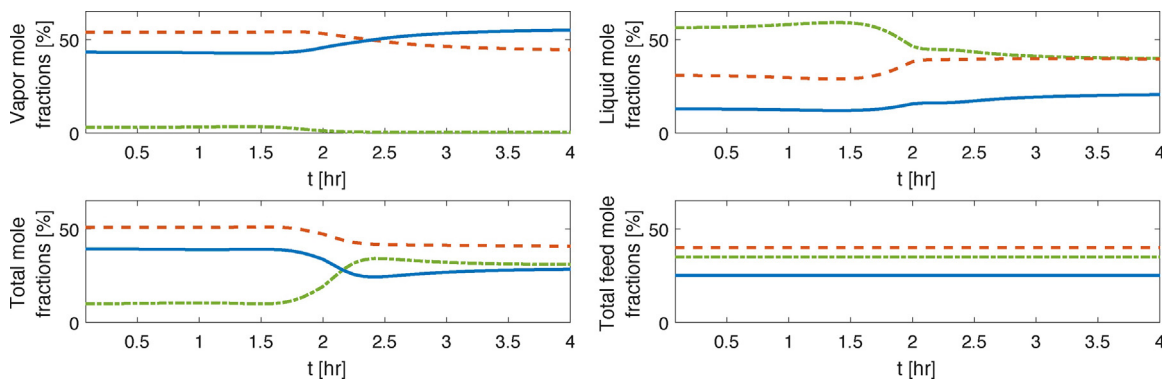


Fig. 4. The mixture mole fractions and the total feed mole fractions obtained with the optimal tracking strategy. Blue solid: benzene. Red dashed: toluene. Green dash-dotted: diphenyl. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of the article.)

composition changes at time $t = 12$ h such that the total feed mole fraction of H_2S increases from 2.0% to 6.0%. The length of the control intervals is $\Delta t_k = 5$ min for $k \in \mathcal{N}$, which gives a total of 288 control intervals during the 24 h time horizon. The objective is to minimize the energy consumption (in terms of cooling). Meanwhile, an upper bound on the H_2S vapor mole fraction should be satisfied.

7.2.1. Objective function

Since the heat input is negative (non-positive), $Q \leq 0$, in the case of cooling, the objective function is

$$\phi = - \int_{t_0}^{t_f} Q dt. \quad (77)$$

We incorporate bound constraints on the H_2S vapor mole fraction, y_{H_2S} , and on the overall vapor fraction, β , into the objective

function using logarithmic barrier functions. The H_2S vapor mole fraction is bounded from above: $y_{H_2S,k} \leq y_{H_2S,k}^{\max}$ for $k = 1, \dots, N$. The overall vapor fraction is bounded from above and below: $\epsilon_\beta \leq \beta_k \leq 1 - \epsilon_\beta$ for $k = 1, \dots, N$. The discrete objective function is augmented with the logarithmic barrier functions in the adjoint algorithm:

$$\Phi_k = -Q_k \Delta t_k - \alpha_{H_2S} \ln \left(y_{H_2S,k+1}^{\max} - y_{H_2S,k+1} \right) - \alpha_\beta \left[\ln (\beta_{k+1} - \epsilon_\beta) + \ln ((1 - \epsilon_\beta) - \beta_{k+1}) \right], \quad (78)$$

for $k \in \mathcal{N}$. The first term is a term in the discrete approximation of the objective function (77). The second term represents the upper bound on the H_2S vapor mole fraction, and the third term represents the bounds on the vapor fraction. The weights are

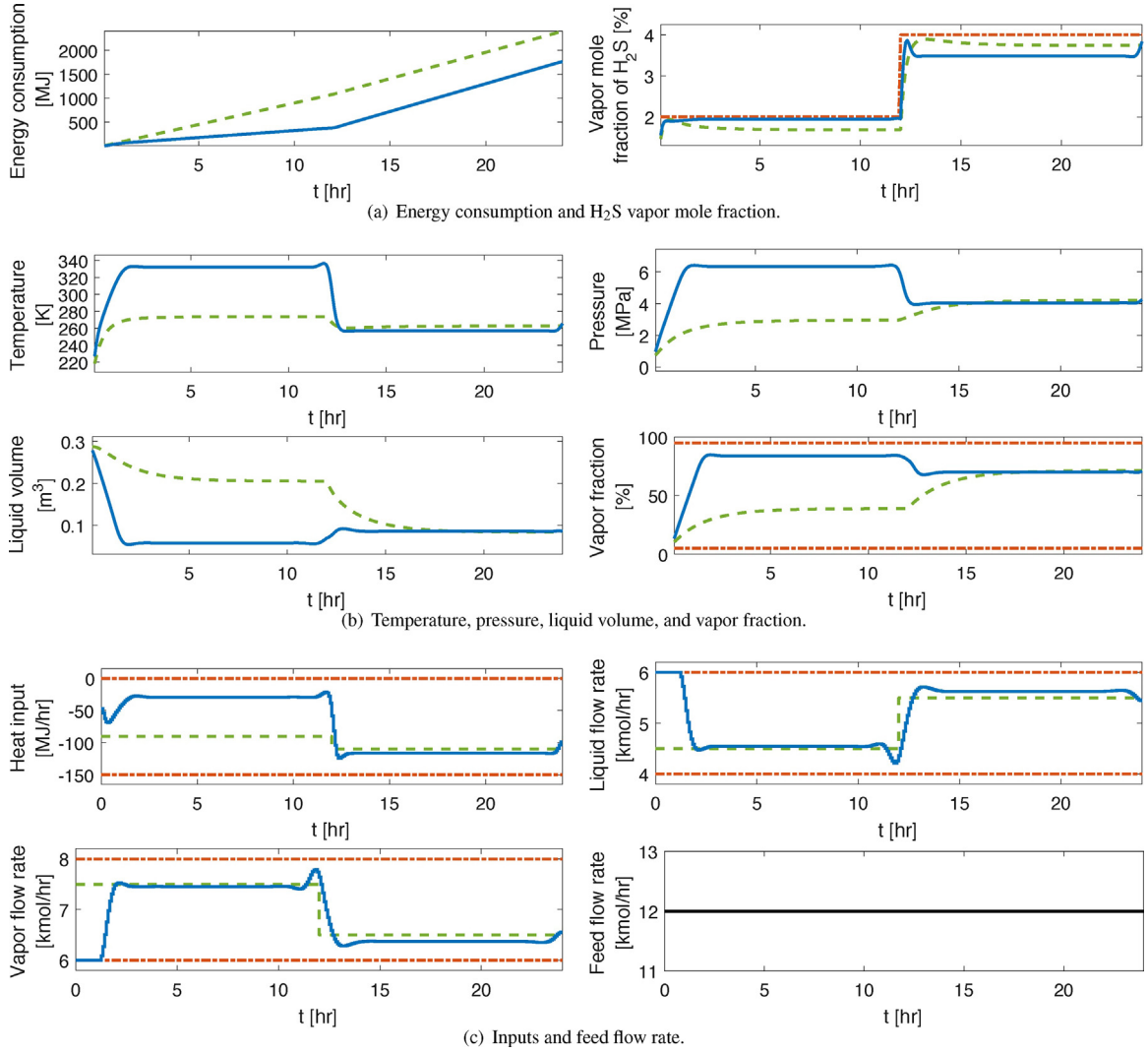


Fig. 5. Optimal flash separation of a mixture of four hydrocarbons and hydrogen-sulfide with minimal energy consumption (in terms of cooling). Blue solid: optimal strategy. Green dashed: reference strategy. Red dash-dotted: bounds. Black solid: feed. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of the article.)

$(\alpha_{\text{H}_2\text{S}}, \alpha_\beta) = (1.0, 0.6)$, and the vapor fraction threshold is $\epsilon_\beta = 0.05$. The upper bound on the H₂S vapor mole fraction is

$$y_{\text{H}_2\text{S},k}^{\max} = \begin{cases} 0.02, & k = 1, \dots, N/2, \\ 0.04, & k = N/2 + 1, \dots, N, \end{cases} \quad (79)$$

i.e. the upper bound is 2% during the first 12 h and 4% during the last 12 h. The objective function (78) is furthermore augmented with a regularization term that penalizes the temporal change in the controls:

$$\hat{\Phi}_k = \Phi_k + \left[\alpha_Q (Q_k - Q_{k-1})^2 + \alpha_{F_V} (F_{V,k} - F_{V,k-1})^2 + \alpha_{F_L} (F_{L,k} - F_{L,k-1})^2 \right] \Delta t_k, \quad k = 1, \dots, N - 1. \quad (80)$$

The weights $(\alpha_Q, \alpha_{F_V}, \alpha_{F_L})$ determine the size of the penalty of the temporal change in each control variable. The weights are $(\alpha_Q, \alpha_{F_V}, \alpha_{F_L}) = (0.1, 10, 10)$. The derivatives of this additional regularization term are derived analytically. In this problem, there is

no set of initial reference controls as was the case in the optimal tracking problem in Section 7.1.

7.2.2. Constraints

The total mass supplied by the feed should be equal to the total mass extracted by the vapor and liquid streams:

$$F_{V,k} + F_{L,k} = F_{F,k}, \quad k \in \mathcal{N}. \quad (81)$$

Furthermore, the controls are subject to the following bound constraints

$$Q_k \in [-150 \text{ MJ/h}, 0 \text{ MJ/h}], \quad (82a)$$

$$F_{L,k} \in [4 \text{ kmol/h}, 6 \text{ kmol/h}], \quad (82b)$$

$$F_{V,k} \in [6 \text{ kmol/h}, 8 \text{ kmol/h}], \quad (82c)$$

for $k \in \mathcal{N}$. The derivatives of the linear constraints (81) and the bound constraints (82) are derived analytically.

7.2.3. The optimal strategy

Fig. 5(a) shows the energy consumption and the H₂S vapor mole fraction obtained with the optimal strategy (blue solid) and with a reference strategy (green dashed). The optimal strategy reduces the total energy consumption from 2400 MJ to around 1700 MJ. This is

Table 1
Performance indicators for the solution of the two optimal control problems using the single-shooting algorithm with the simultaneous approach (Sim.) and the nested approach (Nest.). The implementations use either an exact or an inexact Newton method. Sim. time and Adj. time are the relative computation times of the simulation and the adjoint gradient computations. There is no relation between the relative computation times in the two examples. NLP Iter. and NLP Func. are the number of iterations and function evaluations performed by Matlab's fmincon, respectively. Fact. is the number of factorizations and Back. sub. is the number of back substitutions. Furthermore, the number of evaluations of thermodynamic properties for evaluating residual equations and for evaluating Jacobians are shown.

Optimal tracking problem								
	Sim. time	Adj. time	NLP Iter.	NLP Func.	Fact.	Back. sub.	Thermo. for residual eq.	Thermo. for Jacobian
Sim. (inexact)	0.271	0.118	202	203	20,881	42,946	33,202	20,881
Sim. (exact)	0.264	0.118	202	203	32,632	32,632	0	42,376
Nest. (inexact)	1.000	0.145	202	203	138,358	179,585	105,328	118,853
Nest. (exact)	0.680	0.140	202	203	142,507	142,507	0	113,792
Minimal cooling problem								
	Sim. time	Adj. time	NLP Iter.	NLP Func.	Fact.	Back. sub.	Thermo. for residual eq.	Thermo. for Jacobian
Sim. (inexact)	0.184	0.094	156	173	103,763	159,501	109,677	103,763
Sim. (exact)	0.300	0.116	147	213	166,462	166,462	0	227,806
Nest. (inexact)	1.000	0.129	150	230	738,140	889,473	475,960	605,349
Nest. (exact)	0.792	0.123	146	205	654,674	654,674	0	507,831

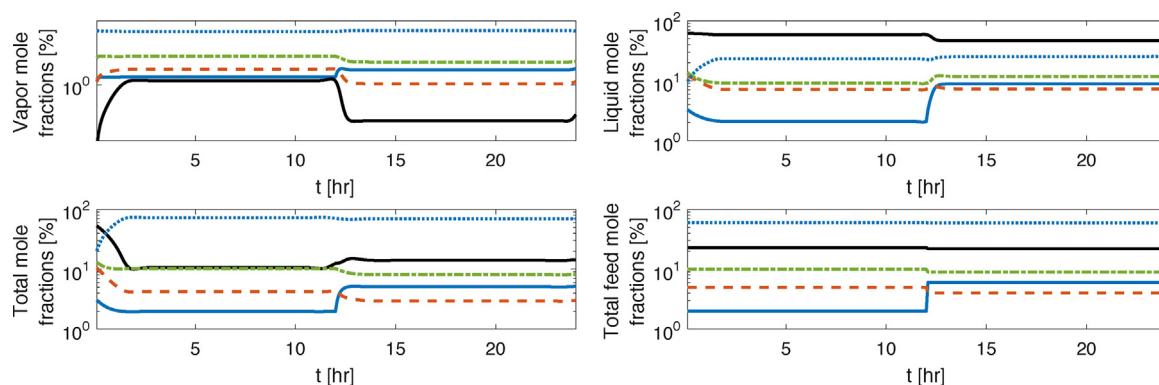


Fig. 6. The mixture mole fractions and the total feed mole fractions for the minimal cooling strategy. Blue dotted: methane. Green dash-dotted: ethane. Red dashed: propane. Black solid: heptane. Blue solid: hydrogen sulfide. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of the article.)

a reduction of 26%. The upper bound on the H_2S vapor mole fraction is satisfied at all times. Fig. 5(b) shows the temperature, the pressure, the liquid volume, and the vapor fraction. The large variations in the temperature and pressure cause large variations in the vapor fraction. However, the vapor–liquid composition is not affected significantly as can be seen from Fig. 6. The vapor consists primarily of methane, and the liquid phase is primarily heptane and methane. Fig. 5(c) shows the optimal strategy and the reference strategy. The optimal strategy uses the vapor flow rate and the liquid flow rate to compensate for the increase in the heat input during the first 12 h. However, the optimal strategy does not increase the heat input during the last 12 h. This is because the upper constraint on the H_2S vapor mole fraction is incorporated into the objective function. This undesired behavior can be avoided by adjusting the parameter α_{H_2S} in the barrier function during the iterations of the optimization algorithm.

8. Performance study

This section presents a performance study of several implementations of the single-shooting algorithm. The algorithm is used to solve the optimal tracking problem described in Section 7.1 and the minimal cooling problem described in Section 7.2. We compare the efficiency of: (a) the simultaneous and the nested approach, (b) the exact and the inexact Newton method, (c) Matlab and C implementations, (d) different NLP solvers, and (e) different compilers and linear algebra libraries.

8.1. The simultaneous and the nested approaches

Table 1 shows several performance measures for a Matlab implementation of the single-shooting algorithm. The simultaneous approach is faster than the nested approach for both problems and regardless of whether an exact or an inexact Newton method is used. This is due to fewer factorizations, back substitutions, and evaluations of thermodynamic properties. Table 2 shows the computation times of thermodynamic function evaluations, factorizations, and back substitutions. The evaluation of thermodynamic functions and their derivatives is more time consuming than solving linear systems for both the Matlab and the C implementations. However, the difference is significantly larger for the Matlab implementations. There is little or no computational overhead in evaluating thermodynamic functions for the Jacobian instead of for the residual equations when an ideal thermodynamic model is used. This is the case for the optimal tracking problem. The overhead is significantly larger when equations of state are used in the thermodynamic computations, as is the case for the minimal cooling problem.

8.2. Exact and inexact Newton methods

For both problems, it is more efficient to use an exact Newton method in the nested approach. This is likely due to frequent updating of the Jacobian in the inner Newton iterations. For the simultaneous approach, there is no advantage in using an inexact Newton method when solving the optimal tracking problem.

Table 2

Absolute computation time (in microseconds) of evaluating thermodynamical properties and solving a linear system $Ax = b$ for x where A is a dense matrix. The sizes of A are chosen according to the sizes of the matrices that occur in the Newton iterations in the simultaneous approach (Sim.) and in the inner and outer Newton iterations in the nested approach (Nest. inner and Nest. outer, respectively). We distinguish between evaluating thermodynamic properties to be used in residual equations and to be used in Jacobians. The linear systems are solved with an LU factorization (Fact.) and a back substitution (Back. sub.). The C factorizations and back substitutions are obtained with Netlib's LAPACK routines `dgetrf` and `dgetrs`.

	Optimal tracking problem							
	Thermo. eval.		Sim. ($A \in \mathbb{R}^{9 \times 9}$)		Nest. inner ($A \in \mathbb{R}^{5 \times 5}$)		Nest. outer ($A \in \mathbb{R}^{4 \times 4}$)	
	For residual eq.	For Jacobian	Fact.	Back. sub.	Fact.	Back. sub.	Fact.	Back. sub.
Matlab	34.588	39.829	3.514	2.271	2.844	2.117	2.634	1.887
C	1.435	1.428	0.575	0.188	0.205	0.112	0.145	0.092
	Minimal cooling problem							
	Thermo. eval.		Sim. ($A \in \mathbb{R}^{31 \times 31}$)		Nest. inner ($A \in \mathbb{R}^{16 \times 16}$)		Nest. outer ($A \in \mathbb{R}^{15 \times 15}$)	
	For residual eq.	For Jacobian	Fact.	Back. sub.	Fact.	Back. sub.	Fact.	Back. sub.
Matlab	211.319	464.254	5.121	2.516	3.169	2.213	3.032	2.151
C	4.305	6.431	1.165	0.290	0.359	0.150	0.288	0.129

Table 3

Absolute computation time (in seconds) and relative computation time of solving the two optimal control problems using the single-shooting algorithm with the simultaneous approach. Average over ten solutions to the optimal control problem.

NLP solver	Optimal tracking problem								
	fmincon	GCC compilers and Netlib's BLAS/LAPACK				Intel compilers and Intel MKL			
		fmincon	IPOPT	KNITRO	NPSOL	fmincon	IPOPT	KNITRO	NPSOL
Prog. Lang.	Matlab	C	C	C	C	C	C	C	C
Iterations	202	202	486	186	152	202	486	186	152
Func. Eval.	203	203	1517	190	153	203	1517	190	153
Absolute	8.503	0.851	1.319	0.291	0.223	0.843	0.911	0.245	0.102
Relative	1.000	0.100	0.155	0.034	0.026	0.099	0.107	0.029	0.012
Speedup	1.0	10.0	6.4	29.2	38.1	10.1	9.3	34.7	83.4
NLP solver	Minimal cooling problem								
	fmincon	GCC compilers and Netlib's BLAS/LAPACK				Intel compilers and Intel MKL			
		fmincon	IPOPT	KNITRO	NPSOL	fmincon	IPOPT	KNITRO	NPSOL
Prog. Lang.	Matlab	C	C	C	C	C	C	C	C
Iterations	156	156	328	117	150	156	328	117	150
Func. Eval.	173	173	1364	121	152	173	1364	121	152
Absolute	137.803	37.525	6.868	6.654	7.002	37.929	6.437	8.149	2.481
Relative	1.000	0.272	0.050	0.048	0.051	0.275	0.047	0.059	0.018
Speedup	1.0	3.7	20.1	20.7	19.7	3.6	21.4	16.9	55.5

This is because the overhead associated with evaluating thermodynamic properties for the Jacobians is very small. This is not true when equations of state are used. For the minimal cooling problem, where equations of state are used, the performance of the simultaneous approach thus improves by roughly 35% when an inexact Newton method is used. This improvement is also, in part, due to the fewer function evaluations in the NLP algorithm.

8.3. Programming language, NLP solvers, and compilers

Table 3 shows performance measures for several implementations of the single-shooting algorithm using the simultaneous approach with an inexact Newton method. The implementations use different optimization libraries (NLP solvers), compilers, and linear algebra libraries. The least efficient is a pure Matlab implementation using `fmincon`. This implementation solves the optimal tracking problem in 8.5 s and the minimal cooling problem in 137.8 s. A mixed implementation, which uses `fmincon` and C routines for the simulation and the gradient computations, improves over the pure Matlab implementation. It attains a speedup of 10 for the optimal tracking problem and 3.7 for the minimal cooling problem, as compared to the pure Matlab implementation.

The C implementation using IPOPT is not efficient for the optimal tracking problem. This is because IPOPT is used with a limited-memory BFGS approximation. This implementation is more efficient for the larger minimal cooling problem where it performs as well as the implementations using KNITRO and NPSOL when GCC compilers are used. The implementation that uses NPSOL is in general the fastest. Its performance improves significantly when Intel compilers are used. It thus attains a speedup of 83 for the optimal tracking problem and 55 for the minimal cooling problem. The Intel compilers do not improve the performance of the other implementations significantly.

9. Conclusions

We have presented a single-shooting algorithm for gradient-based dynamic optimization of vapor–liquid equilibrium processes. It uses the adjoint method for the computation of gradients. We have described a dynamic UV flash model and demonstrated that the UV flash is equivalent to a UV-constrained PT flash, i.e. a PT flash combined with specifications of the internal energy and volume. We have presented a numerical performance study with two optimal control problems. The first problem is a small tracking-type control problem. The second is a medium-size economical con-

control problem. We have compared the efficiency of a simultaneous approach and a nested approach for the numerical solution of the differential-algebraic constraints in the optimal control problem. The simultaneous approach is faster because it uses fewer factorizations, back substitutions, and thermodynamic function evaluations. Not surprisingly, the performance study also shows that the C implementations are significantly faster than a pure Matlab implementation. Furthermore, we have compared the performance of four nonlinear constrained optimization solvers: Matlab's *fmincon*, IPOPT, KNITRO, and NPSOL. NPSOL is generally the fastest solver for this problem formulation, and the performance of NPSOL is significantly improved when Intel compilers are used. The Intel compilers do not improve the efficiency of the implementations using *fmincon*, IPOPT, and KNITRO significantly. The implementation using NPSOL is able to solve the tracking problem in 0.1 s which is 83.4 times faster than the pure Matlab implementation. Furthermore, it solves the economical control problem in 2.5 s which is 55.5 times faster than the Matlab implementation.

Acknowledgements

We would like to acknowledge the financial support from: (1) Innovation Fund Denmark in the CITIES project (1305-00027B) and in the OPTION project (63-2013-3), (2) the interreg project Smart Cities Accelerator (10606 SCA), and (3) EU DP 64013-0558 in the IEA Annex for energy efficient process control.

Appendix A. Parameter values for the optimal control examples

Table A.4 contains values of quantities related to the optimal tracking problem described in Section 7.1. It contains the control variables and the disturbance variables necessary for computing the initial and final steady states that are used to generate the set-points in the objective function (73). This initial steady state is also used as the initial condition. Furthermore, it contains the feed variables as well as the reference strategy. Table A.5 contains values

Table A.4

Parameter values used in the optimal tracking problem in Section 7.1. Certain parameters have different values on the first and the second half of the time interval $[t_0, t_f] = [0 \text{ h}, 4 \text{ h}]$.

	$t \in [0 \text{ h}, 4 \text{ h}]$	Unit	
Initial steady state			
T_F	505.0	K	
P_F	1.0	MPa	
z_F	[0.25; 0.40; 0.35]		
F_F	1.0	kmol/h	
Q	-1.0	MJ/h	
F_L	0.6	kmol/h	
F_V	0.4	kmol/h	
Final steady state			
T_F	505.0	K	
P_F	1.0	MPa	
z_F	[0.25; 0.40; 0.35]		
F_F	1.5	kmol/h	
Q	-40.0	MJ/h	
F_L	1.3	kmol/h	
F_V	0.2	kmol/h	
	$t \in [0 \text{ h}, 2 \text{ h}]$	$t \in [2 \text{ h}, 4 \text{ h}]$	Unit
Feed			
T_F	505.0	505.0	K
P_F	1.0	1.0	MPa
z_F	[0.25; 0.40; 0.35]	[0.25; 0.40; 0.35]	
F_F	1.0	1.5	kmol/h
Reference control strategy			
Q	-1.0	-40.0	MJ/h
F_L	0.6	1.3	kmol/h
F_V	0.4	0.2	kmol/h

Table A.5

Parameter values used in the minimal cooling problem in Section 7.2. Certain parameters have different values on the first and the second half of the time interval $[t_0, t_f] = [0 \text{ h}, 24 \text{ h}]$.

	$t \in [0 \text{ h}, 24 \text{ h}]$	Unit	
Initial steady state			
T_F	335.15	K	
P_F	1.0	MPa	
z_F	[0.60; 0.10; 0.05; 0.23; 0.02]		
F_F	12.0	kmol/h	
Q	-150.0	MJ/h	
F_L	4.5	kmol/h	
F_V	7.5	kmol/h	
	$t \in [0 \text{ h}, 12 \text{ h}]$	$t \in [12 \text{ h}, 24 \text{ h}]$	Unit
Feed			
T_F	335.15	335.15	K
P_F	1.0	1.0	MPa
z_F	[0.60; 0.10; 0.05; 0.23; 0.02]	[0.59; 0.09; 0.04; 0.22; 0.06]	
F_F	12.0	12.0	kmol/h
Reference control strategy			
Q	-90.0	-110.0	MJ/h
F_L	4.5	5.5	kmol/h
F_V	7.5	6.5	kmol/h

of quantities related to the minimal cooling problem described in Section 7.2. It contains the control variables and the disturbance variables used to compute the initial steady state that is used as initial condition in the minimal cooling problem. Furthermore, the table contains the feed variables and the reference strategy.

References

- Arendsen, A.R.J., Versteeg, G.F., 2009. Dynamic thermodynamics with internal energy, volume, and amount of moles as states: application to liquefied gas tank. *Ind. Eng. Chem. Res.* 48, 3167–3176.
- Barton, P.I., Lee, C.K., 2002. Modeling, simulation, sensitivity analysis, and optimization of hybrid systems. *ACM Trans. Model. Comput. Simul.*, 256–289.
- Betts, J.T., 2001. *Practical Methods for Optimal Control Using Nonlinear Programming*. Advances in Design and Control. SIAM.

- Biegler, L.T., 2007. An overview of simultaneous strategies for dynamic optimization. *Chem. Eng. Process. Process Intensif.* 46, 1043–1053.
- Biegler, L.T., 2010. *Nonlinear Programming: Concepts, Algorithms, and Applications to Chemical Processes*. SIAM.
- Binder, T., Blank, L., Bock, H.G., Bulirsch, R., Dahmen, W., Diehl, M., Kroneder, T., Marquardt, W., Schlöder, J.P., von Stryk, O., 2001. Introduction to model based optimization of chemical processes on moving horizons. In: *Online Optimization of Large Scale Systems*. Springer, pp. 295–339.
- Bisgaard, T., Huusom, J.K., Abildskov, J., 2015. Modeling and analysis of conventional and heat-integrated distillation columns. *AIChE J.* 61, 4251–4263.
- Bisgaard, T., Skogestad, S., Abildskov, J., Huusom, J.K., 2017. Optimal operation and stabilizing control of the concentric heat-integrated distillation column (HIDIC). *Comput. Chem. Eng.* 96, 196–211.
- Bock, H.G., Plitt, K.J., 1984. A multiple shooting algorithm for direct solution of optimal control problems. In: *Proceedings 9th IFAC World Congress*. Pergamon Press, pp. 242–247.
- Bryson Jr., A.E., 1999. *Dynamic Optimization*. Addison-Wesley.
- Capolei, A., Jørgensen, J., 2012. Solution of constrained optimal control problems using multiple shooting and ESDIRK methods. *Proceedings of the 2012 American Control Conference*, 295–300.
- Capolei, A., Völcker, C., Frydendall, J., Jørgensen, J.B., 2012. Oil reservoir production optimization using single shooting and ESDIRK methods. *Automatic Control in Offshore Oil and Gas Production*, International Federation of Automatic Control, 286–291.
- Castier, M., 2009. Solution of the isochoric–isoenergetic flash problem by direct entropy maximization. *Fluid Phase Equilib.* 276, 7–17.
- Castier, M., 2010. Dynamic simulation of fluids in vessels via entropy maximization. *J. Ind. Eng. Chem.* 16, 122–129.
- Colson, B., Marcotte, P., Savard, G., 2007. An overview of bilevel optimization. *Ann. Oper. Res.* 153, 235–256.
- Diehl, M., Bock, H.G., Schlöder, J.P., Findeisen, R., Nagy, Z., Allgöwer, F., 2002. Real-time optimization and nonlinear model predictive control of processes governed by differential-algebraic equations. *J. Process Control* 12, 577–585.
- Diehl, M., Ferreau, H.J., Haverbeke, N., 2009. Efficient numerical methods for nonlinear MPC and moving horizon estimation. In: Magni, L., Raimondo, D.M., Allgöwer, F. (Eds.), *Nonlinear Model Predictive Control. Towards New Challenging Application*. Springer. Volume 384 of Lecture Notes in Control and Information Sciences.
- Flatby, P., Skogestad, S., Lundström, P., 1994. Rigorous dynamic simulation of distillation columns based on UV-flash. *IFAC Symposium on Advanced Control of Chemical Processes (ADCHEM '94)*, 261–266.
- Hammer, M., Morin, A., 2014. A method for simulating two-phase pipe flow with real equations of state. *Comput. Fluids* 100, 45–58.
- Ingebritsen, S.E., Geiger, S., Hurwitz, S., Driesner, T., 2010. Numerical simulation of magmatic hydrothermal systems. *Rev. Geophys.* 48, 1–33.
- Jørgensen, J.B., 2007. Adjoint sensitivity results for predictive control, state- and parameter-estimation with nonlinear models. In: *European Control Conference (ECC)*, IEEE, pp. 3649–3656.
- Kourounis, D., Durlöfsky, L.J., Jansen, J.D., Aziz, K., 2014. Adjoint formulation and constraint handling for gradient-based optimization of compositional reservoir flow. *Comput. Geosci.* 18, 117–137.
- Kristensen, M.R., Jørgensen, J.B., Thomsen, P.G., Jørgensen, S.B., 2004. An ESDIRK method with sensitivity analysis capabilities. *Comput. Chem. Eng.* 28, 2695–2707.
- Kristensen, M.R., Jørgensen, J.B., Thomsen, P.G., Michelsen, M.L., Jørgensen, S.B., 2005. Sensitivity analysis in index-1 differential algebraic equations by ESDIRK methods. In: *Proceedings of the 16th IFAC World Congress, Prague, Czech Republic*, pp. 212–217.
- Laiglecia, J.I., Lopez-Negrete, R., Diaz, M.S., Biegler, L.T., 2012. A simultaneous dynamic optimization approach for natural gas processing plants. *Proceedings of Foundations of Computer Aided Process Operations (FOCAPO)*.
- Li, Y., Johns, R.T., 2006. Rapid flash calculations for compositional simulation. *SPE Reserv. Eval. Eng. SPE* 95732, 521–529.
- Lima, E.R.A., Castier, M., Bisciaia Jr., E.C., 2008. Differential-algebraic approach to dynamic simulations of flash drums with rigorous evaluation of physical properties. *Oil Gas Sci. Technol. Rev. IFP* 63, 677–686.
- Lucia, A., Bonk, B.M., Waterman, R.A., Roy, A., 2012. A multi-scale framework for multi-phase equilibrium flash. *Comput. Chem. Eng.* 36, 79–98.
- Luo, Z.Q., Pang, J.S., Ralph, D., 1996. *Mathematical Programs with Equilibrium Constraints*. Cambridge University Press.
- Luyben, W.L. (Ed.), 1992. *Practical Distillation Control*. Van Nostrand Reinhold, New York.
- Michelsen, M., Mollerup, J., 2007. *Thermodynamic Models: Fundamentals and Computational Aspects*. Tie-Line Publications.
- Michelsen, M.L., 1999. State function based flash specifications. *Fluid Phase Equilib.* 158, 617–626.
- Outrata, J., Kocvara, M., Zowe, J., 2013. *Nonsmooth Approach to Optimization Problems with Equilibrium Constraints: Theory, Applications and Numerical Results*, vol. 28. Springer Science & Business Media.
- Qiu, L., Wang, Y., Reitz, R.D., 2014. Multiphase dynamic flash simulations using entropy maximization and application to compressible flow with phase change. *AIChE J.* 60, 3013–3024.
- Ragunathan, A.U., Diaz, M.S., Biegler, L.T., 2004. An MPEC formulation for dynamic optimization of distillation operations. *Comput. Chem. Eng.* 28, 2037–2052.
- Ritschel, T.K.S., Capolei, A., Jørgensen, J.B., 2017a. The adjoint method for gradient-based optimization of UV flash processes. In: Espuna, A., Graells, M., Puigjaner, L. (Eds.), *27th European Symposium on Computer Aided Process Engineering – ESCAPE 27*. Barcelona, Spain.
- Ritschel, T.K.S., Capolei, A., Jørgensen, J.B., 2017b. Dynamic optimization of UV flash processes. In: *FOCAPO/CPC 2017*, Tucson, Arizona.
- Ritschel, T.K.S., Gaspar, J., Capolei, A., Jørgensen, J.B., 2016. An Open-source Thermodynamic Software Library. Technical Report DTU Compute Technical Report-2016-12. Department of Applied Mathematics and Computer Science, Technical University of Denmark.
- Ritschel, T.K.S., Gaspar, J., Jørgensen, J.B., 2017c. A thermodynamic library for simulation and optimization of dynamic processes. *Proceedings of the 20th World Congress of the International Federation of Automatic Control*.
- Saha, S., Carroll, J.J., 1997. The isoenergetic–isochoric flash. *Fluid Phase Equilib.* 138, 23–41.
- Sahlodin, A.M., Watson, H.A.J., Barton, P.I., 2016. Nonsmooth model for dynamic simulation of phase changes. *AIChE J.* 62, 3334–3351.
- Schäfer, A., Kühl, P., Diehl, M., Schlöder, J., Bock, H.G., 2007. Fast reduced multiple shooting methods for nonlinear model predictive control. *Chem. Eng. Process. Process Intensif.* 46, 1200–1214.
- Smith, J.M., Van Ness, H.C., Abbott, M.M., 2005. *Introduction to Chemical Engineering Thermodynamics*, 7th ed. McGraw-Hill, New York, NY.
- Stauffer, P.H., Viswanathan, H.S., Pawar, R.J., Guthrie, G.D., 2009. A system model for geologic sequestration of carbon dioxide. *Environ. Sci. Technol.* 43, 565–570.
- Stichlmair, J.G., Fair, J.R., 1998. *Distillation. Principles and Practice*. Wiley-VCH, New York.
- Thomson, G., 1996. The DIPPR® databases. *Int. J. Thermophys.* 17, 223–232.
- Tolsma, J.E., Barton, P.I., 2000. DAEPACK an open modeling environment for legacy models. *Ind. Eng. Chem. Res.* 39, 1826–1839.
- Völcker, C., Jørgensen, J.B., Stenby, E.H., 2011. Oil reservoir production optimization using optimal control. In: *IEEE Conference on Decision and Control 2011, IEEE Control Systems Society*, pp. 7937–7943.
- Völcker, C., Jørgensen, J.B., Thomsen, P.G., Stenby, E.H., 2010. Adaptive stepsize control in implicit Runge–Kutta methods for reservoir simulation. *Proceedings of the 9th International Symposium on Dynamics and Control of Process Systems (DYCOPS 2010)*, 509–514.
- Wächter, A., Biegler, L.T., 2006. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Math. Programm.* 106, 25–57.
- Watson, H.A.J., Vikse, M., Gundersen, T., Barton, P.I., 2017. Reliable flash calculations: part 1. Nonsmooth inside-out algorithms. *Ind. Eng. Chem. Res.* 56, 960–973.
- Wilhelmsen, Ø., Skaugen, G., Hammer, M., Wahl, P.E., Morud, J.C., 2013. Time efficient solution of phase equilibria in dynamic and distributed systems with differential algebraic equation solvers. *Ind. Eng. Chem. Res.* 52, 2130–2140.
- Zavala, V.M., Biegler, L.T., 2009. Nonlinear programming strategies for state estimation and model predictive control. In: Magni, L., Raimondo, D.M., Allgöwer, F. (Eds.), *Nonlinear Model Predictive Control. Towards New Challenging Applications*. Springer. Volume 384 of Lecture Notes in Control and Information Sciences. , pp. 419–432.
- Zaydullin, R., Voskov, D.V., James, S.C., Henley, H., Lucia, A., 2014. Fully compositional and thermal reservoir simulation. *Comput. Chem. Eng.* 63, 51–65.

Paper VI

Computation of Phase Equilibrium in Reservoir Simulation and Optimization

Authors:

Tobias K. S. Ritschel and John Bagterp Jørgensen

Published in:

IFAC-PapersOnLine 51(8), 94–101, 2018.

DOI: <https://doi.org/10.1016/j.ifacol.2018.06.361>.

Computation of Phase Equilibrium in Reservoir Simulation and Optimization^{*}

Tobias K. S. Ritschel, John Bagterp Jørgensen

*Department of Applied Mathematics and Computer Science &
Center for Energy Resources Engineering (CERE),
Technical University of Denmark, DK-2800 Kgs. Lyngby, Denmark*

Abstract: In this paper, we discuss mathematical models and computational methods for computation of vapor-liquid equilibrium in systems relevant to reservoir simulation and optimization. We formulate the phase equilibrium problem as an optimization problem and discuss the UV-flash, the TV-flash, and the PT-flash. The UV-flash occurs for thermal and compositional dynamical simulation problems, the TV-flash occurs for compositional dynamical simulation problems, and the PT-flash occurs for steady-state problems.

© 2018, IFAC (International Federation of Automatic Control) Hosting by Elsevier Ltd. All rights reserved.

Keywords: Thermodynamics, Vapor-Liquid Equilibrium, Oil & Gas, Simulation, Optimization

1. INTRODUCTION

The operation of off-shore oil and gas fields in terms of e.g. controlling and optimizing the flow in the reservoir, the flow of oil and gas from the well to the well-head, the flow from the well-head to the riser, and the flow from the riser to the topside and separation plant all involve operation of two- or multi-phase systems. Consequently, reservoir simulation and optimization for digitalization and automation of off-shore oil and gas operations require efficient and reliable computation of phase equilibrium in vapor-liquid systems. Vapor-liquid phase equilibrium computations are also important for construction of the digital twin (a simulator) for oil and gas systems.

In this paper, we combine the second law of thermodynamics, optimization theory, and the implicit function theorem to discuss the computation of phase equilibrium and their sensitivities in vapor-liquid equilibrium systems relevant for the oil and gas industry. The sensitivities are needed when the phase equilibrium computation is embedded as part of a gradient-based method, i.e. typically a Newton-based method, for optimization, dynamic simulation, or steady-state simulation. We present and discuss the UV-flash, the TV-flash, and the PT-flash problems as well as numerically efficient ways of computing their solutions.

1.1 Literature

Several authors describe the simulation and optimization of models, related to the oil and gas industry, that incorporate phase equilibrium constraints. Li and Johns (2006) describe a method for improving the efficiency of flash calculations in compositional reservoir simulations, and Zaydullin et al. (2014) describe a fully thermal and compositional reservoir flow model. Hammer and Morin (2014) and Qiu et al. (2014) simulate two-phase pipe flow systems,

^{*} This project is funded by Innovation Fund Denmark in the OPTION project (63-2013-3) and by EUDP in the IEA project "Energy Efficient Process Control".

and Laiglecia et al. (2012) solve a dynamic optimization problem for a natural gas separation process. It is common to solve dynamic optimization problems with single-shooting algorithms. Such algorithms combine numerical simulation with numerical optimization algorithms. Efficient optimization algorithms require gradients of the objective function. Such gradients can be computed with adjoint methods (Jørgensen, 2007). Kourounis et al. (2014) use an adjoint method for solving production optimization problems for compositional reservoir models. Støren and Hertzberg (1997) describe an alternative method for efficient estimation of gradients based on approximate local thermodynamic models. Dynamic models that incorporate phase equilibrium constraints consist of 1) conservation equations (differential equations) and 2) phase equilibrium conditions (algebraic equations). Numerical algorithms for solving such differential-algebraic equations either solve the algebraic equations simultaneously with the conservation equations, or in a nested inner loop. Lima et al. (2008) use a simultaneous approach to simulate a dynamic UV-flash process, and Ritschel et al. (2017a,c,d) conclude that a simultaneous approach is faster than a nested approach for dynamic optimization of a UV-flash process. Wilhelmsen et al. (2013) find that a simultaneous approach is faster for dynamic simulation of a UV-flash process, but not for a steady-state PH-flash process.

1.2 Mathematics for the nested formulation

In the nested formulation for dynamic simulation, steady-state simulation, and optimization, the phase equilibrium is represented and solved as the optimization problem

$$y = y(x) = \arg \min_y \{f(y; x) : g(y; x) = 0\}. \quad (1)$$

In this formulation, x denotes the states, while y are variables related to the solution of the vapor-liquid equilibrium. It is important to notice that $y = y(x)$ is regarded as a function of x .

$$y_x = y_x(x) = \frac{\partial y}{\partial x}(x) \quad (2)$$

In dynamic simulation, the initial value problem (IVP),

$$x(t_0) = x_0, \quad (3a)$$

$$\dot{x}(t) = F(y(x(t))), \quad (3b)$$

may be solved numerically using the implicit Euler method. In the implicit Euler method, the states, x_{k+1} , are computed by solving the residual equations

$$R(x_{k+1}) = x_{k+1} - \Delta t F(y(x_{k+1})) - x_k = 0, \quad (4)$$

using a variant of Newton's method. Newton's method requires computation of the derivative

$$\frac{\partial}{\partial x_{k+1}} R(x_{k+1}) = I - \Delta t \frac{\partial F}{\partial y}(y(x_{k+1})) \frac{\partial y}{\partial x}(x_{k+1}). \quad (5)$$

Consequently, it is obvious that simulation by the implicit Euler method requires evaluation of $y = y(x)$ and $y_x = y_x(x)$. Similarly, computation of the steady-state, x_s , by solution of

$$\dot{x}(t) = F(y(x_s)) = 0, \quad (6)$$

using Newton's method,

$$F(y(x_s^{[k]})) + \left[\frac{\partial F}{\partial y}(y(x_s^{[k]})) \frac{\partial y}{\partial x}(x_s^{[k]}) \right] \Delta x_s^{[k]} = 0, \quad (7a)$$

$$x_s^{[k+1]} = x_s^{[k]} + \Delta x_s^{[k]}, \quad (7b)$$

requires evaluation of $y = y(x)$ and $y_x = y_x(x)$.

1.3 Mathematics for the simultaneous formulation

In the simultaneous formulation, the phase equilibrium problem,

$$\min_y f(y; x), \quad (8a)$$

$$s.t. \quad g(y; x) = 0, \quad (8b)$$

is represented by its first order optimality conditions denoted as the algebraic equations

$$G(x, y, \lambda) = 0, \quad (9)$$

where λ denotes Lagrange multipliers associated with (8). Dynamic simulation using the simultaneous formulation is conducted by numerical solution of the initial value index-1 differential algebraic equation system

$$x(t_0) = x_0, \quad (10a)$$

$$\dot{x}(t) = F(y(t)), \quad (10b)$$

$$G(x(t), y(t), \lambda(t)) = 0. \quad (10c)$$

Solution of this system using the implicit Euler method involves solution of

$$R_{k+1} = \begin{bmatrix} D_{k+1} \\ G_{k+1} \end{bmatrix} = \begin{bmatrix} x_{k+1} - \Delta t F(y_{k+1}) - x_k \\ G(x_{k+1}, y_{k+1}, \lambda_{k+1}) \end{bmatrix} = 0, \quad (11)$$

by a Newton method. Let $w = [x; y; \lambda]$. The Newton method requires computation of the derivative

$$\frac{\partial R_{k+1}}{\partial w_{k+1}} = \begin{bmatrix} I & -\Delta t \frac{\partial F}{\partial y} & 0 \\ \frac{\partial G}{\partial x} & \frac{\partial G}{\partial y} & \frac{\partial G}{\partial \lambda} \end{bmatrix}. \quad (12)$$

Similarly, the steady state, (x_s, y_s, λ_s) , is computed by solution of

$$R = \begin{bmatrix} \dot{x}(t) \\ G \end{bmatrix} = \begin{bmatrix} F(y_s) \\ G(x_s, y_s, \lambda_s) \end{bmatrix} = 0, \quad (13)$$

using a variant of Newton's method. Newton's method requires computation of the derivatives

$$\frac{\partial R}{\partial w} = \begin{bmatrix} 0 & \frac{\partial F}{\partial y} & 0 \\ \frac{\partial G}{\partial x} & \frac{\partial G}{\partial y} & \frac{\partial G}{\partial \lambda} \end{bmatrix}. \quad (14)$$

Consequently, the simultaneous method requires evaluation of the function, $G = G(x, y, \lambda)$, in the algebraic equations and computation of the derivatives, $\partial G/\partial x$, $\partial G/\partial y$, and $\partial G/\partial \lambda$.

1.4 Perspective

The formulations that we present in this paper are valid when all phases are nonempty at the solution. It is not always possible to determine the exact number of nonempty phases before solving the flash problem. This is particularly true for complex models that may require the solution of several flash problems. Consequently, the solutions to the flash problems that we describe in this paper will be incorrect if the number of specified phases is incorrect. It is difficult to incorporate the disappearance of phases because it makes the problem non-smooth. There are two approaches which can extend the formulations that we present in this paper such that the solution is correct even when some phases are empty. The first approach recasts the flash problem using non-smooth equations and solves them with non-smooth methods (Stechlinski and Barton, 2017; Watson et al., 2017; Sahlodin et al., 2016; Barton and Lee, 2002). The second approach requires the solution of a mathematical program with complementarity constraints (MPCC) (Biegler, 2010; Baumrucker et al., 2008; Raghunathan et al., 2004). Such a mathematical program belongs to the class of mathematical programs with equilibrium constraints (Outrata et al., 2013; Luo et al., 1996). Both approaches combine equations that are valid in single-phase regions with the equations that are valid in the two-phase region. Both approaches therefore extend the formulations that we present in this paper.

1.5 Paper organization

The remaining part of the paper is organized as follows. Section 2 presents methods for computation of the minimizer and its sensitivities for the unconstrained optimization problem, while Section 3 presents similar methods for the equality-constrained optimization problem. Section 4 presents the evaluation of the required thermodynamic functions using an open-source library, www.psetools.org. We discuss different flash problems, their use, and their equivalence in Section 5. Section 6 discusses the Rachford-Rice method for solution of the PT-flash. Section 7 presents numerical examples, and Section 8 contains the conclusions.

2. UNCONSTRAINED OPTIMIZATION

The unconstrained optimization problem

$$\min_y f(y; x) \quad (15)$$

has the solution denoted as

$$y = y(x) = \arg \min_y f(y; x). \quad (16)$$

In this section, we discuss the solution of this problem when $f(y; x)$ is smooth.

2.1 Optimality conditions

The first-order optimality condition for a minimizer of (15) are

$$\nabla_y f(y; x) = 0. \quad (17)$$

2.2 Newton's method and the Hessian matrix

The first-order optimality conditions (17) are solved using Newton's method. Each iteration in Newton's method involves solution of the linear system of equations

$$\nabla_{yy}^2 f(y; x) \Delta y = -\nabla_y f(y; x), \quad (18)$$

using a Cholesky factorization of the Hessian matrix, $LL' = H = \nabla_{yy}^2 f(y; x)$. This factorization also reveals if the Hessian matrix is positive definite.

2.3 Sensitivity

Application of the implicit function theorem to (17) provides the following equation

$$\nabla_{yy}^2 f(y; x) y_x = -[\nabla_{yx} f(y; x)]', \quad (19)$$

for computation of the sensitivities

$$y_x = y_x(x) = \frac{\partial y}{\partial x}(x) = [\nabla_x y(x)]'. \quad (20)$$

It is important to notice that the existing Cholesky factorization, $LL' = H = \nabla_{yy}^2 f(y; x)$, can be re-used in this computation.

2.4 Summary

The solution

$$y = y(x) = \arg \min_y f(y; x), \quad (21)$$

to the unconstrained optimization problem is a solution of the nonlinear first-order optimality conditions (17). Its sensitivities are

$$y_x = y_x(x) = -[\nabla_{yy}^2 f(y; x)]^{-1} [\nabla_{yx} f(y; x)]'. \quad (22)$$

3. EQUALITY CONSTRAINED OPTIMIZATION

We denote the solution to the equality constrained optimization problem

$$\min_y f(y; x), \quad (23a)$$

$$s.t. \quad g(y; x) = 0, \quad (23b)$$

as

$$y = y(x) = \arg \min_y \{f(y; x) : g(y; x) = 0\}. \quad (24)$$

3.1 Optimality conditions

The Lagrangian function is

$$\mathcal{L}(y, \lambda; x) = f(y; x) - \lambda' g(y; x), \quad (25)$$

where λ are Lagrange multipliers. The Karush-Kuhn-Tucker (KKT) conditions (first-order optimality conditions) for a minimizer of (23) require that the gradients of the Lagrangian with respect to y are zero and that the equality constraints are satisfied:

$$\nabla_y \mathcal{L}(y, \lambda; x) = \nabla_y f(y; x) - \nabla_y g(y; x) \lambda = 0, \quad (26a)$$

$$g(y; x) = 0. \quad (26b)$$

3.2 Newton's method and the KKT-matrix

We solve the KKT conditions (26) for both the minimizer, y , and the Lagrange multipliers, λ . We use Newton's

method. Each Newton iteration requires the solution of the linear system of equations

$$\begin{bmatrix} \nabla_{yy}^2 \mathcal{L}(y, \lambda; x) - \nabla_y g(y; x) \\ -\nabla_y g(y; x)' & 0 \end{bmatrix} \begin{bmatrix} \Delta y \\ \Delta \lambda \end{bmatrix} = - \begin{bmatrix} \nabla_y \mathcal{L}(y, \lambda; x) \\ g(y; x) \end{bmatrix}, \quad (27)$$

where the Hessian of the Lagrangian is

$$\nabla_{yy}^2 \mathcal{L}(y, \lambda; x) = \nabla_{yy}^2 f(y; x) - \sum_i \lambda_i \nabla_{yy}^2 g_i(y; x). \quad (28)$$

The Newton update is

$$\bar{y} = y + \Delta y, \quad (29a)$$

$$\bar{\lambda} = \lambda + \Delta \lambda. \quad (29b)$$

It is possible to reformulate the linear system (27) such that its solution contains $\bar{\lambda}$ instead of $\Delta \lambda$:

$$\begin{bmatrix} \nabla_{yy}^2 \mathcal{L}(y, \lambda; x) - \nabla_y g(y; x) \\ -\nabla_y g(y; x)' & 0 \end{bmatrix} \begin{bmatrix} \Delta y \\ \bar{\lambda} \end{bmatrix} = - \begin{bmatrix} \nabla_y \mathcal{L}(y, \lambda; x) \\ g(y; x) \end{bmatrix}. \quad (30)$$

The system matrix in (27) and (30) is called the KKT matrix,

$$K = \begin{bmatrix} \nabla_{yy}^2 \mathcal{L}(y, \lambda; x) - \nabla_y g(y; x) \\ -\nabla_y g(y; x)' & 0 \end{bmatrix}. \quad (31)$$

The KKT matrix is indefinite. We therefore use an LDL factorization, $LDL' = K$, when solving (30).

3.3 Sensitivity equations

The sensitivity equations result from the application of the implicit function theorem to (26):

$$\begin{bmatrix} \nabla_{yy}^2 \mathcal{L}(y, \lambda; x) - \nabla_y g(y; x) \\ -\nabla_y g(y; x)' & 0 \end{bmatrix} \begin{bmatrix} y_x \\ \lambda_x \end{bmatrix} = - \begin{bmatrix} [\nabla_{yx} f(y; x)]' \\ [\nabla_{yx} g(y; x)]' \end{bmatrix}. \quad (32)$$

The solution of (32) gives the sensitivities of both the minimizer and the Lagrange multipliers,

$$y_x = y_x(x) = \frac{\partial y}{\partial x}(x) = [\nabla_x y(x)]', \quad (33a)$$

$$\lambda_x = \lambda_x(x) = \frac{\partial \lambda}{\partial x}(x) = [\nabla_x \lambda(x)]'. \quad (33b)$$

We reuse the LDL factorization of the KKT matrix from the solution of (30).

Special case: In the case where $f(y; x) = f(y)$ and $g(y; x) = g(y) - h(x)$, the sensitivity equations become

$$\begin{bmatrix} \nabla_{yy}^2 \mathcal{L}(y, \lambda; x) - \nabla_y g(y) \\ -\nabla_y g(y)' & 0 \end{bmatrix} \begin{bmatrix} y_x \\ \lambda_x \end{bmatrix} = \begin{bmatrix} 0 \\ h_x(x) \end{bmatrix}. \quad (34)$$

3.4 Summary

The solution to the equality constrained optimization problem,

$$y = y(x) = \arg \min_y \{f(y; x) : g(y; x) = 0\}, \quad (35)$$

is a solution to the nonlinear KKT conditions (26). The sensitivities of the solution, y , and the associated Lagrange multipliers, λ , are

$$\begin{bmatrix} y_x \\ \lambda_x \end{bmatrix} = \begin{bmatrix} \nabla_{yy}^2 \mathcal{L}(y, \lambda; x) - \nabla_y g(y; x) \\ -\nabla_y g(y; x)' & 0 \end{bmatrix}^{-1} \begin{bmatrix} [\nabla_{yx} f(y; x)]' \\ [\nabla_{yx} g(y; x)]' \end{bmatrix}. \quad (36)$$

4. THERMODYNAMIC FUNCTIONS

The evaluation of thermodynamic functions is central to phase equilibrium computations. In this section, we discuss the relations between thermodynamic state functions, and we describe the relation between the fugacity coefficients and Gibbs energy. The methods described in Section 2 and Section 3 require the first and second order derivatives of thermodynamic functions. For that reason, Ritschel et al. (2017b, 2016) developed an open-source thermodynamic library, ThermoLib, which provides routines for evaluating the enthalpy, entropy, and volume as well as their first and second order derivatives with respect to temperature, pressure, and the composition vector. We use ThermoLib in this work. For more information on thermodynamic functions, we refer the reader to introductory texts on chemical engineering thermodynamics by Smith et al. (2005), Gmehling et al. (2012), Koretsky (2014), and Walas (1985).

4.1 Gibbs energy and thermodynamic functions

The temperature, T , pressure, P , and phase composition vector, n , completely specify a phase. The thermodynamic model in ThermoLib provides expressions for enthalpy, entropy, and volume:

$$H = H(T, P, n), \quad (37a)$$

$$S = S(T, P, n), \quad (37b)$$

$$V = V(T, P, n). \quad (37c)$$

We compute the remaining thermodynamic functions, internal energy, Gibbs energy, and Helmholtz energy, from the fundamental thermodynamic relations,

$$U = U(T, P, n) = H(T, P, n) - PV(T, P, n), \quad (38a)$$

$$G = G(T, P, n) = H(T, P, n) - TS(T, P, n), \quad (38b)$$

$$A = A(T, P, n) = U(T, P, n) - TS(T, P, n). \quad (38c)$$

When Gibbs energy is expressed as a function of its canonical variables, (T, P, n) , it plays the role of a *generating function*, i.e. all other thermodynamic functions can be computed from it by simple mathematical operations such as differentiation and elementary algebra. Therefore, Gibbs energy implicitly represents *complete* property information:

$$H = H(T, P, n) = G(T, P, n) - T \frac{\partial G}{\partial T}(T, P, n), \quad (39a)$$

$$S = S(T, P, n) = -\frac{\partial G}{\partial T}(T, P, n), \quad (39b)$$

$$V = V(T, P, n) = \frac{\partial G}{\partial P}(T, P, n). \quad (39c)$$

4.2 Chemical potential

We introduce the chemical potential in order to define the fugacity. The chemical potential of component i in a mixture is the partial derivative of the Gibbs energy of that mixture with respect to the i 'th mole number:

$$\mu_i(T, P, n) = \frac{\partial G}{\partial n_i}(T, P, n). \quad (40)$$

The definition of the fugacity of component i involves 1) the i 'th chemical potential of an ideal gas mixture, $\mu_i^{ig}(T, P, n) = \partial G^{ig}/\partial n_i(T, P, n)$, and 2) the chemical potential of a pure component ideal gas that only

contains n_i moles of component i , $\mu_i^{pc,ig}(T, P, n_i) = \partial G_i^{pc,ig}/\partial n_i(T, P, n_i)$. The latter is independent of the mole number, i.e. $\mu_i^{pc,ig}(T, P, n_i) = \mu_i^{pc,ig}(T, P)$. The i 'th chemical potential of an ideal gas mixture is

$$\mu_i^{ig}(T, P, n) = \frac{\partial G^{ig}}{\partial n_i}(T, P, n) = \mu_i^{pc,ig}(T, P_0) + RT \ln \frac{z_i P}{P_0}, \quad (41)$$

where P_0 is a reference pressure, and $z_i = n_i/\sum_i n_i$ is the mole fraction of component i .

4.3 Fugacity

The fugacity is implicitly defined such that the chemical potential of nonideal mixtures, $\mu_i(T, P, n)$, is given by the right-hand side of (41) where the fugacity, $f_i(T, P, n)$, replaces $z_i P$:

$$\begin{aligned} \mu_i(T, P, n) &= \frac{\partial G}{\partial n_i}(T, P, n) \\ &= \mu_i^{pc,ig}(T, P_0) + RT \ln \frac{f_i(T, P, n)}{P_0}. \end{aligned} \quad (42)$$

We subtract (41) from (42):

$$RT \ln \frac{f_i(T, P, n)}{z_i P} = \frac{\partial}{\partial n_i} (G(T, P, n) - G^{ig}(T, P, n)). \quad (43)$$

The difference between the actual Gibbs energy of a mixture and the Gibbs energy of that mixture if it was an ideal gas is defined as the residual Gibbs energy, $G^R(T, P, n) = G(T, P, n) - G^{ig}(T, P, n)$. We also introduce the fugacity coefficients, $\phi_i(T, P, n) = f_i(T, P, n)/(z_i P)$. With these two definitions, we obtain the following expression for the logarithm of the fugacity coefficients:

$$\ln \phi_i(T, P, n) = \frac{1}{RT} \frac{\partial G^R}{\partial n_i}(T, P, n). \quad (44)$$

The fugacity coefficients play an important role in isothermal and isobaric (constant temperature and pressure) vapor-liquid equilibrium problems as we discuss in Section 6. The logarithm of the fugacity coefficients of ideal gas mixtures are by definition zero, i.e. $\ln \phi_i^{ig}(T, P, n^v) = 0$. For ideal liquid mixtures, the logarithm of the fugacity coefficients are (Ritschel and Jørgensen, 2017)

$$\ln \phi_i^{id}(T, P, n^l) = \frac{P_i^{sat}(T)}{P} \exp\left(\frac{v_i^l(T)(P - P_i^{sat}(T))}{RT}\right). \quad (45)$$

$v_i^l(T)$ is the liquid volume, and $P_i^{sat}(T)$ is the saturation pressure. We see that the ideal liquid fugacity coefficients are independent of composition, i.e. $\ln \phi_i^{id}(T, P, n^l) = \ln \phi_i^{id}(T, P)$. ThermoLib uses cubic equations of state for nonideal mixtures. It is outside the scope of this work to describe the corresponding fugacity coefficients. However, both Ritschel et al. (2016) and Ritschel and Jørgensen (2017) provide expressions for them.

5. FLASH PROBLEMS

Table 1 shows the specified variables, the state function that is minimal at equilibrium, and relevant types of models for different flash problems (Paterson, 2017). We demonstrate that the UV-, the TV-, and the PT-flash problems can be formulated as unconstrained (15) and

Table 1. Examples of flash problems

Spec.	State function	Example of relevant model
(P, T)	G	Isothermal-isobaric steady-state flow
(T, V)	A	Isothermal unsteady-state flow
(P, H)	$-S$	Thermal steady-state flow
(U, V)	$-S$	Thermal unsteady-state flow
(P, S)	H	Reversible expansion/compression

constrained (23) optimization problems (Michelsen, 1999; Michelsen and Mollerup, 2007). We use a single-stage dynamic model to illustrate the use of the flash problems in dynamic models. Typically, more complex models can be formulated as interconnected single-stage models. The PT-flash is of particular interest because 1) it can be formulated as unconstrained optimization, and 2) other types of flash problems are equivalent to a combination of algebraic constraints and the PT-flash. The PT-flash is therefore often used in unsteady-state computations using a nested approach. We discuss both of these aspects.

5.1 UV-flash problem

The UV-flash problem is considered difficult to solve (Saha and Carroll, 1997; Castier, 2009). However, it is a key component in rigorous models of vapor-liquid equilibrium processes such as fluid vessels (Castier, 2010), distillation columns (Flatby et al., 1994), and thermal- and compositional oil recovery. We consider a single-stage model, essentially a fluid vessel, that contains a vapor-liquid mixture. A feed stream (f) supplies mass, and therefore energy, to the mixture. A vapor stream (v) and a liquid stream (l) extract mass and energy from the mixture. Furthermore, a heat input, Q , supplies energy to the mixture. The mass and energy conservation equations are

$$\dot{n}_i = f_i - l_i - v_i, \quad (46a)$$

$$\dot{U} = H_f - H_l - H_v + Q, \quad (46b)$$

where n_i is the moles of component i , and U is the internal energy of the mixture. The mixture temperature, T , pressure, P , and vapor-liquid composition, n^v and n^l , completely specify the vapor-liquid streams, i.e.

$$l_i = l_i(T, P, n^l), \quad (47a)$$

$$v_i = v_i(T, P, n^v), \quad (47b)$$

and

$$H_l = H_l(T, P, n^l), \quad (48a)$$

$$H_v = H_v(T, P, n^v). \quad (48b)$$

We assume that the vapor phase and the liquid phase are in equilibrium at all times. The conservation equations (46) specify the internal energy, U , and the total moles of each chemical component, n . The volume of the tank, V , is fixed. U , V , and n completely specify the temperature, T , pressure, P , and phase compositions of the vapor-liquid mixture, n^v and n^l . That is because the second law of thermodynamics states that the entropy, S , of a closed system is maximal at equilibrium, or equivalently that $-S$ is minimal:

$$\min_{T, P, n^v, n^l} -S = -(S^v(T, P, n^v) + S^l(T, P, n^l)), \quad (49a)$$

$$s.t. \quad U^v(T, P, n^v) + U^l(T, P, n^l) = U, \quad (49b)$$

$$V^v(T, P, n^v) + V^l(T, P, n^l) = V, \quad (49c)$$

$$n_i^v + n_i^l = n_i, \quad i = 1, \dots, N_C. \quad (49d)$$

We use the linear mass balance constraint (49d) to eliminate the liquid mole numbers, $n^l = n - n^v$:

$$\min_{T, P, n^v} -S = -(S^v(T, P, n^v) + S^l(T, P, n - n^v)), \quad (50a)$$

$$s.t. \quad U^v(T, P, n^v) + U^l(T, P, n - n^v) = U, \quad (50b)$$

$$V^v(T, P, n^v) + V^l(T, P, n - n^v) = V. \quad (50c)$$

5.2 TV-flash problem

We again consider the single-stage model (46). We assume that the internal energy is constant:

$$\dot{n}_i = f_i - l_i - v_i, \quad (51a)$$

$$\dot{U} = H_f - H_l - H_v + Q = 0. \quad (51b)$$

The dynamic equations therefore only consist of the mass conservation equations:

$$\dot{n}_i = f_i - l_i - v_i. \quad (52)$$

The condition of constant internal energy requires that the heat input precisely matches the difference between the enthalpies of the feed and the vapor-liquid streams:

$$Q = H_l + H_v - H_f = \Delta H. \quad (53)$$

When the temperature is constant, the condition of maximal entropy is equivalent to a condition of minimal Helmholtz energy, A :

$$\min_{P, n^v, n^l} A = A^v(T, P, n^v) + A^l(T, P, n^l), \quad (54a)$$

$$s.t. \quad V^v(T, P, n^v) + V^l(T, P, n^l) = V, \quad (54b)$$

$$n_i^v + n_i^l = n_i, \quad i = 1, \dots, N_C. \quad (54c)$$

T , V , and n therefore completely specify P , n^v , and n^l . Again, we eliminate the liquid mole numbers:

$$\min_{P, n^v} A = A^v(T, P, n^v) + A^l(T, P, n - n^v), \quad (55a)$$

$$s.t. \quad V^v(T, P, n^v) + V^l(T, P, n - n^v) = V. \quad (55b)$$

5.3 PT-flash problem

We consider the single-stage model (46) again. Isothermal and isobaric processes (constant temperature and pressure) are relevant to steady state processes:

$$\dot{n}_i = f_i - l_i - v_i = 0, \quad (56a)$$

$$\dot{U} = H_f - H_l - H_v + Q = 0. \quad (56b)$$

The heat input exactly matches the difference in enthalpy of the feed and the vapor-liquid streams,

$$Q = H_l + H_v - H_f = \Delta H, \quad (57)$$

and the molar flow rates of the vapor-liquid streams exactly match those of the feed stream:

$$l_i + v_i = f_i. \quad (58)$$

The condition of maximal entropy is equivalent to a condition of minimal Gibbs energy when both temperature and pressure are constant:

$$\min_{n^v, n^l} G = G^v(T, P, n^v) + G^l(T, P, n^l), \quad (59a)$$

$$s.t. \quad n_i^v + n_i^l = n_i, \quad i = 1, \dots, N_C. \quad (59b)$$

T , P , and n thus completely specify n^v and n^l . We eliminate the liquid mole numbers and obtain an unconstrained minimization problem:

$$\min_{n^v} G = G^v(T, P, n^v) + G^l(T, P, n - n^v). \quad (60)$$

5.4 Equivalence of different flash problems

The UV-flash problem (50) is equivalent to a combination of the PT-flash (60) and constraints on U and V as in (50b)-(50c). The UV constraints are thereby effectively moved outside of the optimization problem at the cost of changing the objective function. The VT flash problem (55) is also equivalent to the PT-flash (60) combined with a constraint on V as in (55b). That means that it is possible to solve UV- and TV-flash problems with unconstrained optimization methods. It also means that existing software for PT-flash problems can be reused when solving the more complex flash problems.

6. THE RACHFORD-RICE EQUATIONS

The Rachford-Rice equations are often used to solve PT-flash problems. We outline the associated computation and the computations of the needed sensitivities.

6.1 Vapor-liquid equilibrium constant

The PT-flash problem can be formulated as the unconstrained optimization problem

$$\min_{n^v} G = G^v(T, P, n^v) + G^l(T, P, n - n^v), \quad (61)$$

for which the first-order optimality conditions are

$$\frac{\partial G}{\partial n_i^v} = \frac{\partial G^v}{\partial n_i^v}(T, P, n^v) - \frac{\partial G^l}{\partial n_i^l}(T, P, n^l) = 0, \quad (62)$$

for all i and with the liquid phase mole numbers being $n^l = n - n^v$. The chemical potentials of the vapor and liquid phases are defined as

$$\mu_i^v(T, P, n^v) = \frac{\partial G^v}{\partial n_i^v}(T, P, n^v), \quad (63a)$$

$$\mu_i^l(T, P, n^l) = \frac{\partial G^l}{\partial n_i^l}(T, P, n^l). \quad (63b)$$

The equilibrium conditions (62) are therefore

$$\mu_i^v(T, P, n^v) = \mu_i^l(T, P, n^l). \quad (64)$$

Because of the definition of the fugacities (42), the condition of equal chemical potentials (64) is equivalent to

$$f_i^v(T, P, n^v) = f_i^l(T, P, n^l). \quad (65)$$

Let $x_i = n_i^l / \sum_i n_i^l$ and $y_i = n_i^v / \sum_i n_i^v$ be the mole fractions of the liquid and vapor phase, respectively. Because of the definition of the fugacity coefficients, the equality of fugacities (65) is equivalent to $\phi_i^v(T, P, n^v) y_i P = \phi_i^l(T, P, n^l) x_i P$ or equivalently

$$\phi_i^v(T, P, n^v) y_i = \phi_i^l(T, P, n^l) x_i. \quad (66)$$

The equilibrium constants (sometimes referred to as equilibrium ratios) are defined as $K_i = y_i / x_i$. We use (66) to derive an expression for the equilibrium constants:

$$K_i(T, P, n^v, n^l) = \frac{y_i}{x_i} = \frac{\phi_i^l(T, P, n^l)}{\phi_i^v(T, P, n^v)}. \quad (67)$$

6.2 The Rachford-Rice method - Ideal vapor-liquid mixture

For mixtures of an ideal gas and an ideal liquid, the vapor liquid equilibrium constant defined by (67) is

$$K_i = K_i(T, P) = \frac{P_i^{sat}(T)}{P} \exp\left(\frac{v_i^l(T)(P - P_i^{sat}(T))}{RT}\right). \quad (68)$$

In that case, the vapor-liquid equilibrium constant is independent of composition and depends only on temperature, T , and pressure, P , but not on the compositions of the liquid and vapor mixture. Define $z_i = n_i / \sum_i n_i$ such that $n_i^l + n_i^v = n_i$ can be expressed as

$$(1 - \beta)x_i + \beta y_i = z_i, \quad (69)$$

where β is the vapor fraction defined as

$$\beta = \frac{\sum_i n_i^v}{\sum_i n_i}. \quad (70)$$

The relation (69) and $K_i = y_i / x_i$ imply that

$$\begin{aligned} z_i &= (1 - \beta)x_i + \beta y_i \\ &= (1 - \beta)x_i + \beta K_i x_i = (1 + \beta(K_i - 1))x_i, \end{aligned} \quad (71)$$

such that

$$x_i = \frac{1}{1 + \beta(K_i - 1)} z_i, \quad (72a)$$

$$y_i = K_i x_i = \frac{K_i}{1 + \beta(K_i - 1)} z_i. \quad (72b)$$

This implies that

$$n_i^l = x_i(1 - \beta) \sum_i n_i = \frac{1 - \beta}{1 + \beta(K_i - 1)} n_i, \quad (73a)$$

$$n_i^v = y_i \beta \sum_i n_i = \frac{\beta K_i}{1 + \beta(K_i - 1)} n_i. \quad (73b)$$

The relations $\sum_i x_i = \sum_i y_i = 1$ imply that $\sum_i (y_i - x_i) = 0$. By combination of this observation and (72), we obtain the following relation

$$\begin{aligned} f(\beta) &= \sum_i (y_i - x_i) = \sum_i (K_i - 1) x_i \\ &= \sum_i \frac{K_i - 1}{1 + \beta(K_i - 1)} z_i = 0, \end{aligned} \quad (74)$$

for computation of β . The derivative of this function is

$$f'(\beta) = - \sum_i \left[\frac{K_i - 1}{1 + \beta(K_i - 1)} \right]^2 z_i \leq 0. \quad (75)$$

Newton's method for determination of β is

$$\beta_{k+1} = \beta_k - \frac{f(\beta_k)}{f'(\beta_k)}. \quad (76)$$

When β has been computed, we compute the vapor composition, n_i^v , from (73b) and the liquid composition from $n_i^l = n_i - n_i^v$.

Sensitivity: The vapor fraction is a function of temperature, pressure, and total composition, i.e. $\beta = \beta(K, n) = \beta(K(T, P), n) = \beta(T, P, n)$. The sensitivities of β are

$$\frac{\partial \beta}{\partial T} = \sum_i \frac{\partial \beta}{\partial K_i} \frac{\partial K_i}{\partial T}, \quad (77a)$$

$$\frac{\partial \beta}{\partial P} = \sum_i \frac{\partial \beta}{\partial K_i} \frac{\partial K_i}{\partial P}, \quad (77b)$$

$$\frac{\partial \beta}{\partial n_j} = \sum_i \frac{\partial \beta}{\partial z_i} \frac{\partial z_i}{\partial n_j}, \quad (77c)$$

where

$$\frac{\partial \beta}{\partial K_i} = - \frac{\partial f}{\partial K_i} / f'(\beta), \quad \frac{\partial \beta}{\partial z_i} = - \frac{\partial f}{\partial z_i} / f'(\beta). \quad (78)$$

Similarly, the vapor-liquid mole numbers, n_i^v and n_i^l , are functions of temperature, pressure, and total composition,

i.e. $n^v = n^v(\beta, K, n) = n^v(\beta(T, P, n), K(T, P), n) = n^v(T, P, n)$ and $n^l = n^l(n^v, n) = n^l(n^v(T, P, n), n) = n^l(T, P, n)$. In order to describe the sensitivities of n^l and n^v , we introduce the auxiliary variables

$$\tau_i = \frac{1}{1 + \beta(K_i - 1)}, \quad (79a)$$

$$\gamma_i = K_i n_i - (K_i - 1)n_i^v. \quad (79b)$$

The sensitivities of the vapor mole numbers are

$$\frac{\partial n_i^v}{\partial T} = \tau_i \left(\frac{\partial \beta}{\partial T} \gamma_i + \beta \frac{\partial K_i}{\partial T} (n_i - n_i^v) \right), \quad (80a)$$

$$\frac{\partial n_i^v}{\partial P} = \tau_i \left(\frac{\partial \beta}{\partial P} \gamma_i + \beta \frac{\partial K_i}{\partial P} (n_i - n_i^v) \right), \quad (80b)$$

$$\frac{\partial n_i^v}{\partial n_j} = \tau_i \left(\frac{\partial \beta}{\partial n_j} \gamma_i + \beta K_i \delta_{ij} \right), \quad (80c)$$

and the sensitivities of the liquid mole numbers are

$$\frac{\partial n_i^l}{\partial T} = -\frac{\partial n_i^v}{\partial T}, \quad (81a)$$

$$\frac{\partial n_i^l}{\partial P} = -\frac{\partial n_i^v}{\partial P} \quad (81b)$$

$$\frac{\partial n_i^l}{\partial n_j} = \delta_{ij} - \frac{\partial n_i^v}{\partial n_j}. \quad (81c)$$

δ_{ij} is Kronecker's delta, i.e. $\delta_{ij} = 1$ if $i = j$ and 0 if $i \neq j$.

7. NUMERICAL EXAMPLE

In this section, we solve the UV-, TV-, and PT-flash problems for different values of the specified variables. We consider a mixture of 60% C₁, 8% C₂, 5% C₃, 25% n-C₇, and 2% CO₂. Figure 1 shows the solutions to the flash problems (in blue and red) in a phase diagram and as functions of the specified variables. The vapor fraction is constant along the black and gray curves. The curves meet at the critical point which is located around 425 K and 16.3 MPa. The upper and the lower black curves are the bubble-point ($\beta = 0$) and the dew-point ($\beta = 1$) curves, which together constitute the phase envelope. Ritschel and Jørgensen (2017) describe the computation of the isoparametric (constant vapor fraction) curves. The vapor fraction of the curves increases in steps of 0.1 starting from the bubble-point curve and ending in the dew-point curve. The vapor fraction depends nonlinearly on the specified variables. The pressure exhibits an inverse relationship with volume. The remaining variables are close to linear in the specified variables for this example.

8. CONCLUSION

We describe how the solution to the phase equilibrium problem enters into dynamic and steady-state simulations for both nested and simultaneous formulations. We formulate the phase equilibrium problems as optimization problems and present methods for their solution and computation of the needed sensitivities. In particular, we describe the UV-, and the TV-flash problems which are relevant to thermal and isothermal compositional reservoir flow models, respectively. We also describe the commonly used PT-flash and its relation to the other types of flash problems. Finally, using a numerical example, we illustrate how the solutions to the UV-, TV-, and PT-flash problems depend on the specified variables.

REFERENCES

- Barton, P.I. and Lee, C.K. (2002). Modeling, simulation, sensitivity analysis, and optimization of hybrid systems. *ACM Transactions on Modeling and Computer Simulation*, 256–289.
- Baumrucker, B., Renfro, J., and Biegler, L.T. (2008). MPEC problem formulations and solution strategies with chemical engineering applications. *Computers & Chemical Engineering*, 32(12), 2903–2913.
- Biegler, L.T. (2010). *Nonlinear programming: Concepts, Algorithms, and Applications to Chemical Processes*. SIAM.
- Castier, M. (2009). Solution of the isochoric-isoeenergetic flash problem by direct entropy maximization. *Fluid Phase Equilibria*, 276(1), 7–17.
- Castier, M. (2010). Dynamic simulation of fluids in vessels via entropy maximization. *Journal of Industrial and Engineering Chemistry*, 16(1), 122–129.
- Flatby, P., Skogestad, S., and Lundström, P. (1994). Rigorous dynamic simulation of distillation columns based on UV-flash. In *IFAC Symposium on Advanced Control of Chemical Processes (ADCHEM '94)*, 261–266.
- Gmehling, J., Kolbe, B., Kleiber, M., and Rarey, J. (2012). *Chemical Thermodynamics for Process Simulation*. Wiley-VCH.
- Hammer, M. and Morin, A. (2014). A method for simulating two-phase pipe flow with real equations of state. *Computers & Fluids*, 100, 45–58.
- Jørgensen, J.B. (2007). Adjoint sensitivity results for predictive control, state- and parameter-estimation with nonlinear models. In *European Control Conference (ECC), 2007*, 3649–3656. IEEE.
- Koretsky, M.D. (2014). *Engineering and Chemical Thermodynamics*. Wiley, 2nd edition.
- Kourounis, D., Durlinsky, L.J., Jansen, J.D., and Aziz, K. (2014). Adjoint formulation and constraint handling for gradient-based optimization of compositional reservoir flow. *Computational Geosciences*, 18(2), 117–137.
- Laiglecia, J.I., Lopez-Negrete, R., Diaz, M.S., and Biegler, L.T. (2012). A simultaneous dynamic optimization approach for natural gas processing plants. In *Proceedings of Foundations of Computer Aided Process Operations (FOCAPO)*.
- Li, Y. and Johns, R.T. (2006). Rapid flash calculations for compositional simulation. *SPE Reservoir Evaluation & Engineering*, SPE 95732, 521–529.
- Lima, E.R.A., Castier, M., and Biscaia Jr., E.C. (2008). Differential-algebraic approach to dynamic simulations of flash drums with rigorous evaluation of physical properties. *Oil & Gas Science and Technology-Revue de l'IFP*, 63(5), 677–686.
- Luo, Z.Q., Pang, J.S., and Ralph, D. (1996). *Mathematical programs with equilibrium constraints*. Cambridge University Press.
- Michelsen, M.L. and Mollerup, J.M. (2007). *Thermodynamic Models: Fundamentals and Computational Aspects*. Tie-Line Publications.
- Michelsen, M.L. (1999). State function based flash specifications. *Fluid Phase Equilibria*, 158, 617–626.
- Outrata, J., Kocvara, M., and Zowe, J. (2013). *Nonsmooth Approach to Optimization Problems with Equilibrium Constraints: Theory, Applications and Numerical results*, volume 28. Springer Science & Business Media.

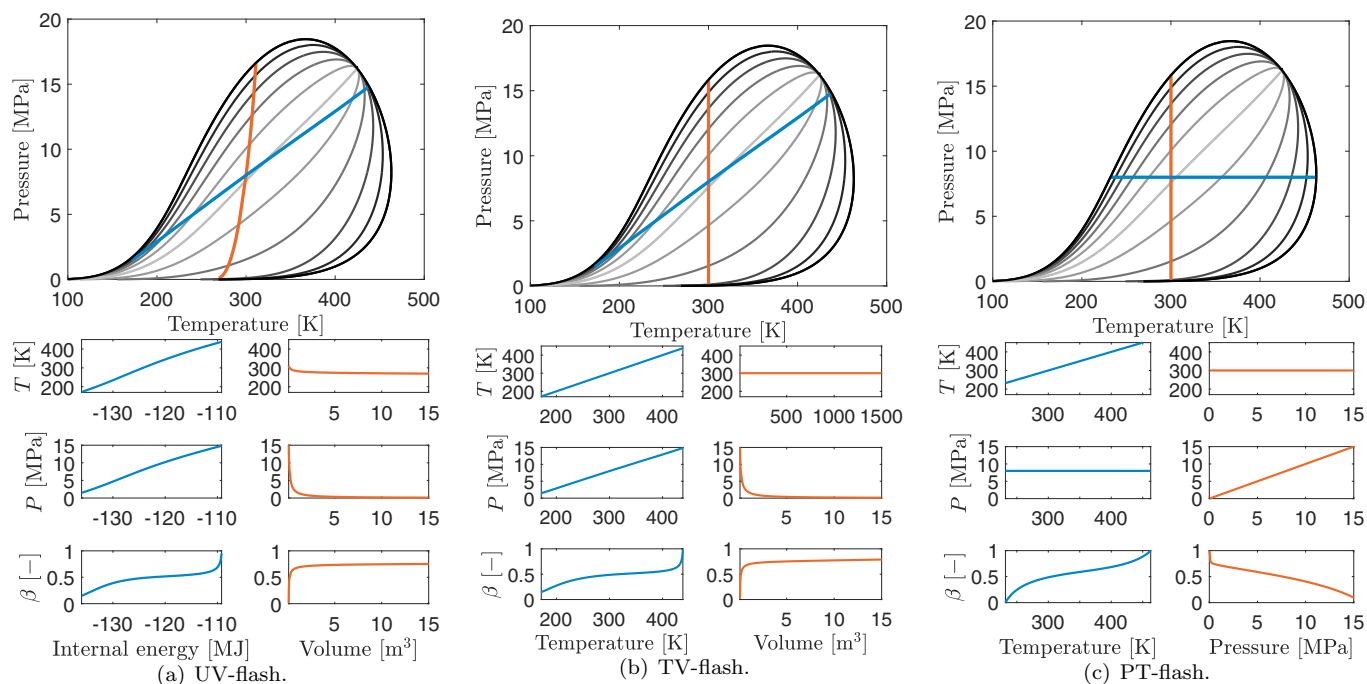


Fig. 1. Solutions to UV-, TV-, and PT-flash problems for a mixture of C_1 - C_3 , C_7 , and CO_2 .

- Paterson, D. (2017). *Flash Computation and EoS Modelling for Compositional Thermal Simulation of Flow in Porous Media*. Ph.D. thesis, Technical University of Denmark.
- Qiu, L., Wang, Y., and Reitz, R.D. (2014). Multiphase dynamic flash simulations using entropy maximization and application to compressible flow with phase change. *AIChE Journal*, 60(8), 3013–3024.
- Raghunathan, A.U., Diaz, M.S., and Biegler, L.T. (2004). An MPEC formulation for dynamic optimization of distillation operations. *Computers & chemical engineering*, 28(10), 2037–2052.
- Ritschel, T.K.S., Capolei, A., Gaspar, J., and Jørgensen, J.B. (2017a). An algorithm for gradient-based dynamic optimization of UV-flash processes. *Computers and Chemical Engineering*. Accepted.
- Ritschel, T.K.S., Gaspar, J., and Jørgensen, J.B. (2017b). A thermodynamic library for simulation and optimization of dynamic processes. In *Proceedings of the 20th World Congress of the International Federation of Automatic Control*, 3542–3547.
- Ritschel, T.K.S., Capolei, A., and Jørgensen, J.B. (2017c). The adjoint method for gradient-based optimization of UV flash processes. In A. Espuna, M. Graells, and L. Puigjaner (eds.), *27th European Symposium on Computer Aided Process Engineering - ESCAPE 27*. Barcelona, Spain.
- Ritschel, T.K.S., Capolei, A., and Jørgensen, J.B. (2017d). Dynamic optimization of UV flash processes. In *FOCAPO / CPC 2017*. Tucson, Arizona.
- Ritschel, T.K.S., Gaspar, J., Capolei, A., and Jørgensen, J.B. (2016). An open-source thermodynamic software library. Technical Report DTU Compute Technical Report-2016-12, Department of Applied Mathematics and Computer Science, Technical University of Denmark.
- Ritschel, T.K.S. and Jørgensen, J.B. (2017). Computation of phase equilibrium and phase envelopes. Technical Report DTU Compute Technical Report-2017-11, Department of Applied Mathematics and Computer Science, Technical University of Denmark.
- Saha, S. and Carroll, J.J. (1997). The isoenergetic-isochoric flash. *Fluid phase equilibria*, 138(1), 23–41.
- Sahlodin, A.M., Watson, H.A.J., and Barton, P.I. (2016). Nonsmooth model for dynamic simulation of phase changes. *AIChE Journal*, 62(9), 3334–3351.
- Smith, J.M., Van Ness, H.C., and Abbott, M.M. (2005). *Introduction to Chemical Engineering Thermodynamics*. McGraw-Hill, New York, NY, 7th edition.
- Stechlinski, P.G. and Barton, P.I. (2017). Dependence of solution of nonsmooth differential-algebraic equations on parameters. *Journal of Differential Equations*, 262, 2254–2285.
- Støren, S. and Hertzberg, T. (1997). Local thermodynamic models used in sensitivity estimation of dynamic systems. *Computers & Chemical Engineering*, 21, S709–S714.
- Walas, S.M. (1985). *Phase Equilibria in Chemical Engineering*. Butterworth-Heinemann.
- Watson, H.A.J., Vikse, M., Gundersen, T., and Barton, P.I. (2017). Reliable flash calculations: Part 1. nonsmooth inside-out algorithms. *Industrial & Engineering Chemistry Research*, 56(4), 960–973.
- Wilhelmsen, Ø., Skaugen, G., Hammer, M., Wahl, P.E., and Morud, J.C. (2013). Time efficient solution of phase equilibria in dynamic and distributed systems with differential algebraic equation solvers. *Industrial & Engineering Chemistry Research*, 52(5), 2130–2140.
- Zaydullin, R., Voskov, D.V., James, S.C., Henley, H., and Lucia, A. (2014). Fully compositional and thermal reservoir simulation. *Computers & Chemical Engineering*, 63, 51–65.

Paper VII

Production Optimization of a Rigorous Thermal and Compositional Reservoir Flow Model

Authors:

Tobias K. S. Ritschel and John Bagterp Jørgensen

Published in:

IFAC-PapersOnLine 51(8), 76–81, 2018.

DOI: <https://doi.org/10.1016/j.ifacol.2018.06.358>.

Production Optimization of a Rigorous Thermal and Compositional Reservoir Flow Model^{*}

Tobias K. S. Ritschel, John Bagterp Jørgensen

*Department of Applied Mathematics and Computer Science &
Center for Energy Resources Engineering (CERE),
Technical University of Denmark, DK-2800 Kgs. Lyngby, Denmark*

Abstract: We model thermal and compositional reservoir production as mass and energy balances combined with a phase equilibrium constraint. The phase equilibrium constraint is modeled as a thermodynamically rigorous UV flash process. The UV flash problem is a mathematical statement of the second law of thermodynamics, and it replaces the condition of equality of fugacities that is often used. We demonstrate that such a thermal and compositional reservoir model is in a semi-explicit index-1 differential-algebraic form, and we briefly describe a gradient-based single-shooting algorithm for the solution of production optimization problems. We implement the algorithm in C/C++ using the software DUNE, the thermodynamic software ThermoLib, and the optimization software KNITRO. We present an example of optimal water flooding where the injected water has a higher temperature than the reservoir fluid.

© 2018, IFAC (International Federation of Automatic Control) Hosting by Elsevier Ltd. All rights reserved.

Keywords: Thermal and compositional model, Phase equilibrium, Optimal control

1. INTRODUCTION

Production optimization is concerned with maximizing a financial measure over the expected lifetime of an oil reservoir. It combines numerical simulation of the subsurface reservoir flow with numerical optimization. Production optimization is applicable to both traditional recovery methods, such as waterflooding, and to enhanced oil recovery methods such as chemical, biological, and thermal methods. The simulation and optimization of enhanced oil recovery processes often require compositional flow models. In particular, thermal recovery processes require thermal and compositional models.

Thermal and compositional reservoir flow models combine two main principles; 1) conservation of mass and energy, and 2) phase equilibrium. The phase equilibrium condition is based on a thermodynamic state function being minimal or maximal (Michelsen, 1999). The phase equilibrium condition is therefore formulated as a mathematical optimization problem, i.e. it is an optimization problem within the production optimization problem. The optimization problem that is relevant to thermal and compositional models is called the UV n flash problem because the internal energy, U , the volume, V , and the total amount of moles of each chemical species, n , are specified as parameters in the problem. The UV n flash is a mathematical formulation of the second law of thermodynamics which states that the entropy, S , of a closed system in equilibrium is maximal. This problem is often just called the UV flash because the total amount of moles are specified in all flash problems. It is also called the isochoric-isoenergetic flash. The solution to the UV flash problem is the equilibrium temperature,

pressure, and the phase compositions (in moles). The most commonly known flash problem is the PT flash problem where both temperature and pressure are specified and Gibbs energy is minimized. This is the problem that is most often encountered in the reservoir simulation and optimization literature. However, there it is formulated as the equality of fugacities and not as an optimization problem (Zaydullin et al., 2014; Kourounis et al., 2014). The condition of equal fugacities can be derived from the first-order optimality conditions of the PT flash problem. The PT flash is common because it can be solved efficiently with unconstrained optimization methods and because it is equivalent to other types of flash problems when it is combined with algebraic constraints on the specified quantities, e.g. the UV flash is equivalent to the combination of constraints on the internal energy, U , and the volume, V , and the PT flash. The UV flash is a key component in rigorous modeling of several vapor-liquid equilibrium processes such as fluid vessels and flash drums (Arendsen and Versteeg, 2009; Castier, 2010; Lima et al., 2008), distillation columns (Flatby et al., 1994), and two-phase computational fluid dynamical problems (Qiu et al., 2014; Hammer and Morin, 2014). Recently, a gradient-based single-shooting algorithm for the dynamic optimization of UV flash processes was developed by Ritschel et al. (2017a). The algorithm uses an adjoint method to compute gradients.

Most research in production optimization algorithms considers simple two-phase flow models that contain a water component and an oil pseudo-component. Some authors have solved production optimization problems involving polymer flooding (Lei et al., 2012). Such problems do not involve phase equilibrium conditions. More recently Kourounis et al. (2014) applied a gradient-based single-

^{*} This project is funded by Innovation Fund Denmark in the OPTION project (63-2013-3).

shooting algorithm to a compositional model. Furthermore, Zaydullin et al. (2014) described a framework for fully thermal and compositional reservoir simulation. It is common to use single-shooting algorithms to solve production optimization problems (Bukshytynov et al., 2015; Kourounis et al., 2014; Forouzanfar et al., 2013; Capolei et al., 2012). Some authors have also used multiple-shooting (Cudas et al., 2017; Capolei and Jørgensen, 2012) and simultaneous collocation (Heirung et al., 2011). These three methods belong to the class of gradient-based methods. There are two other classes of algorithms that authors use to solve production optimization problems; 1) gradient-free methods (Zhao et al., 2016) and 2) artificial intelligence methods (Onwunalu and Durlofsky, 2010; Saputelli et al., 2002). The advantage of gradient-based methods is that they are computationally efficient. The disadvantages of gradient-based methods are 1) that they require the computation of gradients which can be cumbersome for complex models, and 2) that they converge to local minima.

The novelty of this work is the rigorous modeling of thermal and compositional production optimization as a UV flash process. We describe the UV flash problem and the mass and energy conservation equations of the model. We implement the single-shooting algorithm described by Ritschel et al. (2017a) in C/C++ and provide an example of an optimal water injection strategy where the injected water has a higher temperature than the reservoir fluid.

Section 2 describes the thermal and compositional model, and Section 3 briefly describes the single-shooting algorithm. Section 4 provides a few details on the implementation, and Section 5 presents the numerical example. Section 6 presents conclusions.

2. RESERVOIR FLOW MODEL

In this section, we describe the thermal and compositional reservoir flow model and demonstrate that it is in the semi-explicit index-1 differential-algebraic form that is considered by Ritschel et al. (2017a). The model consists of a set of phase equilibrium conditions based on the second law of thermodynamics, a set of mass balance equations, and one energy balance equation. The phase equilibrium is reached at a much faster timescale than the flow processes and we therefore assume that the phases are in equilibrium at all times. The mass and energy conservation equations are based on models of advective fluid flow and thermal conduction in the rock.

2.1 Phase equilibrium

The fluid consists of water (w), oil (o), and gas (g). The oil and gas phases contain N_C chemical components. We assume that all fluid phases are in thermal, mechanical and chemical equilibrium with each other. We furthermore assume that the fluid is in thermal and mechanical equilibrium with the rock (r), i.e. $T = T^\alpha = T^r$ and $P = P^\alpha = P^r$ for $\alpha \in \{w, o, g\}$. We comment further on the thermal equilibrium between the fluid and the rock in Section 2.3. The phase equilibrium is governed by the UV flash in which the internal energy, U , the volume V , and the total amount of moles of each chemical species, n_w

and $n = [n_1; \dots; n_{N_C}]$, are specified as parameters. The solution to the UV flash problem is the temperature, T , pressure, P , and phase composition vectors, n^w , n^o , and n^g , that maximize entropy under the above constraints:

$$\max_{T, P, n^w, n^o, n^g} S^w + S^o + S^g + S^r, \quad (1a)$$

$$\text{subject to } U^w + U^o + U^g + U^r = U, \quad (1b)$$

$$V^w + V^o + V^g + V^r = V, \quad (1c)$$

$$n^w = n_w, \quad (1d)$$

$$n_k^o + n_k^g = n_k, \quad k = 1, \dots, N_C. \quad (1e)$$

The above optimization problem only contains equality constraints. The necessary first-order optimality conditions are therefore algebraic equations which are solved for each grid cell, simultaneously with the conservation equations, during simulation.

2.2 Mass conservation equations

The fluid flow process is advective. Each conservation equation contains a molar flux term and an injection/production source term:

$$\partial_t C_w = -\nabla \cdot \mathbf{N}^w + Q^w, \quad (2a)$$

$$\partial_t C_k = -\nabla \cdot \mathbf{N}_k + Q_k, \quad k = 1, \dots, N_C. \quad (2b)$$

The molar component flux, \mathbf{N}_k , is

$$\mathbf{N}_k = x_k \mathbf{N}^o + y_k \mathbf{N}^g. \quad (3)$$

\mathbf{N}^α is the molar flux of phase $\alpha \in \{w, o, g\}$. x_k and y_k are oil and gas mole fractions. The molar injection/production terms are

$$Q^w = Q^{w, \text{inj}} - Q^{w, \text{prod}}, \quad (4a)$$

$$Q_k = -(x_k Q^{o, \text{prod}} + y_k Q^{g, \text{prod}}). \quad (4b)$$

Water is injected at a molar rate of $Q^{w, \text{inj}}$ while all fluid phases are produced at molar rates of $Q^{\alpha, \text{prod}}$.

2.3 Energy conservation equations

We first describe the energy conservation equations without assuming that the fluid and the rock are in thermal equilibrium. Then we describe how this assumption affects the model equations and present the model that assumes thermal equilibrium. The fluid (f) and rock (r) energy conservation equations are

$$\partial_t u^f = -\nabla \cdot \mathbf{N}_u^f + Q_u^f, \quad (5a)$$

$$\partial_t u^r = -\nabla \cdot \mathbf{N}_u^r + Q_u^r. \quad (5b)$$

u^f and u^r are internal energies per unit volume. The fluid heat flux is caused by advection of the phases:

$$\mathbf{N}_u^f = h^w \mathbf{N}^w + h^o \mathbf{N}^o + h^g \mathbf{N}^g. \quad (6)$$

h^α is the molar enthalpy of phase α . We model the conductive rock heat flux with Fourier's law of thermal conduction (Holman, 2010, Chap. 1):

$$\mathbf{N}_u^r = -k_T^r \nabla T^r. \quad (7)$$

k_T^r is the thermal conductivity of the rock. Both the wells and the conduction at the rock-fluid interface affect the fluid energy balance:

$$Q_u^f = h^{w, \text{inj}} Q^{w, \text{inj}} - \sum_{\alpha \in \{w, o, g\}} h^\alpha Q^{\alpha, \text{prod}} + Q^{r, f}. \quad (8)$$

We model the thermal conduction at the rock-fluid interface with Newton's law of cooling (Holman, 2010, Chap. 1):

$$Q^{r, f} = -k_T^{r, f} (T^f - T^r). \quad (9)$$

T^f is the fluid temperature, and k_T^{rf} is the thermal conductivity of the rock-fluid interface. It is only the conduction at the rock-fluid interface that affects the rock energy balance:

$$Q_u^r = -Q^{rf}. \quad (10)$$

We now assume that the rock and the fluid are in thermal equilibrium. This corresponds to energy being transferred instantly through the rock-fluid interface, i.e. k_T^{rf} is infinite and $T = T^f = T^r$. We add (5a) and (5b) to obtain an energy conservation equation for the internal energy of the combined fluid-rock system, $u = u^f + u^r$:

$$\partial_t u = -\nabla \cdot \mathbf{N}_u + Q_u. \quad (11)$$

The combined heat flux and source terms are

$$\mathbf{N}_u = h^w \mathbf{N}^w + h^o \mathbf{N}^o + h^g \mathbf{N}^g - k_T^r \nabla T, \quad (12a)$$

$$Q_u = h^{w,\text{inj}} Q^{w,\text{inj}} - \sum_{\alpha \in \{w,o,g\}} h^\alpha Q^{\alpha,\text{prod}}. \quad (12b)$$

2.4 Darcy's law

The molar phase flux is the product of density and the volumetric phase flux, $\mathbf{N}^\alpha = \rho^\alpha \mathbf{u}^\alpha$. We describe the volumetric phase flux with Darcy's law:

$$\mathbf{u}^\alpha = -(k_r^\alpha / \mu^\alpha) \mathbf{K} (\nabla P - \rho^\alpha g \nabla z). \quad (13)$$

k_r^α is relative permeability, μ^α is viscosity, \mathbf{K} is a permeability tensor, g is the gravity acceleration, and z is depth.

2.5 Well terms

The wells perforate certain grid cells in the discretized reservoir. The well models for these cells are

$$Q^{w,\text{inj}} = (1/V) \text{WI} \rho^w (k_r^w / \mu^w) (P^{\text{bhp}} - P), \quad (14a)$$

$$Q^{\alpha,\text{prod}} = (1/V) \text{WI} \rho^\alpha (k_r^\alpha / \mu^\alpha) (P - P^{\text{bhp}}). \quad (14b)$$

V is the volume of the perforated cell, WI is the well index, and P^{bhp} is the bottom-hole pressure in the well.

2.6 Relative permeabilities

We model the relative permeabilities with Stone's model II as described by Delshad and Pope (1989). The relative permeabilities are functions of the phase saturations, i.e. $k_r^\alpha = k_r^\alpha(\hat{S}^\alpha)$ where $\hat{S}^\alpha = V^\alpha / (V^w + V^o + V^g)$. The relative permeabilities are therefore functions of the temperature, pressure, and phase composition vectors:

$$k_r^\alpha = k_r^\alpha(T, P, n^w, n^o, n^g). \quad (15)$$

2.7 Viscosity

We use the phase viscosity model by Lohrenz et al. (1964). They describe the viscosity as a function of temperature, pressure, and phase composition, i.e.

$$\mu^\alpha = \mu^\alpha(T, P, n^\alpha). \quad (16)$$

2.8 Thermodynamics

We use a thermodynamical model by Ritschel et al. (2017b) to evaluate the enthalpy, entropy, and volumes of the fluid phases:

$$H^\alpha = H^\alpha(T, P, n^\alpha), \quad (17a)$$

$$S^\alpha = S^\alpha(T, P, n^\alpha), \quad (17b)$$

$$V^\alpha = V^\alpha(T, P, n^\alpha). \quad (17c)$$

We evaluate other thermodynamic functions with the fundamental thermodynamical relations $U = H - PV$, $G = H - TS$, and $A = U - TS$ (we only need U in this model).

2.9 Finite volume discretization

The conservation equations (2) and (11) are all in the form $\partial_t C = -\nabla \cdot \mathbf{N} + Q$. We integrate the equation over the i 'th grid cell, Ω_i :

$$\partial_t \int_{\Omega_i} C dV = - \int_{\Omega_i} \nabla \cdot \mathbf{N} dV + \int_{\Omega_i} Q dV. \quad (18)$$

We apply Gauss' divergence theorem to the flux term and split up the resulting surface integral:

$$\int_{\Omega_i} \nabla \cdot \mathbf{N} dV = \int_{\partial\Omega_i} \mathbf{N} \cdot \mathbf{n} dA = \sum_{j \in \mathcal{N}^{(i)}} \int_{\gamma_{ij}} \mathbf{N} \cdot \mathbf{n} dA. \quad (19)$$

$\partial\Omega_i$ is the boundary of Ω_i , and γ_{ij} is the face shared by grid cell i and j . \mathbf{n}_{ij} is the outward normal vector, and $\mathcal{N}^{(i)}$ is the set of cells that share a face with grid cell i . We evaluate the left-hand side integrals in (18) exactly:

$$U_i = \int_{\Omega_i} u dV, \quad (20a)$$

$$n_{w,i} = \int_{\Omega_i} C_w dV, \quad (20b)$$

$$n_{k,i} = \int_{\Omega_i} C_k dV. \quad (20c)$$

We approximate the remaining integrals with quadrature:

$$\int_{\Omega_i} Q dV \approx (QV)_i, \quad (21a)$$

$$\int_{\gamma_{ij}} \mathbf{N} \cdot \mathbf{n} dA \approx (\mathbf{AN} \cdot \mathbf{n})_{ij}. \quad (21b)$$

The right-hand side in (21b) contains gradients of T and P when applied to (2) and (11). We approximate these flux terms with a two-point flux approximation as described by Lie (2014). The resulting differential equations are

$$\dot{U}_i = \sum_{j \in \mathcal{N}^{(i)}} \left(\sum_{\alpha \in \{w,o,g\}} (h^\alpha \Gamma \hat{H}^\alpha \Delta \Phi^\alpha)_{ij} + (\Gamma_T \Delta T)_{ij} \right) + (Q_u V)_i, \quad (22a)$$

$$\dot{n}_{w,i} = \sum_{j \in \mathcal{N}^{(i)}} (\Gamma \hat{H}^w \Delta \Phi^w)_{ij} + (Q^w V)_i, \quad (22b)$$

$$\dot{n}_{k,i} = \sum_{j \in \mathcal{N}^{(i)}} (x_k \Gamma \hat{H}^o \Delta \Phi^o + y_k \Gamma \hat{H}^g \Delta \Phi^g)_{ij} + (Q_k V)_i. \quad (22c)$$

The term $(\Gamma \hat{H}^\alpha \Delta \Phi^\alpha)_{ij}$ approximates $-(\mathbf{AN}^\alpha \cdot \mathbf{n})_{ij}$. Γ_{ij} is the geometric part of the transmissibilities:

$$\Gamma_{ij} = A_{ij} \left(\hat{\Gamma}_{ij}^{-1} + \hat{\Gamma}_{ji}^{-1} \right)^{-1}, \quad (23a)$$

$$\hat{\Gamma}_{ij} = \left(\mathbf{K}_i \frac{c_{ij} - c_i}{|c_{ij} - c_i|^2} \right) \cdot \mathbf{n}_{ij}. \quad (23b)$$

A_{ij} is the area of γ_{ij} , c_{ij} is the center of γ_{ij} , and c_i is the center of Ω_i . $\hat{\Gamma}_{ij}$ is the one-sided transmissibility. We define $\Gamma_{T,ij}$ similar to Γ_{ij} where k_T^r replaces \mathbf{K} . The potential difference and the fluid part of the transmissibilities are

$$\Delta\Phi_{ij}^\alpha = (\Delta P - \rho^\alpha g \Delta z)_{ij}, \quad (24a)$$

$$\hat{H}_{ij}^\alpha = \begin{cases} (\rho^\alpha k_r^\alpha / \mu^\alpha)_i, & \Delta\Phi_{ij}^\alpha < 0, \\ (\rho^\alpha k_r^\alpha / \mu^\alpha)_j, & \Delta\Phi_{ij}^\alpha \geq 0, \end{cases} \quad (24b)$$

where $\Delta P_{ij} = P_j - P_i$, $\Delta z_{ij} = z_j - z_i$, and $\rho_{ij}^\alpha = 0.5(\rho_i^\alpha + \rho_j^\alpha)$. We have upwinded the fluid part of the transmissibilities, \hat{H}^α , in order to ensure numerical stability. We upwind x_k , y_k , and h^α in the same way.

2.10 Differential-algebraic model

We introduce the state variables $x_i = [U; n_w; n]_i \in \mathbb{R}^{2+N_C}$, the algebraic variables $y_i = [T; P; n^w; n^o; n^g]_i \in \mathbb{R}^{3+2N_C}$, the manipulated input variables $u_i = P_i^{\text{bhp}} \in \mathbb{R}$, and the disturbance variables $d_i = T_i^{\text{inj}} \in \mathbb{R}$. T^{inj} is the temperature of the injected water used to evaluate $h^{w,\text{inj}}$ in (8). The UV flash problem (1) is thus in the form

$$\min_{y_i} f(y_i), \quad (25a)$$

$$\text{s.t.} \quad g(y_i) = x_i, \quad (25b)$$

$$h(y_i) = 0. \quad (25c)$$

The optimality conditions of (25) are in the form $G_i(x_i, y_i, z_i)$ where $z_i \in \mathbb{R}^{3+N_C}$ are Lagrange multipliers (Ritschel et al., 2017a). We enforce the phase equilibrium in each grid cell. The left-hand side of the differential equations (22) contains derivatives of the state variables, x_i , and all quantities on the right-hand side depend on the algebraic variables, y_i , the manipulated inputs, u_i , or the disturbance variables, d_i . The differential equations (22) are therefore in the form $\dot{x}_i(t) = F(y_i(t), u_i(t), d_i(t))$, and the discretized reservoir flow model is in the form

$$G(x(t), y(t), z(t)) = 0, \quad (26a)$$

$$\dot{x}(t) = F(y(t), u(t), d(t)), \quad (26b)$$

where G is the phase equilibrium conditions for all cells, and F is the spatially discretized right-hand side of the flow equations for all cells.

3. PRODUCTION OPTIMIZATION

In this section, we briefly describe the gradient-based single-shooting algorithm by Ritschel et al. (2017a). The production optimization problem is in the form

$$\min_{\substack{\{x(t); y(t); z(t)\}_{t_0}^{t_f}, \\ \{u_k\}_{k \in \mathcal{N}}}} \phi = \int_{t_0}^{t_f} \Phi(y(t), u(t), d(t)) dt, \quad (27a)$$

subject to

$$x(t_0) = \hat{x}_0, \quad (27b)$$

$$G(x(t), y(t), z(t)) = 0, \quad t \in \mathcal{T}, \quad (27c)$$

$$\dot{x}(t) = F(y(t), u(t), d(t)), \quad t \in \mathcal{T}, \quad (27d)$$

$$u(t) = u_k, \quad t \in [t_k, t_{k+1}[, \quad k \in \mathcal{N}, \quad (27e)$$

$$d(t) = \hat{d}_k, \quad t \in [t_k, t_{k+1}[, \quad k \in \mathcal{N}, \quad (27f)$$

$$\{u_k\}_{k \in \mathcal{N}} \in \mathcal{U}. \quad (27g)$$

The objective function, ϕ in (27a), is a financial measure, e.g. total oil production or net present value, (27b) is an initial condition on the state variables, (27c)-(27d) are the equilibrium conditions and the spatially discretized reservoir flow equations, and (27e)-(27f) are zero-order

hold parametrizations of the manipulated inputs and disturbance variables. Finally, (27g) are constraints on the manipulated inputs, typically linear or bound constraints. $\mathcal{T} = [t_0, t_f]$ is the time interval, and $\mathcal{N} = \{0, \dots, N-1\}$ is the set of timestep indices. N is the number of timesteps.

3.1 Numerical simulation

For simplicity, we describe the algorithm with the assumption that the timesteps coincide with the control intervals. The actual implementation uses an ESDIRK12 method with a simplified version of the stepsize controller described by Völcker et al. (2010). The differential equations are discretized with Euler's implicit method. That results in the nonlinear residual equations $R_{k+1} = 0$ where

$$R_{k+1} = \begin{bmatrix} x_{k+1} - x_k - \Delta t_k F(y_{k+1}, u_k, \hat{d}_k) \\ G(x_{k+1}, y_{k+1}, z_{k+1}) \end{bmatrix}, \quad (28)$$

for $k \in \mathcal{N}$. We introduce $w = [x; y; z]$ and solve the nonlinear equations with Newton's method:

$$w_{k+1}^{m+1} = w_{k+1}^m - (\partial R_{k+1} / \partial w_{k+1})^{-1} R_{k+1}(w_{k+1}^m). \quad (29)$$

We use an ILU preconditioned GMRES method to solve the linear system in (29).

3.2 The single-shooting algorithm

In the single-shooting approach, we transcribe the infinite-dimensional optimal control problem (27) into the following finite-dimensional optimization problem

$$\min_{\{u_k\}_{k \in \mathcal{N}}} \psi = \psi(\{u_k\}_{k \in \mathcal{N}}; \hat{x}_0, \{\hat{d}_k\}_{k \in \mathcal{N}}), \quad (30a)$$

$$\text{subject to} \quad \{u_k\}_{k \in \mathcal{N}} \in \mathcal{U}, \quad (30b)$$

where the objective function is

$$\psi = \left\{ \phi = \sum_{k \in \mathcal{N}} \Phi_k(y_{k+1}, u_k, \hat{d}_k) : \right. \quad (31a)$$

$$x_0 = \hat{x}_0, \quad (31b)$$

$$\left. R_{k+1}(w_{k+1}; x_k, u_k, \hat{d}_k) = 0, \quad k \in \mathcal{N} \right\}, \quad (31c)$$

and $\Phi_k(y_{k+1}, u_k, \hat{d}_k) = \Delta t_k \Phi(y_{k+1}, u_k, \hat{d}_k)$. Efficient algorithms for the solution of the optimization problem (30) require the gradients of ψ , $\{\nabla_{u_k} \psi\}_{k \in \mathcal{N}}$. We use a discrete adjoint method to compute these gradients. We solve the following adjoint equations for the adjoints, $\{\lambda_{k+1}\}_{k \in \mathcal{N}}$, with an ILU preconditioned GMRES method:

$$(\partial R_N / \partial w_N)^T \lambda_N = -\nabla_{w_N} \Phi_{N-1}, \quad (32a)$$

$$(\partial R_k / \partial w_k)^T \lambda_k = -(\partial R_{k+1} / \partial w_k)^T \lambda_{k+1} - \nabla_{w_k} \Phi_{k-1}. \quad (32b)$$

The adjoint equations are solved in a backwards manner starting with (32a) and proceeding with (32b) for $k = N-1, N-2, \dots, 1$. The gradients of ψ are computed with

$$\nabla_{u_k} \psi = \nabla_{u_k} \Phi_k + (\partial R_{k+1} / \partial u_k)^T \lambda_{k+1}, \quad k \in \mathcal{N}. \quad (33)$$

4. IMPLEMENTATION

We implement the single-shooting algorithm in C++. We use the DUNE software to solve linear systems with iterative methods (Blatt and Bastian, 2007) and for grid management (Bastian et al., 2008). We use C routines

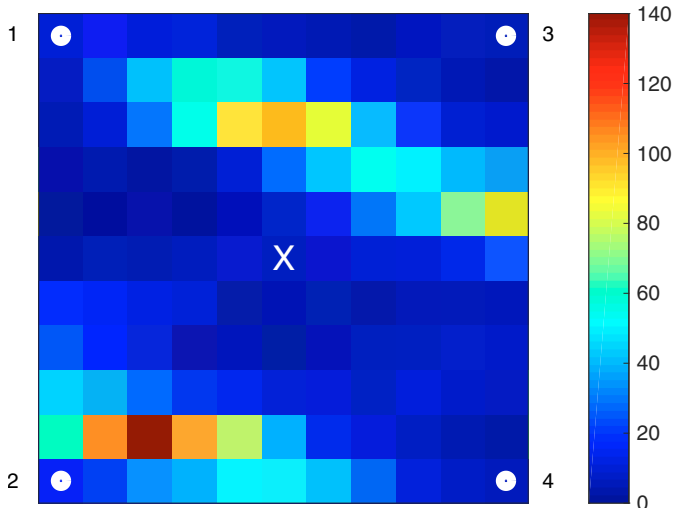


Fig. 1. Permeability field [mD]. A white circle indicates an injector, and the white X indicates the producer.

from the open-source software ThermoLib (Ritschel et al., 2017b, 2016) to evaluate thermodynamic functions. We use the optimization software KNITRO 10.2 to solve the optimization problem (30).

5. NUMERICAL EXAMPLE

In this section, we present a numerical example where four injectors inject water at 40°C into a $110 \times 110 \times 10$ m reservoir that contains a fluid at 20°C . The oil and gas phases consist of methane, ethane, propane, n-heptane, and hydrogen sulfide. The reservoir is discretized with $11 \times 11 \times 1$ cells. The objective is to maximize the oil production. The wells are placed in a five spot pattern as shown in Fig. 1 which also shows the permeability field. The producer bottom-hole pressure (BHP) must be in the interval $[10 \text{ MPa}, 11 \text{ MPa}]$ and the injector BHPs must be in $[11 \text{ MPa}, 12 \text{ MPa}]$. The optimal production strategy is shown in Fig. 2 together with the cumulative oil and gas production. Fig. 2 also shows (in dashed lines) the oil and gas production for a maximum injection strategy with maximum injector BHP and minimum producer BHP. The optimal oil production is 35% higher than what is obtained with the maximum injection strategy while the gas production is lower.

6. CONCLUSIONS

In this work, we present a fully thermal and compositional reservoir flow model based on a rigorous formulation of the phase equilibrium using the second law of thermodynamics, i.e. the entropy of a closed system in equilibrium is maximal. This results in an inner optimization problem called the UV flash problem. The reservoir flow model is in the semi-explicit index-1 differential-algebraic form that Ritschel et al. (2017a) consider, and we implement their gradient-based single-shooting algorithm for production optimization. We present a numerical example where the injected water has a higher temperature than the reservoir fluid. Future work will involve isothermal compositional models based on the model presented in this work.

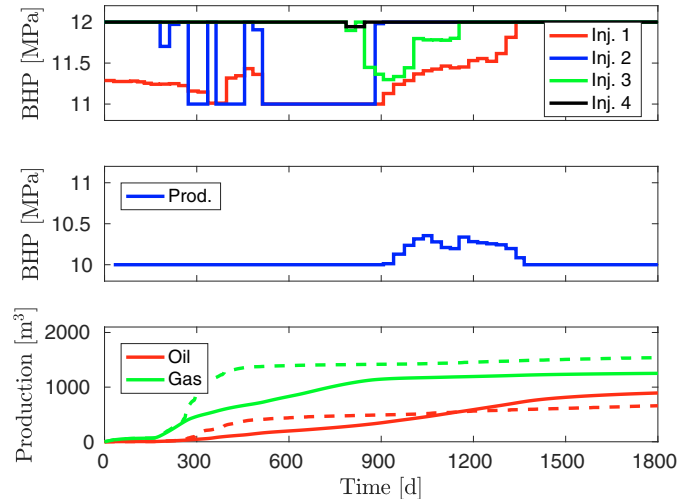


Fig. 2. Optimal well BHPs and oil and gas production (solid: optimal strategy, dashed: reference strategy).

REFERENCES

- Arendsen, A.R.J. and Versteeg, G.F. (2009). Dynamic thermodynamics with internal energy, volume, and amount of moles as states: Application to liquefied gas tank. *Ind. Eng. Chem. Res.*, 48(6), 3167–3176.
- Bastian, P., Blatt, M., Dedner, A., Engwer, C., Klöforn, R., Ohlberger, M., and Sander, O. (2008). A generic grid interface for parallel and adaptive scientific computing. Part I: Abstract framework. *Computing*, 82(2-3), 103–119.
- Blatt, M. and Bastian, P. (2007). The iterative solver template library. In B. Kågström et al. (ed.), *Applied Parallel Computing. State of the Art in Scientific Computing. PARA 2006*, volume 4699 of *Lecture Notes in Computer Science*. Springer, Berlin, Heidelberg.
- Bukshtynov, V., Volkov, O., Durlifsky, L.J., and Aziz, K. (2015). Comprehensive framework for gradient-based optimization in closed-loop reservoir management. *Comput. Geosci.*, 19(4), 877–897.
- Capolei, A. and Jørgensen, J.B. (2012). Solution of constrained optimal control problems using multiple shooting and ESDIRK methods. In *Proceedings of the 2012 American Control Conference*, 295–300.
- Capolei, A., Völcker, C., Frydendall, J., and Jørgensen, J.B. (2012). Oil reservoir production optimization using single shooting and ESDIRK methods. In *Automatic Control in Offshore Oil and Gas Production*, 286–291. International Federation of Automatic Control.
- Castier, M. (2010). Dynamic simulation of fluids in vessels via entropy maximization. *J. Ind. Eng. Chem.*, 16(1), 122–129.
- Codas, A., Hanssen, K.G., Foss, B., Capolei, A., and Jørgensen, J.B. (2017). Multiple shooting applied to robust reservoir control optimization including output constraints on coherent risk measures. *Comput. Geosci.*, 21(3), 479–497.
- Delshad, M. and Pope, G.A. (1989). Comparison of the three-phase oil relative permeability models. *Transp. Porous Media*, 4(1), 59–83.
- Flatby, P., Skogestad, S., and Lundström, P. (1994). Rigorous dynamic simulation of distillation columns based on UV-flash. In *IFAC Symposium on Advanced Control*

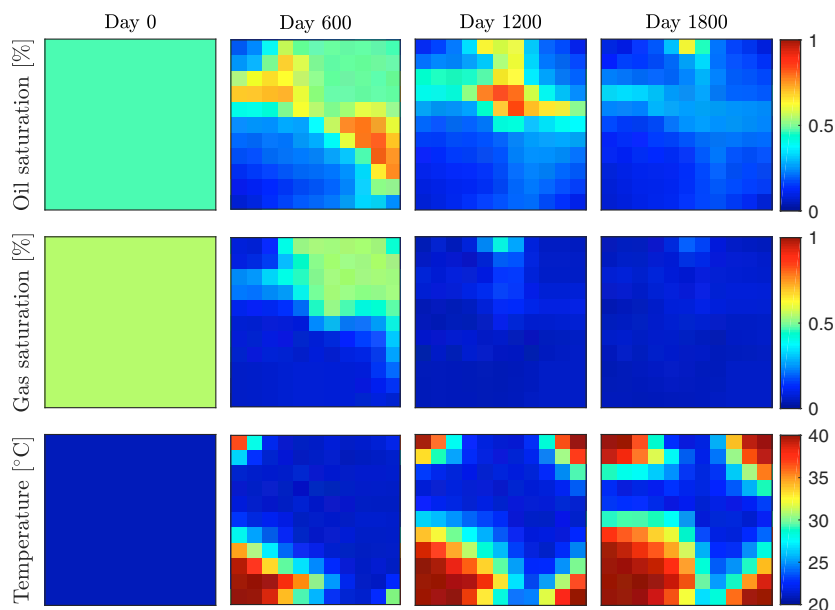


Fig. 3. Oil saturation, gas saturation, and temperature for the optimal oil production strategy.

- of Chemical Processes (ADCHEM '94), 261–266.
- Forouzanfar, F., Rossa, E.D., Russo, R., and Reynolds, A.C. (2013). Life-cycle production optimization of an oil field with an adjoint-based gradient approach. *J. Pet. Sci. Eng.*, 112, 351–358.
- Hammer, M. and Morin, A. (2014). A method for simulating two-phase pipe flow with real equations of state. *Comput. Fluids*, 100, 45–58.
- Heirung, T.A.N., Wartmann, M.R., Jansen, J.D., Ydstie, B.E., and Foss, B.A. (2011). Optimization of the water-flooding process in a small 2D horizontal oil reservoir by direct transcription. In *Proceedings of the 18th World Congress*, 10863–10868. International Federation of Automatic Control, Milano, Italy.
- Holman, J.P. (2010). *Heat Transfer*. McGraw-Hill, 10th edition.
- Kourounis, D., Durlofsky, L.J., Jansen, J.D., and Aziz, K. (2014). Adjoint formulation and constraint handling for gradient-based optimization of compositional reservoir flow. *Comput. Geosci.*, 18(2), 117–137.
- Lei, Y., Li, S., Zhang, X., Zhang, Q., and Guo, L. (2012). Optimal control of polymer flooding based on maximum principle. *J. Appl. Math.*
- Lie, K.A. (2014). *An Introduction to Reservoir Simulation Using MATLAB*. Sintef ICT, Oslo, Norway.
- Lima, E.R.A., Castier, M., and Biscaia Jr., E.C. (2008). Differential-algebraic approach to dynamic simulations of flash drums with rigorous evaluation of physical properties. *Oil Gas Sci. Technol. Rev. IFP*, 63(5), 677–686.
- Lohrenz, J., Bray, B.G., and Clark, C.R. (1964). Calculating viscosities of reservoir fluids from their compositions. *J. Pet. Technol.*, 16(10), 1171–1176.
- Michelsen, M.L. (1999). State function based flash specifications. *Fluid Phase Equilib.*, 158-160, 617–626.
- Onwunalu, J.E. and Durlofsky, L.J. (2010). Application of a particle swarm optimization algorithm for determining optimum well location and type. *Comput. Geosci.*, 14(1), 183–198.
- Qiu, L., Wang, Y., and Reitz, R.D. (2014). Multiphase dynamic flash simulations using entropy maximization and application to compressible flow with phase change. *AIChE J.*, 60(8), 3013–3024.
- Ritschel, T.K.S., Capolei, A., Gaspar, J., and Jørgensen, J.B. (2017a). An algorithm for gradient-based dynamic optimization of UV-flash processes. *Comput. Chem. Eng.* Accepted.
- Ritschel, T.K.S., Gaspar, J., and Jørgensen, J.B. (2017b). A thermodynamic library for simulation and optimization of dynamic processes. In *Proceedings of the 20th World Congress of the International Federation of Automatic Control*.
- Ritschel, T.K.S., Gaspar, J., Capolei, A., and Jørgensen, J.B. (2016). An open-source thermodynamic software library. Technical Report DTU Compute Technical Report-2016-12, Department of Applied Mathematics and Computer Science, Technical University of Denmark.
- Saputelli, L., Malki, H., Canelon, J., and Nikolaou, M. (2002). A critical overview of artificial neural network applications in the context of continuous oil field optimization. In *SPE Annual Technical Conference and Exhibition*. Society of Petroleum Engineers, San Antonio, Texas.
- Völcker, C., Jørgensen, J.B., Thomsen, P.G., and Stenby, E.H. (2010). Explicit singly diagonally implicit Runge-Kutta methods and adaptive stepsize control for reservoir simulation. In *Proceedings of the 12th European Conference on the Mathematics of Oil Recovery*. Oxford, UK.
- Zaydullin, R., Voskov, D.V., James, S.C., Henley, H., and Lucia, A. (2014). Fully compositional and thermal reservoir simulation. *Comput. Chem. Eng.*, 63, 51–65.
- Zhao, H., Tang, Y.W., Li, Y., Shi, Y.B., Cao, L., Gong, R.X., and Shang, G.H. (2016). Reservoir production optimization using general stochastic approximate algorithm under the mixed-linear-nonlinear constraints. *J Residuals Sci. Technol.*, 13(8).

Paper VIII

The Extended Kalman Filter for State Estimation of Dynamic UV Flash Processes

Authors:

Tobias K. S. Ritschel and John Bagterp Jørgensen

Published in:

IFAC-PapersOnLine 51(8), 164–169, 2018.

DOI: <https://doi.org/10.1016/j.ifacol.2018.06.372>.

The Extended Kalman Filter for State Estimation of Dynamic UV Flash Processes^{*}

Tobias K. S. Ritschel, John Bagterp Jørgensen

*Department of Applied Mathematics and Computer Science &
Center for Energy Resources Engineering (CERE),
Technical University of Denmark, DK-2800 Kgs. Lyngby, Denmark*

Abstract: We present an extended Kalman filter for state estimation of semi-explicit index-1 differential-algebraic equations. It is natural to model dynamic UV flash processes with such differential-algebraic equations. The UV flash is a mathematical statement of the second law of thermodynamics. It is therefore important to thermodynamically rigorous models of many phase equilibrium processes. State estimation of UV flash processes has applications in control, prediction, monitoring, and fault detection of chemical processes in the oil and gas industry, e.g. separation, distillation, drilling of oil wells, multiphase flow in oil pipes, and oil production. We present a numerical example of a UV flash separation process. It involves soft sensing of vapor-liquid compositions based on temperature and pressure measurements.

© 2018, IFAC (International Federation of Automatic Control) Hosting by Elsevier Ltd. All rights reserved.

Keywords: Extended Kalman filter, State estimation, UV flash, Differential-algebraic equations

1. INTRODUCTION

State estimation is concerned with reconstructing the state of a process based on noisy measurements and a model of the process. It is central to nonlinear model predictive control, prediction, monitoring, and fault detection of several chemical processes in the oil and gas industry. For instance, authors have considered state estimation of distillation columns (Pan et al., 2009; Kataria et al., 2016), oil and gas pipe flow (Binder et al., 2015), oil well drilling (Nikoofard et al., 2017), and oil reservoir production (Oliver and Chen, 2011).

Many chemical processes involve phase equilibrium, i.e. thermodynamic equilibrium between two or more phases. The condition of phase equilibrium is derived from the second law of thermodynamics which states that the entropy of a closed system in equilibrium is maximal. The UV flash problem is a mathematical statement of the second law of thermodynamics. It is therefore a key component in thermodynamically rigorous models of phase equilibrium processes, e.g. flash separation (Castier, 2010; Arendsen and Versteeg, 2009; Lima et al., 2008), distillation columns (Flatby et al., 1994), and computational fluid dynamical problems (Hammer and Morin, 2014; Qiu et al., 2014). The UV flash problem can be formulated as an optimization problem (Michelsen, 1999). The solution to the optimization problem is the temperature, pressure, and phase compositions that maximize entropy. The optimization problem contains equality constraints on internal energy, U , volume, V , and total mass of each chemical component, n . U , V , and n are parameters in the optimization problem. The phase equilibrium conditions are the optimality

^{*} This project is funded by Innovation Fund Denmark in the OPTION project (63-2013-3).

conditions of the optimization problem which are algebraic equations. Dynamical UV flash processes are therefore modeled with differential-algebraic equations (DAEs). Recently, Ritschel et al. (2017a) developed dynamic optimization algorithms for UV flash processes. However, state estimation of dynamic UV flash processes has not been treated in the open literature.

The Kalman filter is optimal for linear systems. However, many chemical processes are inherently nonlinear such that the Kalman filter cannot be used. There exist a number of nonlinear filters, e.g. the extended Kalman filter, the unscented Kalman filter, and particle filters (Simon, 2006). The extended Kalman filter applies the Kalman filter equations to a linearization of the nonlinear model. For highly nonlinear processes, this linearization can limit the accuracy of the extended Kalman filter. The unscented Kalman filter and particle filters use samples of the states to provide better estimates than the extended Kalman filter for severely nonlinear processes. The unscented Kalman filter uses deterministic samples whereas particle filters use random samples. A particular particle filter, called the ensemble Kalman filter, has gained much attention in oceanography and oil reservoir characterization (Evensen, 2009a,b; Gillijns et al., 2006). Alternatives to the above state estimation algorithms include moving-horizon estimation (Alessandri et al., 2010), which is an optimization-based algorithm, and neural network-based algorithms (Talebi et al., 2010). Research on state estimation algorithms was originally focused on systems of ordinary differential equations (ODEs). However, many processes are naturally modeled with DAEs. Algebraic equations often result from the approximation of a fast process as a quasi-steady-state, e.g. it is common to assume that phase equilibrium occurs instantaneously in dy-

dynamic processes. Recently, several authors have developed algorithms for state estimation of DAE models, e.g. the extended Kalman filter (Mobed et al., 2016; Jørgensen et al., 2007; Becerra et al., 2001), the unscented Kalman filter (Purohit et al., 2015, 2013; Pastorino et al., 2013; Mandela et al., 2010, 2009), particle filters (Haßkerl et al., 2017, 2016), and the ensemble Kalman filter (Puranik et al., 2012).

In this work, we present an extended Kalman filter for state estimation of dynamic UV flash processes. Such processes are modeled with DAEs in a semi-explicit index-1 form where the right-hand side of the differential equations are independent of the differential states. We exploit this fact in the computations. We present a numerical example that involves soft sensing of vapor-liquid compositions in a UV flash separation process based on temperature and pressure measurements.

The remainder of this paper is structured as follows. In Section 2, we describe the stochastic semi-explicit index-1 DAE system that we consider, and in Section 3, we describe the numerical simulation of such systems. In Section 4, we describe the extended Kalman filter. We briefly describe the model of the UV flash separation process in Section 5, and we present numerical results in Section 6. In Section 7, we present conclusions.

2. STOCHASTIC SEMI-EXPLICIT INDEX-1 DAE SYSTEMS

We consider stochastic DAE systems in the form

$$G(x(t), y(t), z(t)) = 0, \quad (1a)$$

$$dx(t) = F(y(t), u(t))dt + \sigma(y(t), u(t))d\omega(t). \quad (1b)$$

$x(t)$ is the state vector, $y(t)$ is a vector of algebraic variables, and $z(t)$ is a vector of adjoint algebraic variables. Phase equilibrium conditions can be formulated as the Karush-Kuhn-Tucker (KKT) conditions of an optimization problem. The algebraic equations (1a) represent such KKT conditions, and $z(t)$ represents the corresponding Lagrange multipliers. The stochastic differential equations (1b) represent conservation equations, and the states represent the conserved quantities. The right-hand side of the stochastic differential equations is independent of the states. The initial states are distributed as $x(t_0) \sim N(x_0, P_0)$. $u(t)$ are manipulated inputs, and $\omega(t)$ is a standard Wiener process, i.e. it has an incremental covariance of Idt . For the systems that we consider, the algebraic equations (1a) can be solved for $y(t)$ and $z(t)$ when $x(t)$ is given, i.e. the DAE system (1) is of index 1. The system is observed at discrete times, t_k , using the measurement equation,

$$y^m(t_k) = H(y(t_k)) + v(t_k). \quad (2)$$

$y^m(t_k)$ are the measurements. The right-hand side of the measurement equation is independent of the states. The measurement noise, $v(t_k)$, is normally distributed, i.e. $v(t_k) = v_k \sim N(0, T_k)$.

3. NUMERICAL SIMULATION

In this section, we describe the numerical simulation of the stochastic DAE system (1). We solve the differential equations and the algebraic equations in a simultaneous

manner. We discretize the deterministic and stochastic part of the stochastic differential equations (1b) with Euler's implicit and explicit method, respectively. That results in the equation $D_{k+1} = 0$ where

$$\begin{aligned} D_{k+1} &= D_{k+1}(x_{k+1}, y_{k+1}) = D_{k+1}(x_{k+1}, y_{k+1}; x_k, y_k, u_k) \\ &= x_{k+1} - F(y_{k+1}, u_k)\Delta t_k - \sigma(y_k, u_k)\Delta\omega_k - x_k. \end{aligned} \quad (3)$$

We introduce $w_{k+1} = [x_{k+1}; y_{k+1}; z_{k+1}]$. For each time step, we solve the residual equations, $R_{k+1} = 0$, for w_{k+1} . The residual function is

$$\begin{aligned} R_{k+1} &= R_{k+1}(w_{k+1}) = R_{k+1}(x_{k+1}, y_{k+1}, z_{k+1}) \\ &= R_{k+1}(x_{k+1}, y_{k+1}, z_{k+1}; x_k, y_k, u_k) \\ &= \begin{bmatrix} D_{k+1}(x_{k+1}, y_{k+1}; x_k, y_k, u_k) \\ G(x_{k+1}, y_{k+1}, z_{k+1}) \end{bmatrix}. \end{aligned} \quad (4)$$

We solve the residual equations with Newton's method:

$$w_{k+1}^{m+1} = w_{k+1}^m + \Delta w_{k+1}^m. \quad (5)$$

In each Newton iteration, we solve the linear system

$$M\Delta w_{k+1}^m = -R_{k+1}(w_{k+1}^m), \quad (6)$$

where the iteration matrix, M , is

$$M \approx \frac{\partial R_{k+1}}{\partial w_{k+1}} = \begin{bmatrix} I & -\frac{\partial F}{\partial y}\Delta t_k & 0 \\ \frac{\partial G}{\partial x} & \frac{\partial G}{\partial y} & \frac{\partial G}{\partial z} \end{bmatrix}. \quad (7)$$

3.1 Efficient solution of the linear system

The main computational task in the solution of the linear system (6) is the factorization of the iteration matrix, M . Because of the identity matrix in the Jacobian in (7), we can obtain an explicit expression for Δx_{k+1}^m from (6):

$$\Delta x_{k+1}^m = \left(\frac{\partial F}{\partial y}\Delta t_k \right) \Delta y_{k+1}^m - D_{k+1}. \quad (8)$$

We use (6) and (8) to obtain a reduced linear system for Δy_{k+1}^m and Δz_{k+1}^m :

$$\bar{M} \begin{bmatrix} \Delta y_{k+1}^m \\ \Delta z_{k+1}^m \end{bmatrix} = \frac{\partial G}{\partial x} D_{k+1} - G(x_{k+1}^m, y_{k+1}^m, z_{k+1}^m). \quad (9)$$

The reduced iteration matrix, \bar{M} , is

$$\bar{M} \approx \begin{bmatrix} \frac{\partial G}{\partial y} + \frac{\partial G}{\partial x} \frac{\partial F}{\partial y}\Delta t_k & \frac{\partial G}{\partial z} \end{bmatrix}. \quad (10)$$

It is cheaper to factorize \bar{M} than M because \bar{M} is smaller.

4. THE EXTENDED KALMAN FILTER

The extended Kalman filter consists of a) a measurement-update that incorporates the current measurement and b) a time-update that propagates the state mean and its covariance through time in between measurements. The initial state estimate and its covariance are the mean and covariance of the initial states:

$$\hat{x}_{0|-1} = x_0, \quad (11a)$$

$$P_{0|-1} = P_0. \quad (11b)$$

4.1 Measurement-update

The one-step ahead prediction of the measurement, $\hat{y}_{k|k-1}^m$, and its approximate covariance matrix, $T_{k|k-1}$, are

$$\hat{y}_{k|k-1}^m = H(\hat{y}_{k|k-1}), \quad (12a)$$

$$T_{k|k-1} = C_k P_{k|k-1} C_k' + T_k. \quad (12b)$$

The one-step ahead estimate of the algebraic variables, $\hat{y}_{k|k-1}$, is available from the previous time-update. T_k is the covariance matrix of the measurement noise. The matrix C_k is

$$\begin{aligned} C_k &= \frac{\partial H}{\partial x}(\hat{y}_{k|k-1}) \\ &= \frac{\partial H}{\partial y}(\hat{y}_{k|k-1}) \frac{\partial \hat{y}_{k|k-1}}{\partial \hat{x}_{k|k-1}}. \end{aligned} \quad (13)$$

As we describe later, the one-step ahead estimates satisfy the algebraic equations, i.e. $G(\hat{x}_{k|k-1}, \hat{y}_{k|k-1}, \hat{z}_{k|k-1}) = 0$. We apply the implicit function theorem to the algebraic equations in order to compute $\frac{\partial \hat{y}_{k|k-1}}{\partial \hat{x}_{k|k-1}}$ and $\frac{\partial \hat{z}_{k|k-1}}{\partial \hat{x}_{k|k-1}}$:

$$\begin{bmatrix} \frac{\partial G}{\partial y} & \frac{\partial G}{\partial z} \end{bmatrix} \begin{bmatrix} \frac{\partial \hat{y}_{k|k-1}}{\partial \hat{x}_{k|k-1}} \\ \frac{\partial \hat{z}_{k|k-1}}{\partial \hat{x}_{k|k-1}} \end{bmatrix} = -\frac{\partial G}{\partial x}. \quad (14)$$

The partial derivatives of G in (14) are evaluated at $\hat{x}_{k|k-1}$, $\hat{y}_{k|k-1}$, and $\hat{z}_{k|k-1}$. The innovation error of the measurement is

$$e_k = y_k^m - \hat{y}_{k|k-1}^m, \quad (15)$$

where $y_k^m = y^m(t_k)$ is the measurement. The Kalman filter gain matrix is

$$K_{f,x,k} = P_{k|k-1} C_k' T_k^{-1}. \quad (16)$$

The filtered state and its approximate covariance matrix are updated by

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_{f,x,k} e_k, \quad (17a)$$

$$P_{k|k} = P_{k|k-1} - K_{f,x,k} T_k^{-1} K_{f,x,k}'. \quad (17b)$$

The filtered estimates of the algebraic and adjoint algebraic variables satisfy the algebraic equations:

$$G(\hat{x}_{k|k}, \hat{y}_{k|k}, \hat{z}_{k|k}) = 0. \quad (18)$$

The corresponding approximate covariance matrices are

$$P_{y,k|k} = \Phi_{yx}(t_k, t_k) P_{k|k} \Phi_{yx}(t_k, t_k)', \quad (19a)$$

$$P_{z,k|k} = \Phi_{zx}(t_k, t_k) P_{k|k} \Phi_{zx}(t_k, t_k)'. \quad (19b)$$

We use the implicit function theorem to compute the sensitivities, $\Phi_{yx}(t_k, t_k) = \frac{\partial \hat{y}_{k|k}}{\partial \hat{x}_{k|k}}$ and $\Phi_{zx}(t_k, t_k) = \frac{\partial \hat{z}_{k|k}}{\partial \hat{x}_{k|k}}$:

$$\begin{bmatrix} \frac{\partial G}{\partial y} & \frac{\partial G}{\partial z} \end{bmatrix} \begin{bmatrix} \Phi_{yx}(t_k, t_k) \\ \Phi_{zx}(t_k, t_k) \end{bmatrix} = -\frac{\partial G}{\partial x}. \quad (20)$$

The partial derivatives of G in (20) are evaluated at $\hat{x}_{k|k}$, $\hat{y}_{k|k}$, and $\hat{z}_{k|k}$.

4.2 Time-update

In the time-update, we propagate the state estimate and covariance matrix from time t_k to time t_{k+1} where the next measurement arrives. We obtain the one-step ahead predictions at time t_{k+1} by solving the initial value problem

$$\hat{x}_k(t_k) = \hat{x}_{k|k}, \quad (21a)$$

$$G(\hat{x}_k(t), \hat{y}_k(t), \hat{z}_k(t)) = 0, \quad t \in [t_k; t_{k+1}], \quad (21b)$$

$$\frac{d\hat{x}_k(t)}{dt} = F(\hat{y}_k(t), u(t)), \quad t \in [t_k; t_{k+1}]. \quad (21c)$$

The sensitivities, $\Phi_{xx}(t, s) = \frac{\partial \hat{x}_k(t)}{\partial \hat{x}_k(s)}$, $\Phi_{yx}(t, s) = \frac{\partial \hat{y}_k(t)}{\partial \hat{x}_k(s)}$, and $\Phi_{zx}(t, s) = \frac{\partial \hat{z}_k(t)}{\partial \hat{x}_k(s)}$, satisfy

$$\Phi_{xx}(s, s) = I, \quad (22a)$$

$$\frac{\partial G}{\partial x} \Phi_{xx}(t, s) + \frac{\partial G}{\partial y} \Phi_{yx}(t, s) + \frac{\partial G}{\partial z} \Phi_{zx}(t, s) = 0, \quad (22b)$$

$$\frac{d\Phi_{xx}(t, s)}{dt} = \frac{\partial F}{\partial y} \Phi_{yx}(t, s). \quad (22c)$$

The partial derivatives of F and G in (22) are evaluated at $\hat{x}_k(t)$, $\hat{y}_k(t)$, and $\hat{z}_k(t)$. We compute the covariance matrix from the sensitivities (Jørgensen et al., 2007):

$$\begin{aligned} P_k(t) &= \Phi_{xx}(t, t_k) P_{k|k} \Phi_{xx}(t, t_k)' \\ &+ \int_{t_k}^t \Phi_{xx}(t, s) \sigma(\hat{y}_k(s), u(s)) \sigma(\hat{y}_k(s), u(s))' \Phi_{xx}(t, s)' ds. \end{aligned} \quad (23)$$

4.3 Numerical solution of the time-update equations

We use Euler's implicit method to solve the initial value problem (21) for the one-step ahead predictions, $\hat{x}_{k+1|k} = \hat{x}_k(t_{k+1})$, $\hat{y}_{k+1|k} = \hat{y}_k(t_{k+1})$, and $\hat{z}_{k+1|k} = \hat{z}_k(t_{k+1})$. That corresponds to solving the nonlinear equations

$$\begin{aligned} R_{k+1|k} &= R_{k+1|k}(\hat{x}_{k+1|k}, \hat{y}_{k+1|k}, \hat{z}_{k+1|k}) \\ &= R_{k+1|k}(\hat{x}_{k+1|k}, \hat{y}_{k+1|k}, \hat{z}_{k+1|k}; \hat{x}_{k|k}, u_k) \\ &= \begin{bmatrix} \hat{x}_{k+1|k} - F(\hat{y}_{k+1|k}, u_k) \Delta t_k - \hat{x}_{k|k} \\ G(\hat{x}_{k+1|k}, \hat{y}_{k+1|k}, \hat{z}_{k+1|k}) \end{bmatrix} = 0. \end{aligned} \quad (24)$$

We solve the nonlinear equations (24) with Newton's method. The approach is similar to the one described in Section 3. We also discretize the sensitivity equations (22) with Euler's implicit method:

$$\begin{bmatrix} I & -\frac{\partial F}{\partial y} \Delta t_k & 0 \\ \frac{\partial G}{\partial x} & \frac{\partial G}{\partial y} & \frac{\partial G}{\partial z} \end{bmatrix} \begin{bmatrix} \Phi_{xx}(t_{k+1}, t_k) \\ \Phi_{yx}(t_{k+1}, t_k) \\ \Phi_{zx}(t_{k+1}, t_k) \end{bmatrix} = \begin{bmatrix} I \\ 0 \end{bmatrix}. \quad (25)$$

The partial derivatives of F and G in (25) are evaluated at $\hat{x}_{k+1|k}$, $\hat{y}_{k+1|k}$, and $\hat{z}_{k+1|k}$. We exploit the structure of the linear system (25) in the same way that we did in Section 3.1. We discretize the integral in (23) with a left rectangle quadrature rule to obtain an expression for $P_{k+1|k} = P_k(t_{k+1})$:

$$P_{k+1|k} = \Phi_{xx}(t_{k+1}, t_k) \Lambda_k \Phi_{xx}(t_{k+1}, t_k)' \quad (26a)$$

$$\Lambda_k = P_{k|k} + \sigma(\hat{y}_{k|k}, u_k) \sigma(\hat{y}_{k|k}, u_k)' \Delta t_k \quad (26b)$$

5. DYNAMIC UV FLASH SEPARATION PROCESS

In this section, we consider a flash separation process where a mixture of N_C chemical components is separated into a vapor phase and a liquid phase. The vapor phase and the liquid phase are in thermodynamic equilibrium. A feed stream supplies the separator with a vapor-liquid mixture. The mixture exits the separator from a vapor stream and a liquid stream, and the separator is subject to external heating. The two main principles of the model of the separation process are 1) vapor-liquid equilibrium and 2) conservation of mass and energy.

5.1 Vapor-liquid equilibrium

The vapor phase (v) and the liquid phase (l) are in thermodynamic equilibrium. The second law of thermodynamics states that the entropy, S , of a closed system in equilibrium

is maximal. As we discuss later, the internal energy, U , and the total composition of the mixture (in moles), n , are given by conservation equations. The volume, V , of the separator is fixed. The above conditions constitute the UV flash optimization problem,

$$\max_{T, P, n^v, n^l} S = S^v(T, P, n^v) + S^l(T, P, n^l), \quad (27a)$$

$$\text{s.t.} \quad U^v(T, P, n^v) + U^l(T, P, n^l) = U, \quad (27b)$$

$$V^v(T, P, n^v) + V^l(T, P, n^l) = V, \quad (27c)$$

$$n_i^v + n_i^l = n_i, \quad i = 1, \dots, N_C. \quad (27d)$$

The solution to the UV flash optimization problem is the temperature, T , pressure, P , and vapor-liquid composition (in moles), n^v and n^l , that maximize entropy while satisfying the constraints on the internal energy, U , volume, V , and total composition, n . The UV flash is sometimes called the UVn flash or the isoenergetic-isochoric (constant energy - constant volume) flash. The UV flash optimization problem (27) is in the form

$$\min_y f(y), \quad (28a)$$

$$\text{s.t.} \quad g(y) = x, \quad (28b)$$

$$h(y) = 0, \quad (28c)$$

where the states are $x = [U; n]$, and the algebraic variables are $y = [T; P; n^v; n^l]$. Because (28) does not contain inequality constraints, the first-order optimality conditions are a set of algebraic equations:

$$G(x, y, z) = 0. \quad (29)$$

z are Lagrange multipliers.

5.2 Conservation of mass and energy

The internal energy, U , and the total composition (in moles), n , of the mixture are conserved. The deterministic conservation equations are

$$\dot{U}(t) = H_F^v(t) + H_F^l(t) - H_V(t) - H_L(t) + Q(t), \quad (30a)$$

$$\dot{n}_i(t) = f_{F,i}^v(t) + f_{F,i}^l(t) - v_i(t) - l_i(t), \quad i = 1, \dots, N_C. \quad (30b)$$

H_F^v and H_F^l are the vapor-liquid enthalpies of the feed stream, and H_V and H_L are the enthalpies of the vapor and liquid streams. Similarly, $f_{F,i}^v$ and $f_{F,i}^l$ are the vapor-liquid flow rates of the feed stream, and v_i and l_i are the flow rates of the vapor stream and the liquid stream. Q is the heat flux from the external heating. The deterministic conservation equations (30) are in the form

$$\dot{x}(t) = F(y(t), u(t)), \quad (31)$$

where u are inputs. Adding process noise to (31) results in the stochastic differential equations (1b).

5.3 Thermodynamic model

Because the vapor-liquid equilibrium conditions are the first-order optimality conditions of an optimization problem, they contain first-order derivatives of thermodynamic functions. We therefore need to evaluate second-order derivatives of thermodynamic functions in order to evaluate the Jacobian matrices of the equilibrium conditions. We use the open-source thermodynamic library, ThermoLib, developed by Ritschel et al. (2017b, 2016) for that purpose. It implements a thermodynamic model

based on the DIPPR database (Thomson, 1996) and cubic equations of state. It provides routines that evaluate enthalpy, $H = H(T, P, n)$, entropy, $S = S(T, P, n)$, and volume, $V = V(T, P, n)$, as well as first- and second-order derivatives. Given H , S , and V , the internal energy is $U = H - PV$, Gibbs energy is $G = H - TS$, and Helmholtz energy is $A = U - TS$. ThermoLib is available from www.psetools.org.

6. NUMERICAL EXAMPLE

We use the extended Kalman filter as a soft sensor that can estimate the vapor-liquid composition of a mixture based on temperature and pressure measurements. Soft sensing of compositions is an economical alternative to physical sensors which can be slow, expensive, and lack accuracy. We consider the separation of a mixture of 60% C_1 , 8% C_2 , 5% C_3 , 25% n- C_7 , and 2% CO_2 in a 0.2 m³ separator. We consider a 72 h time interval and a sampling time of 5 min. We assume that the inputs are known. The tank is cooled, i.e. $Q \leq 0$. Q increases from -9 MJ/h to -4 MJ/h after 24 h. The feed flow rate is constant at 1000 mol/h, and the vapor-liquid stream flow rates are constant at 400 mol/h and 600 mol/h. The standard deviations of the temperature and pressure measurement noise are 10 K and $10^{-1/2} \approx 0.3$ MPa. We consider a constant diffusion coefficient, i.e. $\sigma(y(t), u(t)) = \sigma = \text{diag}([\sigma_U; \sigma_{C_1}; \sigma_{C_2}; \sigma_{C_3}; \sigma_{n-C_7}; \sigma_{CO_2}])$, where $\sigma_U = 1$ MJ, $\sigma_{C_1} = \sigma_{C_2} = \sigma_{n-C_7} = 1$ mol, and $\sigma_{C_3} = \sigma_{CO_2} = 0.1$ mol. The mean of the initial states, x_0 , is a steady-state of the deterministic system (30), and the covariance of the initial states is $P_0 = \sigma\sigma'$.

Fig. 1 shows the filtered estimates of the total composition (in moles), internal energy, temperature, and pressure. It also shows the deviation of the estimates from the corresponding true simulated quantities and the root-mean-square deviation (RMSD). The RMSD of the i 'th state variable is

$$\text{RMSD}_i = \left(\frac{1}{N+1} \sum_{k=0}^N (\hat{x}_{i,k|k} - x_{i,k})^2 \right)^{1/2}, \quad (32)$$

where we compute the simulated states, x_k , as described in Section 3. $N+1 = 865$ is the number of measurement samples. The RMSD of the temperature and pressure estimates are computed similarly. We see that the deviations of the estimates are small compared to the scales of the estimates. Fig. 2 shows the filtered estimates of the total mole fractions, the vapor-liquid mole fractions, and the vapor fraction. Such estimates are necessary in model predictive control of processes with constraints on the purity (i.e. mole fractions) of the output streams, e.g. separation and distillation processes.

7. CONCLUSIONS

We present an extended Kalman filter for state estimation of UV flash processes. It is natural to model such processes, as well as other phase equilibrium processes, with DAEs that are in a semi-explicit index-1 form. We describe a model of a UV flash separation process, and demonstrate that it is in the semi-explicit index-1 DAE form. Finally, we demonstrate the accuracy of the extended Kalman

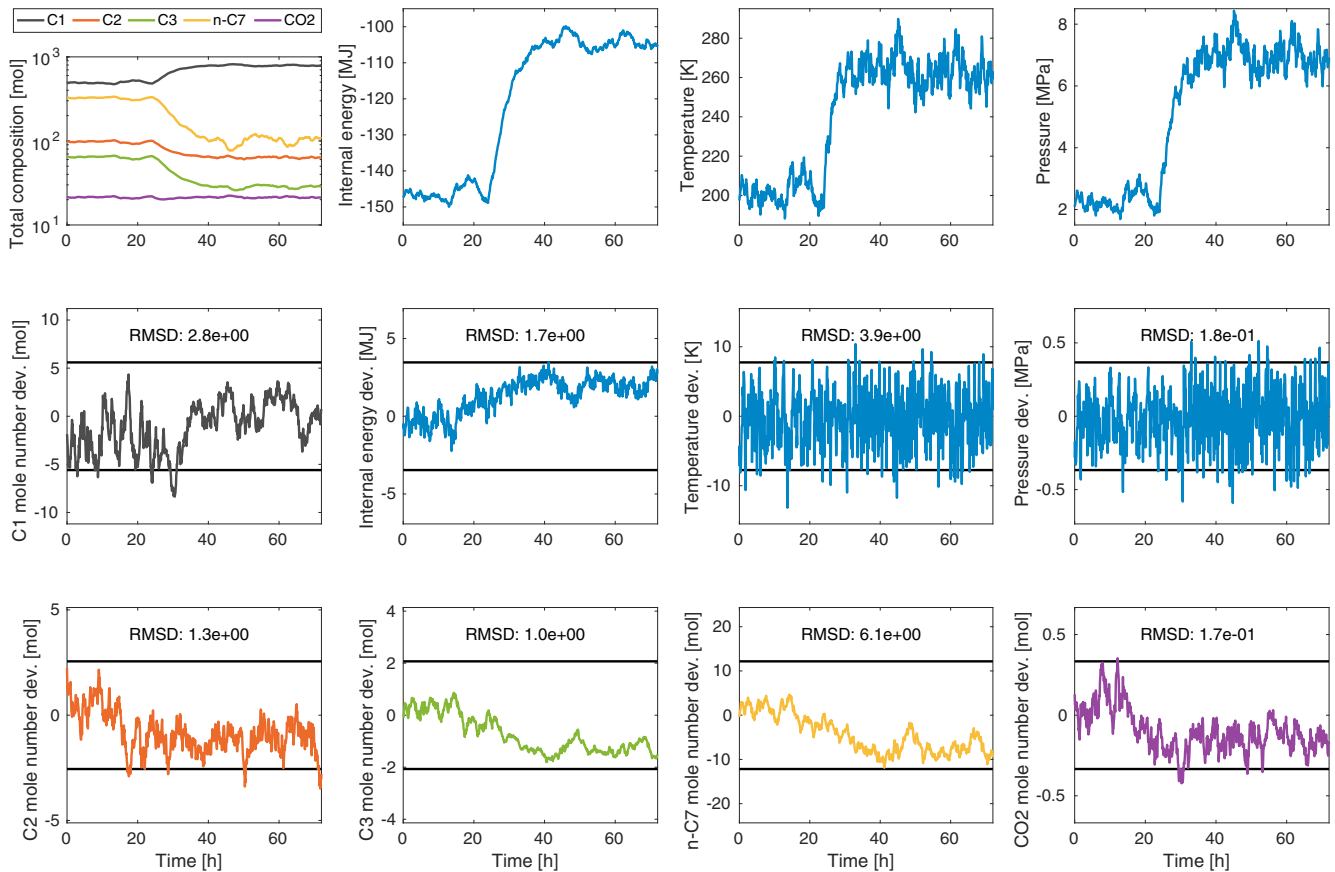


Fig. 1. Top row: filtered estimates of total composition, internal energy, temperature, and pressure. Middle and bottom rows: deviation (dev.) of estimates from the simulated (i.e. true) separation process. The black horizontal lines are \pm two times the RMSD of the estimates.

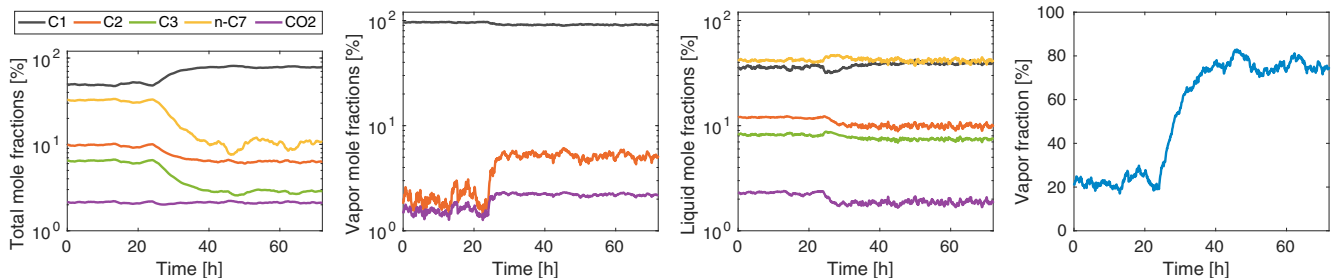


Fig. 2. Estimates of the total mole fractions, vapor mole fractions, liquid mole fractions, and the vapor fraction. We do not show the vapor mole fractions of C_3 and $n-C_7$ because they are very small, i.e. below 1%.

filter with a numerical example that involves soft sensing of vapor-liquid compositions based on temperature and pressure measurements.

REFERENCES

- Alessandri, A., Baglietto, M., Battistelli, G., and Zavala, V. (2010). Advances in moving horizon estimation for nonlinear systems. In *Proceedings of the 49th IEEE Conference on Decision and Control*. Atlanta, Georgia, USA.
- Arendsen, A.R.J. and Versteeg, G.F. (2009). Dynamic thermodynamics with internal energy, volume, and amount of moles as states: Application to liquefied gas tank. *Industrial & Engineering Chemistry Research*, 48(6), 3167–3176.
- Becerra, V.M., Roberts, P.D., and Griffiths, G.W. (2001). Applying the extended Kalman filter to systems described by nonlinear differential-algebraic equations. *Control Engineering Practice*, 9(3), 267–281.
- Binder, B.J.T., Pavlov, A., and Johansen, T.A. (2015). Estimation of flow rate and viscosity in a well with an electric submersible pump using moving horizon estimation. In *Proceedings of the 2nd IFAC Workshop on Automatic Control in Offshore Oil and Gas Production*, 140–146. Florianópolis, Brazil.
- Castier, M. (2010). Dynamic simulation of fluids in vessels via entropy maximization. *Journal of Industrial and Engineering Chemistry*, 16(1), 122–129.
- Evensen, G. (2009a). *Data Assimilation. The Ensemble Kalman Filter*. Springer-Verlag Berlin Heidelberg, 2nd edition.

- Evensen, G. (2009b). The ensemble Kalman filter for combined state and parameter estimation. *IEEE Control Systems*, 29(3), 83–104.
- Flatby, P., Skogestad, S., and Lundström, P. (1994). Rigorous dynamic simulation of distillation columns based on UV-flash. In *IFAC Symposium on Advanced Control of Chemical Processes (ADCHEM '94)*, 261–266.
- Gillijns, S., Mendoza, O.B., Chandrasekar, J., De Moor, B.L.R., Bernstein, D.S., and Ridley, A. (2006). What is the ensemble Kalman filter and how well does it work? In *Proceedings of the 2006 American Control Conference*, 4448–4453. Minneapolis, Minnesota, USA.
- Hammer, M. and Morin, A. (2014). A method for simulating two-phase pipe flow with real equations of state. *Computers & Fluids*, 100, 45–58.
- Haßkerl, D., Arshad, M., Hashemi, R., Subramanian, S., and Engell, S. (2016). Simulation study of the particle filter and the EKF for state estimation of a large-scale DAE-system with multi-rate sampling. In *Proceedings of the 11th IFAC symposium on Dynamics and Control of Process Systems Including Biosystems*, 490–495. Trondheim, Norway.
- Haßkerl, D., Subramanian, S., Hashemi, R., Arshad, M., and Engell, S. (2017). State estimation using a multi-rate particle filter for a reactive distillation column described by a DAE model. In *Proceedings of the 25th Mediterranean Conference on Control and Automation*, 876–881. Valletta, Malta.
- Jørgensen, J.B., Kristensen, M.R., Thomsen, P.G., and Madsen, H. (2007). New extended Kalman filter algorithms for stochastic differential algebraic equations. In *Assessment and Future Directions of Nonlinear Model Predictive Control*, volume 358 of *Lecture Notes in Control and Information Sciences*, 359–366. Springer-Verlag Berlin Heidelberg.
- Kataria, G., Singh, K., and Dohare, R.K. (2016). ANN based soft sensor model for reactive distillation column. *International Journal of Advanced Technology and Engineering Exploration*, 3(24), 182–186.
- Lima, E.R.A., Castier, M., and Biscaia Jr., E.C. (2008). Differential-algebraic approach to dynamic simulations of flash drums with rigorous evaluation of physical properties. *Oil & Gas Science and Technology*, 63(5), 677–686.
- Mandela, R.K., Rengaswamy, R., and Narasimhan, S. (2009). Nonlinear state estimation of differential algebraic systems. In *Proceedings of the 7th IFAC Symposium on Advanced Control of Chemical Processes*, 792–797.
- Mandela, R.K., Rengaswamy, R., Narasimhan, S., and Sridhar, L.N. (2010). Recursive state estimation techniques for nonlinear differential algebraic systems. *Chemical Engineering Science*, 65(16), 4548–4556.
- Michelsen, M.L. (1999). State function based flash specifications. *Fluid Phase Equilibria*, 158-160, 617–626.
- Mobed, P., Munusamy, S., Bhattacharyya, D., and Rengaswamy, R. (2016). State and parameter estimation in distributed constrained systems. 1. extended Kalman filtering of a special class of differential-algebraic equation systems. *Industrial & Engineering Chemistry Research*, 56(1), 206–215.
- Nikoofard, A., Aarsnes, U.J.F., Johansen, T.A., and Kaasa, G.O. (2017). State and parameter estimation of a drift-flux model for underbalanced drilling operations. *IEEE Transactions on Control Systems Technology*, 25(6), 2000–2009.
- Oliver, D.S. and Chen, Y. (2011). Recent progress on reservoir history matching: a review. *Computational Geosciences*, 15(1), 185–221.
- Pan, S., Su, H., Li, P., and Gu, Y. (2009). State estimation for batch distillation operations with a novel extended Kalman filter approach. In *Proceedings of the Joint 48th Conference on Decision and Control and 28th Chinese Control Conference*, 1884–1889. Shanghai, China.
- Pastorino, R., Richiedei, D., Cuadrado, J., and Trevisani, A. (2013). State estimation using multibody models and non-linear Kalman filters. *International Journal of Non-Linear Mechanics*, 53, 83–90.
- Puranik, Y., Bavdekar, V.A., Patwardhan, S.C., and Shah, S.L. (2012). An ensemble Kalman filter for systems governed by differential algebraic equations (DAEs). In *Proceedings of the 8th IFAC Symposium on Advanced Control of Chemical Processes*, 531–536. Singapore.
- Purohit, J.L., Patwardhan, S.C., and Mahajani, S.M. (2013). DAE-EKF-based nonlinear predictive control of reactive distillation systems exhibiting input and output multiplicities. *Industrial & Engineering Chemistry Research*, 52(38), 13699–13716.
- Purohit, J.L., Patwardhan, S.C., and Mahajani, S.M. (2015). Performance evaluation of bayesian state estimators for nonlinear DAE systems using a moderately high dimensional reactive distillation column model. In *Proceedings of the 12th International Symposium on Process Systems Engineering and 25th European Symposium on Computer Aided Process Engineering*, 1763–1768. Copenhagen, Denmark.
- Qiu, L., Wang, Y., and Reitz, R.D. (2014). Multiphase dynamic flash simulations using entropy maximization and application to compressible flow with phase change. *AIChE Journal*, 60(8), 3013–3024.
- Ritschel, T.K.S., Capolei, A., Gaspar, J., and Jørgensen, J.B. (2017a). An algorithm for gradient-based dynamic optimization of UV flash processes. *Computers and Chemical Engineering*. In Press. DOI: <https://doi.org/10.1016/j.compchemeng.2017.10.007>.
- Ritschel, T.K.S., Gaspar, J., and Jørgensen, J.B. (2017b). A thermodynamic library for simulation and optimization of dynamic processes. In *Proceedings of the 20th World Congress of the International Federation of Automatic Control*.
- Ritschel, T.K.S., Gaspar, J., Capolei, A., and Jørgensen, J.B. (2016). An open-source thermodynamic software library. Technical Report DTU Compute Technical Report-2016-12, Department of Applied Mathematics and Computer Science, Technical University of Denmark.
- Simon, D. (2006). *Optimal State Estimation: Kalman, H Infinity, and Nonlinear Approaches*. John Wiley & Sons.
- Talebi, H.A., Abdollahi, F., Patel, R.V., and Khorasani, K. (2010). *Neural Network-Based State Estimation of Nonlinear Systems. Application to Fault Detection and Isolation*, volume 395 of *Lecture Notes in Control and Information Sciences*. Springer-Verlag New York.
- Thomson, G.H. (1996). The DIPPR® databases. *International Journal of Thermophysics*, 17(1), 223–232.

Paper IX

Nonlinear Filters for State Estimation of UV Flash Processes

Authors:

Tobias K. S. Ritschel and John Bagterp Jørgensen

Published in:

Proceedings of the 2nd IEEE Conference on Control Technology and Applications (CCTA), August 21-24, 2018, Copenhagen, Denmark.

Nonlinear Filters for State Estimation of UV Flash Processes

Tobias K. S. Ritschel and John Bagterp Jørgensen

Abstract— We describe four algorithms for state estimation of stochastic differential-algebraic equations. We consider the extended Kalman filter, the unscented Kalman filter, the particle filter, and the ensemble Kalman filter. The differential-algebraic equations that we consider are in a semi-explicit index-1 form. Models of dynamic UV flash processes are in such a form. The UV flash is relevant to rigorous models of many chemical phase equilibrium processes because it is a mathematical representation of the second law of thermodynamics. State estimation is relevant to model predictive control, model identification, fault detection, monitoring, and prediction. State estimation of UV flash processes is therefore important to safe and economical operation of processes such as flash separation, distillation, multiphase flow in pipelines, and oil production. We compare the accuracy and efficiency of the four filters using a numerical example that involves a UV flash separation process. Furthermore, we demonstrate that the filters can be used as soft sensors that estimate the vapor-liquid composition of the separation process based on temperature and pressure measurements.

I. INTRODUCTION

State estimation is concerned with the reconstruction of state variables based on measurements and a model of the relevant process. State estimation is important to model predictive control, model identification, monitoring, prediction, and fault detection of chemical processes [1]. State estimation has been applied for many chemical processes including stirred tank reactors [2]–[4], batch reactors [5]–[8], plug-flow reactors [9], [10], fermentation [11], [12], distillation columns [13]–[15], oil and gas flow in pipes [16], and oil production [17]. Many chemical processes involve thermodynamic equilibrium between fluid phases. The phase equilibrium conditions are derived from the second law of thermodynamics, i.e. the entropy of a closed system is maximal when it is at equilibrium. The UV flash is a key component in rigorous models of dynamic phase equilibrium processes. The UV flash has been used to model flash separation [18]–[20], distillation [21], and computational fluid dynamical processes [22], [23]. It is possible to formulate the UV flash problem as an equality-constrained optimization problem [24]. The optimization variables are temperature, pressure, and vapor-liquid composition (in moles). The optimization problem involves constraints on the internal energy, U , the volume, V , and the total amount of moles of each chemical component, n . The solution to the optimization problem maximizes entropy while satisfying the equality constraints.

The corresponding phase equilibrium conditions are the first-order optimality conditions of the optimization problem. Therefore, the phase equilibrium conditions are a set of algebraic equations. Consequently, it is natural to model dynamic phase equilibrium processes with differential-algebraic equations (DAEs). Dynamic optimization algorithms for UV flash processes have recently been developed [25], but state estimation in such systems has not been addressed yet.

Many processes are nonlinear. There exist a number of state estimation algorithms (filters) for nonlinear systems, e.g. the extended Kalman filter (EKF), the unscented Kalman filter (UKF), the particle filter (PF), and the ensemble Kalman filter (EnKF) [26]. The EKF linearizes the nonlinear model and applies the original Kalman filter equations. This linearization can cause the EKF to be imprecise for highly nonlinear systems. The UKF uses deterministic samples to improve the accuracy compared to the EKF. However, the UKF can also suffer from limited accuracy for severely nonlinear systems. The PF uses a set of random samples to approximate the distribution of the states. It can therefore be more precise than the EKF and the UKF. The number of samples in the UKF is fixed whereas the number of samples in the PF is a tuning parameter. The EnKF is a specific particle filter that uses the Kalman filter equations. It has gained attention in oceanography and oil reservoir characterization [27]–[29] where large-scale models are common. There exist alternatives to the above filters, e.g. moving-horizon estimation [30], and algorithms based on neural networks [31]. State estimation algorithms were originally developed for stochastic difference and stochastic differential equations. However, it is natural to model many processes with DAEs. That is because algebraic equations often arise when a fast process is approximated as a quasi-steady-state process. For instance, it is common to assume that systems reach thermodynamic phase equilibrium instantaneously. Recently, authors have developed the EKF [32]–[34], the UKF [35]–[39], PFs [40], [41], and the EnKF [42] for DAE models.

In this work, we present the EKF, the UKF, the PF, and the EnKF for state estimation of dynamic UV flash processes. We model the UV flash processes with semi-explicit index-1 stochastic DAEs. We compare the accuracy of the four filters with a numerical example that involves flash separation of a hydrocarbon mixture. In the example, the states are estimated based on temperature and pressure measurements. Furthermore, we demonstrate that such state estimates can be used for soft-sensing of the vapor-liquid composition of the mixture.

This paper is structured as follows. We describe the semi-explicit index-1 stochastic DAE form that we consider in

*This work is funded by Innovation Fund Denmark in the OPTION project (63-2013-3). Tobias K. S. Ritschel and John Bagterp Jørgensen are with the Department of Applied Mathematics and Computer Science & the Center for Energy Resources Engineering (CERE), Technical University of Denmark, DK-2800 Kgs. Lyngby, Denmark {tobk, jbjjo}@dtu.dk

Section II. In Section III, we describe the numerical solution of stochastic DAEs in such a form. In Section IV, we describe the EKF, and in Section V, we describe the UKF. We describe the PF in Section VI and the EnKF in Section VII. In Section VIII we describe the model of the UV flash separation process, and in Section IX, we present the numerical example. We present conclusions in Section X.

II. STOCHASTIC SEMI-EXPLICIT INDEX-1 DAE SYSTEMS

In this work, we consider stochastic DAEs in the form

$$G(\mathbf{x}(t), \mathbf{y}(t), \mathbf{z}(t)) = 0, \quad (1a)$$

$$d\mathbf{x}(t) = F(\mathbf{y}(t), u(t))dt + \sigma(\mathbf{y}(t), u(t))d\boldsymbol{\omega}(t). \quad (1b)$$

$\mathbf{x}(t)$ is a vector of state variables, $\mathbf{y}(t)$ is a vector of algebraic variables, and $\mathbf{z}(t)$ is a vector of adjoint algebraic variables. The algebraic equations (1a) are formulated such that they can represent phase equilibrium conditions. The stochastic differential equations (1b) are formulated such that they can represent conservation equations. We assume knowledge of the manipulated inputs, $u(t)$, and the initial distribution of the states, $\mathbf{x}(t_0) \sim N(x_0, P_0)$. $\boldsymbol{\omega}(t)$ is a standard Wiener process, i.e. its incremental covariance is $\text{Id}t$. It is possible to solve the algebraic equations (1a) for $\mathbf{y}(t)$ and $\mathbf{z}(t)$ when $\mathbf{x}(t)$ is specified. We obtain measurements, $\mathbf{y}^m(t_k)$, of the outputs, $\mathbf{z}^m(t_k)$, at discrete times, t_k :

$$\mathbf{z}^m(t_k) = H(\mathbf{y}(t_k)), \quad (2a)$$

$$\mathbf{y}^m(t_k) = \mathbf{z}^m(t_k) + \mathbf{v}(t_k). \quad (2b)$$

The measurement noise, $\mathbf{v}_k = \mathbf{v}(t_k)$, follows a normal distribution, i.e. $\mathbf{v}_k \sim N(0, T_k)$.

III. NUMERICAL SIMULATION

In order to solve the stochastic DAE (1), we discretize the stochastic differential equations with a semi-implicit scheme. We discretize the deterministic and stochastic part with Euler's implicit and explicit method, respectively. We split the time interval between the k 'th and the $k+1$ 'th measurement into N_k time steps. For each time step, we solve the residual equations, $R_{k,n+1} = 0$, for $w_{k,n+1} = [x_{k,n+1}; y_{k,n+1}; z_{k,n+1}]$ where the residual function is

$$\begin{aligned} R_{k,n+1} &= R_{k,n+1}(w_{k,n+1}) \\ &= R_{k,n+1}(x_{k,n+1}, y_{k,n+1}, z_{k,n+1}) \\ &= R_{k,n+1}(x_{k,n+1}, y_{k,n+1}, z_{k,n+1}; x_{k,n}, y_{k,n}, u_k) \\ &= \begin{bmatrix} D_{k,n+1}(x_{k,n+1}, y_{k,n+1}; x_{k,n}, y_{k,n}, u_k) \\ G(x_{k,n+1}, y_{k,n+1}, z_{k,n+1}) \end{bmatrix}, \quad (3) \end{aligned}$$

and

$$\begin{aligned} D_{k,n+1} &= D_{k,n+1}(x_{k,n+1}, y_{k,n+1}; x_{k,n}, y_{k,n}, u_k) \\ &= x_{k,n+1} - F(y_{k,n+1}, u_k)\Delta t_{k,n} \\ &\quad - \sigma(y_{k,n}, u_k)\Delta\omega_{k,n} - x_{k,n}. \quad (4) \end{aligned}$$

The increments, $\Delta\omega_{k,n}$, are sampled from $N(0, \text{I}\Delta t_{k,n})$. We use Newton's method to solve the residual equations:

$$w_{k,n+1}^{l+1} = w_{k,n+1}^l + \Delta w_{k,n+1}^l. \quad (5)$$

We compute the Newton step by solving

$$M\Delta w_{k,n+1}^l = -R_{k,n+1}(w_{k,n+1}^l). \quad (6)$$

The iteration matrix is

$$M \approx \frac{\partial R_{k,n+1}}{\partial w_{k,n+1}} = \begin{bmatrix} \text{I} & -\frac{\partial F}{\partial y}\Delta t_{k,n} & 0 \\ \frac{\partial G}{\partial x} & \frac{\partial G}{\partial y} & \frac{\partial G}{\partial z} \end{bmatrix}. \quad (7)$$

A. Efficient solution of the linear system

We exploit the structure of the Jacobian matrix in (7) to solve the linear system (6) efficiently. We compute $\Delta x_{k,n+1}^l$ directly by

$$\Delta x_{k,n+1}^l = \left(\frac{\partial F}{\partial y}\Delta t_{k,n} \right) \Delta y_{k,n+1}^l - D_{k,n+1}, \quad (8)$$

and we compute $\Delta y_{k,n+1}^l$ and $\Delta z_{k,n+1}^l$ by solving the reduced linear system,

$$\bar{M} \begin{bmatrix} \Delta y_{k,n+1}^l \\ \Delta z_{k,n+1}^l \end{bmatrix} = \frac{\partial G}{\partial x} D_{k,n+1} - G(x_{k,n+1}^l, y_{k,n+1}^l, z_{k,n+1}^l). \quad (9)$$

The reduced iteration matrix is

$$\bar{M} \approx \begin{bmatrix} \frac{\partial G}{\partial y} + \frac{\partial G}{\partial x} \frac{\partial F}{\partial y} \Delta t_{k,n} & \frac{\partial G}{\partial z} \end{bmatrix}. \quad (10)$$

\bar{M} is smaller than M and is therefore cheaper to factorize.

IV. THE EXTENDED KALMAN FILTER

In this section, we describe the extended Kalman filter. We initialize the filter with the mean and covariance of the initial states:

$$\hat{x}_{0|-1} = x_0, \quad (11a)$$

$$P_{0|-1} = P_0. \quad (11b)$$

The initial estimates of the algebraic and adjoint algebraic variables satisfy the algebraic equations:

$$G(\hat{x}_{0|-1}, \hat{y}_{0|-1}, \hat{z}_{0|-1}) = 0. \quad (12)$$

A. Measurement-update

The one-step ahead prediction of the outputs, the measurements, and the covariance matrix are

$$\hat{z}_{k|k-1}^m = H(\hat{y}_{k|k-1}), \quad (13a)$$

$$\hat{y}_{k|k-1}^m = \hat{z}_{k|k-1}^m, \quad (13b)$$

$$T_{k|k-1} = C_k P_{k|k-1} C_k' + T_k, \quad (13c)$$

where T_k is the measurement noise covariance matrix, and

$$\begin{aligned} C_k &= \frac{\partial H}{\partial x}(\hat{y}_{k|k-1}) \\ &= \frac{\partial H}{\partial y}(\hat{y}_{k|k-1}) \frac{\partial \hat{y}_{k|k-1}}{\partial \hat{x}_{k|k-1}}. \quad (14) \end{aligned}$$

We compute the sensitivities of the algebraic and adjoint algebraic variables by solving

$$\begin{bmatrix} \frac{\partial G}{\partial y} & \frac{\partial G}{\partial z} \end{bmatrix} \begin{bmatrix} \frac{\partial \hat{y}_{k|k-1}}{\partial \hat{x}_{k|k-1}} \\ \frac{\partial \hat{z}_{k|k-1}}{\partial \hat{x}_{k|k-1}} \end{bmatrix} = -\frac{\partial G}{\partial x}. \quad (15)$$

The innovation is

$$e_k = y_k^m - \hat{y}_{k|k-1}^m, \quad (16)$$

and the Kalman filter gain matrix is

$$K_{f,x,k} = P_{k|k-1} C_k' T_{k|k-1}^{-1}. \quad (17)$$

We compute the filtered state and its approximate covariance matrix by

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_{f,x,k} e_k, \quad (18a)$$

$$P_{k|k} = P_{k|k-1} - K_{f,x,k} T_{k|k-1} K_{f,x,k}'. \quad (18b)$$

The corresponding estimates of the algebraic and adjoint algebraic variables satisfy

$$G(\hat{x}_{k|k}, \hat{y}_{k|k}, \hat{z}_{k|k}) = 0. \quad (19)$$

We compute the corresponding covariance matrices by

$$P_{y,k|k} = \Phi_{yx}(t_k, t_k) P_{k|k} \Phi_{yx}(t_k, t_k)', \quad (20a)$$

$$P_{z,k|k} = \Phi_{zx}(t_k, t_k) P_{k|k} \Phi_{zx}(t_k, t_k)'. \quad (20b)$$

We compute the sensitivities, $\Phi_{yx}(t_k, t_k) = \frac{\partial \hat{y}_{k|k}}{\partial \hat{x}_{k|k}}$ and $\Phi_{zx}(t_k, t_k) = \frac{\partial \hat{z}_{k|k}}{\partial \hat{x}_{k|k}}$, by solving

$$\begin{bmatrix} \frac{\partial G}{\partial y} & \frac{\partial G}{\partial z} \end{bmatrix} \begin{bmatrix} \Phi_{yx}(t_k, t_k) \\ \Phi_{zx}(t_k, t_k) \end{bmatrix} = -\frac{\partial G}{\partial x}. \quad (21)$$

B. Time-update

In between measurement k and $k+1$, we propagate the mean by solving the DAE,

$$\hat{x}_k(t_k) = \hat{x}_{k|k-1}, \quad (22a)$$

$$G(\hat{x}_k(t), \hat{y}_k(t), \hat{z}_k(t)) = 0, \quad (22b)$$

$$\frac{d\hat{x}_k(t)}{dt} = F(\hat{y}_k(t), u(t)), \quad (22c)$$

for $t \in]t_k; t_{k+1}[$. The sensitivities, $\Phi_{xx}(t, s) = \frac{\partial \hat{x}_k(t)}{\partial \hat{x}_k(s)}$, $\Phi_{yx}(t, s) = \frac{\partial \hat{y}_k(t)}{\partial \hat{x}_k(s)}$, and $\Phi_{zx}(t, s) = \frac{\partial \hat{z}_k(t)}{\partial \hat{x}_k(s)}$, satisfy

$$\Phi_{xx}(s, s) = \mathbf{I}, \quad (23a)$$

$$\frac{\partial G}{\partial x} \Phi_{xx}(t, s) + \frac{\partial G}{\partial y} \Phi_{yx}(t, s) + \frac{\partial G}{\partial z} \Phi_{zx}(t, s) = 0, \quad (23b)$$

$$\frac{d\Phi_{xx}(t, s)}{dt} = \frac{\partial F}{\partial y} \Phi_{yx}(t, s). \quad (23c)$$

The covariance matrix is given in terms of the sensitivities, [33]:

$$\begin{aligned} P_k(t) &= \Phi_{xx}(t, t_k) P_{k|k} \Phi_{xx}(t, t_k)' \\ &+ \int_{t_k}^t \Phi_{xx}(t, s) \sigma(\hat{y}_k(s), u(s)) \sigma(\hat{y}_k(s), u(s))' \Phi_{xx}(t, s)' ds. \end{aligned} \quad (24)$$

C. Numerical solution of the time-update equations

We discretize (22) with Euler's implicit method. For each of the N_k time steps, we solve the residual equations,

$$\begin{bmatrix} D_{k,n+1}(\hat{x}_{k,n+1}, \hat{y}_{k,n+1}; \hat{x}_{k,n}, u_k) \\ G(\hat{x}_{k,n+1}, \hat{y}_{k,n+1}, \hat{z}_{k,n+1}) \end{bmatrix} = 0, \quad (25)$$

where $D_{k,n+1} = D_{k,n+1}(\hat{x}_{k,n+1}, \hat{y}_{k,n+1}; \hat{x}_{k,n}, u_k)$ is

$$D_{k,n+1} = \hat{x}_{k,n+1} - F(\hat{y}_{k,n+1}, u_k) \Delta t_{k,n} - \hat{x}_{k,n}, \quad (26)$$

and $\hat{x}_{k,0} = \hat{x}_{k|k}$. We compute the sensitivities by solving

$$\begin{bmatrix} \mathbf{I} & -\frac{\partial F}{\partial y} \Delta t_{k,n} & 0 \\ \frac{\partial G}{\partial x} & \frac{\partial G}{\partial y} & \frac{\partial G}{\partial z} \end{bmatrix} \begin{bmatrix} \Phi_{xx}(t_{k,n+1}, t_{k,n}) \\ \Phi_{yx}(t_{k,n+1}, t_{k,n}) \\ \Phi_{zx}(t_{k,n+1}, t_{k,n}) \end{bmatrix} = \begin{bmatrix} \mathbf{I} \\ 0 \end{bmatrix}. \quad (27)$$

We exploit the structure of the system matrix in (27) as we described in Section III-A. We approximate the integral in (24) with a left rectangle rule:

$$P_{k,n+1} = \Phi_{xx}(t_{k,n+1}, t_{k,n}) \tau_{k,n} \Phi_{xx}(t_{k,n+1}, t_{k,n})', \quad (28a)$$

$$\tau_{k,n} = P_{k,n} + \sigma(\hat{y}_{k,n}, u_k) \sigma(\hat{y}_{k,n}, u_k)' \Delta t_{k,n}. \quad (28b)$$

$P_{k,0} = P_{k|k}$, and the one-step ahead estimates are $\hat{x}_{k+1|k} = \hat{x}_{k,N_k}$, $\hat{y}_{k+1|k} = \hat{y}_{k,N_k}$, and $\hat{z}_{k+1|k} = \hat{z}_{k,N_k}$. The covariance matrix is $P_{k+1|k} = P_{k,N_k}$.

V. THE UNSCENTED KALMAN FILTER

The initial state estimate and the covariance matrix are

$$\hat{x}_{0|-1} = x_0, \quad (29a)$$

$$P_{0|-1} = P_0. \quad (29b)$$

A. Measurement-update

We compute $2n_x + 1$ samples of the states by

$$\hat{x}_{k|k-1}^{(0)} = \hat{x}_{k|k-1}, \quad (30a)$$

$$\hat{x}_{k|k-1}^{(i)} = \hat{x}_{k|k-1} + \sqrt{c} \left(\sqrt{P_{k|k-1}} \right)_i, \quad (30b)$$

$$\hat{x}_{k|k-1}^{(i+n_x)} = \hat{x}_{k|k-1} - \sqrt{c} \left(\sqrt{P_{k|k-1}} \right)_i, \quad (30c)$$

for $i = 1, \dots, n_x$. n_x is the dimension of the states. $c = \alpha^2(n_x + \kappa)$, $\alpha \in]0; 1]$, and we set κ to zero. We compute $\sqrt{P_{k|k-1}}$ with a cholesky factorization, and $(\sqrt{P_{k|k-1}})_i$ is the i 'th column of $\sqrt{P_{k|k-1}}$ [43]. We introduce the weights,

$$W_m^{(0)} = \frac{\lambda}{n_x + \lambda}, \quad (31a)$$

$$W_c^{(0)} = \frac{\lambda}{n_x + \lambda} + (1 - \alpha^2 + \beta), \quad (31b)$$

$$W_m^{(i)} = \frac{1}{2(n_x + \lambda)}, \quad (31c)$$

$$W_c^{(i)} = \frac{1}{2(n_x + \lambda)}, \quad (31d)$$

for $i = 1, \dots, 2n_x$, where $\lambda = \alpha^2(n_x + \kappa) - n_x$ [44]. We solve the algebraic equations for each sample,

$$G(\hat{x}_{k|k-1}^{(i)}, \hat{y}_{k|k-1}^{(i)}, \hat{z}_{k|k-1}^{(i)}) = 0, \quad (32)$$

and evaluate the output:

$$\hat{z}_{k|k-1}^{m,(i)} = H(\hat{y}_{k|k-1}^{(i)}). \quad (33)$$

We compute the mean, covariance, and cross-covariance by

$$\hat{z}_{k|k-1}^m = \sum_{i=0}^{2n_x} W_m^{(i)} \hat{z}_{k|k-1}^{m,(i)}, \quad (34a)$$

$$T_{k|k-1} = \sum_{i=0}^{2n_x} W_c^{(i)} \left(\hat{z}_{k|k-1}^{m,(i)} - \hat{z}_{k|k-1}^m \right) \left(\hat{z}_{k|k-1}^{m,(i)} - \hat{z}_{k|k-1}^m \right)' + T_k, \quad (34b)$$

$$S_{k|k-1} = \sum_{i=0}^{2n_x} W_c^{(i)} \left(\hat{x}_{k|k-1}^{(i)} - \hat{x}_{k|k-1} \right) \left(\hat{z}_{k|k-1}^{m,(i)} - \hat{z}_{k|k-1}^m \right)'. \quad (34c)$$

The innovation is

$$e_k = y_k^m - \hat{y}_{k|k-1}^m, \quad (35)$$

where $\hat{y}_{k|k-1}^m = \hat{z}_{k|k-1}^m$. The Kalman filter gain matrix is

$$K_{f,x,k} = S_{k|k-1} T_{k|k-1}^{-1}. \quad (36)$$

The filtered state estimate and the covariance matrix are

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_{f,x,k} e_k, \quad (37a)$$

$$P_{k|k} = P_{k|k-1} - K_{f,x,k} T_{k|k-1} K_{f,x,k}', \quad (37b)$$

and we solve the algebraic equations for the estimates of the algebraic and adjoint algebraic variables:

$$G(\hat{x}_{k|k}, \hat{y}_{k|k}, \hat{z}_{k|k}) = 0. \quad (38)$$

B. Time-update

We introduce $\tilde{n} = n_x + n_\omega$ and the sets $\mathcal{N}_0 = \{0\}$, $\mathcal{N}_x = \{1, \dots, 2n_x\}$, and $\mathcal{N}_\omega = \{2n_x + 1, \dots, 2n_x + 2n_\omega\}$. n_ω is the dimension of the process noise. The process noise in (1) is non-additive. We therefore compute $2\tilde{n} + 1$ samples of the states:

$$\hat{x}_{k|k}^{(i)} = \hat{x}_{k|k}, \quad i \in \mathcal{N}_0 \cup \mathcal{N}_\omega, \quad (39a)$$

$$\hat{x}_{k|k}^{(i)} = \hat{x}_{k|k} + \sqrt{\tilde{c}} \left(\sqrt{P_{k|k}} \right)_i, \quad i = 1, \dots, n_x, \quad (39b)$$

$$\hat{x}_{k|k}^{(i+n_x)} = \hat{x}_{k|k} - \sqrt{\tilde{c}} \left(\sqrt{P_{k|k}} \right)_i, \quad i = 1, \dots, n_x. \quad (39c)$$

$\tilde{c} = \alpha^2 (\tilde{n} + \kappa)$, and we introduce the weights,

$$\tilde{W}_m^{(0)} = \frac{\tilde{\lambda}}{\tilde{n} + \tilde{\lambda}}, \quad (40a)$$

$$\tilde{W}_c^{(0)} = \frac{\tilde{\lambda}}{\tilde{n} + \tilde{\lambda}} + (1 - \alpha^2 + \beta), \quad (40b)$$

$$\tilde{W}_m^{(i)} = \frac{1}{2(\tilde{n} + \tilde{\lambda})}, \quad (40c)$$

$$\tilde{W}_c^{(i)} = \frac{1}{2(\tilde{n} + \tilde{\lambda})}, \quad (40d)$$

for $i = 1, \dots, 2\tilde{n}$. $\tilde{\lambda} = \alpha^2 (\tilde{n} + \kappa) - \tilde{n}$. We solve the DAEs,

$$\hat{x}_k^{(i)}(t_k) = \hat{x}_{k|k}^{(i)}, \quad (41a)$$

$$G(\hat{x}_k^{(i)}(t), \hat{y}_k^{(i)}(t), \hat{z}_k^{(i)}(t)) = 0, \quad (41b)$$

$$d\hat{x}_k^{(i)}(t) = F(\hat{y}_k^{(i)}(t), u(t))dt, \quad (41c)$$

for $i \in \mathcal{N}_0 \cup \mathcal{N}_x$ and $t \in]t_k; t_{k+1}]$. Furthermore, we solve

$$\hat{x}_k^{(i)}(t_k) = \hat{x}_{k|k}^{(i)}, \quad (42a)$$

$$G(\hat{x}_k^{(i)}(t), \hat{y}_k^{(i)}(t), \hat{z}_k^{(i)}(t)) = 0, \quad (42b)$$

$$d\hat{x}_k^{(i)}(t) = F(\hat{y}_k^{(i)}(t), u(t))dt + \sigma(\hat{y}_k^{(i)}(t), u(t))d\omega^{(i)}(t), \quad (42c)$$

for $i \in \mathcal{N}_\omega$ and $t \in]t_k; t_{k+1}]$. We sample the increments as

$$d\omega^{(i+2n_x)}(t) = \left(\sqrt{\tilde{c} dt} \right) e_i, \quad (43a)$$

$$d\omega^{(i+2n_x+n_\omega)}(t) = - \left(\sqrt{\tilde{c} dt} \right) e_i, \quad (43b)$$

for $i = 1, \dots, n_\omega$. The i 'th element of the vector e_i is one and all other elements are zero. We compute the state estimate and the covariance matrix by

$$\hat{x}_{k+1|k} = \sum_{i=0}^{2\tilde{n}} \tilde{W}_m^{(i)} \hat{x}_{k+1|k}^{(i)}, \quad (44a)$$

$$P_{k+1|k} = \sum_{i=0}^{2\tilde{n}} \tilde{W}_c^{(i)} \left(\hat{x}_{k+1|k}^{(i)} - \hat{x}_{k+1|k} \right) \left(\hat{x}_{k+1|k}^{(i)} - \hat{x}_{k+1|k} \right)', \quad (44b)$$

where $\hat{x}_{k+1|k}^{(i)} = \hat{x}_k^{(i)}(t_{k+1})$.

C. Numerical solution of the time-update equations

We discretize (41) with Euler's implicit method and (42) with the semi-implicit scheme described in Section III. For each of the N_k time steps, we solve

$$\begin{bmatrix} D_{k,n+1}^{(i)}(\hat{x}_{k,n+1}^{(i)}, \hat{y}_{k,n+1}^{(i)}; \hat{x}_{k,n}^{(i)}, \hat{y}_{k,n}^{(i)}, u_k) \\ G(\hat{x}_{k,n+1}^{(i)}, \hat{y}_{k,n+1}^{(i)}, \hat{z}_{k,n+1}^{(i)}) \end{bmatrix} = 0, \quad (45)$$

where $D_{k,n+1}^{(i)} = D_{k,n+1}^{(i)}(\hat{x}_{k,n+1}^{(i)}, \hat{y}_{k,n+1}^{(i)}; \hat{x}_{k,n}^{(i)}, \hat{y}_{k,n}^{(i)}, u_k)$ is

$$D_{k,n+1}^{(i)} = \hat{x}_{k,n+1}^{(i)} - F(\hat{y}_{k,n+1}^{(i)}, u_k) \Delta t_{k,n} - \hat{x}_{k,n}^{(i)}, \quad (46)$$

for $i \in \mathcal{N}_0 \cup \mathcal{N}_x$ and

$$D_{k,n+1}^{(i)} = \hat{x}_{k,n+1}^{(i)} - F(\hat{y}_{k,n+1}^{(i)}, u_k) \Delta t_{k,n} - \sigma(\hat{y}_{k,n}^{(i)}, u_k) \Delta \omega_{k,n}^{(i)} - \hat{x}_{k,n}^{(i)}, \quad (47)$$

for $i \in \mathcal{N}_\omega$. The increments are

$$\Delta \omega_{k,n}^{(i+2n_x)} = \left(\sqrt{\tilde{c} \Delta t_{k,n}} \right) e_i, \quad (48a)$$

$$\Delta \omega_{k,n}^{(i+2n_x+n_\omega)} = - \left(\sqrt{\tilde{c} \Delta t_{k,n}} \right) e_i, \quad (48b)$$

for $i = 1, \dots, n_\omega$.

VI. THE PARTICLE FILTER

We sample N_p particles, $\hat{x}_{0|-1}^{(i)}$, from the distribution of the initial states, i.e. from $N(x_0, P_0)$. Next, we solve the algebraic equations for each of the particles:

$$G(\hat{x}_{0|-1}^{(i)}, \hat{y}_{0|-1}^{(i)}, \hat{z}_{0|-1}^{(i)}) = 0. \quad (49)$$

A. Measurement-update

For each particle, we compute the output,

$$\hat{z}_{k|k-1}^{m,(i)} = H(\hat{y}_{k|k-1}^{(i)}), \quad (50)$$

and the difference between the output and the measurement:

$$e_k^{(i)} = y_k^m - \hat{z}_{k|k-1}^{m,(i)}. \quad (51)$$

We compute the relative likelihood that y_k^m is observed if the particle output, $\hat{z}_{k|k-1}^{m,(i)}$, is true:

$$\tilde{q}^{(i)} = \frac{1}{\sqrt{(2\pi)^{n_m} |T_k|}} \exp\left(-\frac{1}{2} (e_k^{(i)})' T_k^{-1} e_k^{(i)}\right). \quad (52)$$

n_m is the dimension of the output, and $|T_k|$ is the determinant of T_k . We normalize the relative likelihoods:

$$q^{(i)} = \frac{\tilde{q}^{(i)}}{\sum_{j=1}^{N_p} \tilde{q}^{(j)}}. \quad (53)$$

We use systematic resampling [45], [46]. We sample a single (scalar) uniformly distributed number, $\tilde{p} \sim U([0, 1])$. Next we compute:

$$p^{(i)} = ((i-1) + \tilde{p})/N_p, \quad i = 1, \dots, N_p. \quad (54)$$

The resampled particles, $\{\hat{x}_{k|k}^{(i)}\}_{i=1}^{N_p}$, contain $m^{(i)}$ copies of $\hat{x}_{k|k-1}^{(i)}$ where $m^{(i)}$ is the number of indices, l , for which $p^{(l)}$ is in the interval $]\sum_{j=1}^{i-1} q^{(j)}; \sum_{j=1}^i q^{(j)}]$. We compute the state estimate and the covariance matrix by

$$\hat{x}_{k|k} = W_m \sum_{i=1}^{N_p} \hat{x}_{k|k}^{(i)}, \quad (55a)$$

$$P_{k|k} = W_c \sum_{i=1}^{N_p} (\hat{x}_{k|k}^{(i)} - \hat{x}_{k|k}) (\hat{x}_{k|k}^{(i)} - \hat{x}_{k|k})', \quad (55b)$$

where $W_m = 1/N_p$ and $W_c = 1/(N_p - 1)$. Next, we solve the algebraic equations:

$$G(\hat{x}_{k|k}, \hat{y}_{k|k}, \hat{z}_{k|k}) = 0. \quad (56)$$

B. Time-update

For each particle, we solve the stochastic DAE,

$$\hat{\mathbf{x}}_k^{(i)}(t_k) = \hat{\mathbf{x}}_{k|k}^{(i)}, \quad (57a)$$

$$G(\hat{\mathbf{x}}_k^{(i)}(t), \hat{\mathbf{y}}_k^{(i)}(t), \hat{\mathbf{z}}_k^{(i)}(t)) = 0, \quad (57b)$$

$$d\hat{\mathbf{x}}_k^{(i)}(t) = F(\hat{\mathbf{y}}_k^{(i)}(t), u(t))dt + \sigma(\hat{\mathbf{y}}_k^{(i)}(t), u(t))d\boldsymbol{\omega}(t), \quad (57c)$$

for $t \in]t_k; t_{k+1}]$ as we described in Section III. The one-step ahead predictions for the i 'th particle are $\hat{x}_{k+1|k}^{(i)} = \hat{x}_{k,N_k}^{(i)}$, $\hat{y}_{k+1|k}^{(i)} = \hat{y}_{k,N_k}^{(i)}$, and $\hat{z}_{k+1|k}^{(i)} = \hat{z}_{k,N_k}^{(i)}$.

VII. THE ENSEMBLE KALMAN FILTER

We sample N_p particles, $\hat{x}_{0|-1}^{(i)}$, from $N(x_0, P_0)$, and solve the algebraic equations for each particle:

$$G(\hat{x}_{0|-1}^{(i)}, \hat{y}_{0|-1}^{(i)}, \hat{z}_{0|-1}^{(i)}) = 0. \quad (58)$$

A. Measurement-update

We compute the output for each particle:

$$\hat{z}_{k|k-1}^{m,(i)} = H(\hat{y}_{k|k-1}^{(i)}). \quad (59)$$

We compute the state and output means, the covariance, and the cross-covariance:

$$\hat{x}_{k|k-1} = W_m \sum_{i=1}^{N_p} \hat{x}_{k|k-1}^{(i)}, \quad (60a)$$

$$\hat{z}_{k|k-1}^m = W_m \sum_{i=1}^{N_p} \hat{z}_{k|k-1}^{m,(i)}, \quad (60b)$$

$$T_{k|k-1} = W_c \sum_{i=1}^{N_p} (\hat{z}_{k|k-1}^{m,(i)} - \hat{z}_{k|k-1}^m) (\hat{z}_{k|k-1}^{m,(i)} - \hat{z}_{k|k-1}^m)' + T_k, \quad (60c)$$

$$S_{k|k-1} = W_c \sum_{i=1}^{N_p} (\hat{x}_{k|k-1}^{(i)} - \hat{x}_{k|k-1}) (\hat{z}_{k|k-1}^{m,(i)} - \hat{z}_{k|k-1}^m)'. \quad (60d)$$

$W_m = 1/N_p$ and $W_c = 1/(N_p - 1)$. We sample measurements for each particle:

$$\hat{y}_{k|k-1}^{m,(i)} = \hat{z}_{k|k-1}^{m,(i)} + v_k^{(i)}. \quad (61)$$

Each of the measurement noise samples, $v_k^{(i)}$, is drawn from $N(0, T_k)$. The innovation for the i 'th particle is

$$e_k^{(i)} = y_k^m - \hat{y}_{k|k-1}^{m,(i)}, \quad (62)$$

and the Kalman filter gain matrix is

$$K_{fx,k} = S_{k|k-1} T_{k|k-1}^{-1}. \quad (63)$$

For each particle, we update the states:

$$\hat{x}_{k|k}^{(i)} = \hat{x}_{k|k-1}^{(i)} + K_{fx,k} e_k^{(i)}. \quad (64)$$

The state estimate and covariance matrix are

$$\hat{x}_{k|k} = W_m \sum_{i=1}^{N_p} \hat{x}_{k|k}^{(i)}, \quad (65a)$$

$$P_{k|k} = W_c \sum_{i=1}^{N_p} (\hat{x}_{k|k}^{(i)} - \hat{x}_{k|k}) (\hat{x}_{k|k}^{(i)} - \hat{x}_{k|k})', \quad (65b)$$

and we compute the estimates of the algebraic and adjoint algebraic variables by solving the algebraic equations,

$$G(\hat{x}_{k|k}, \hat{y}_{k|k}, \hat{z}_{k|k}) = 0. \quad (66)$$

B. Time-update

The time-update in the EnKF is identical to the time-update in the PF. We solve the stochastic DAEs,

$$\hat{\mathbf{x}}_k^{(i)}(t_k) = \hat{\mathbf{x}}_{k|k}^{(i)}, \quad (67a)$$

$$G(\hat{\mathbf{x}}_k^{(i)}(t), \hat{\mathbf{y}}_k^{(i)}(t), \hat{\mathbf{z}}_k^{(i)}(t)) = 0, \quad (67b)$$

$$d\hat{\mathbf{x}}_k^{(i)}(t) = F(\hat{\mathbf{y}}_k^{(i)}(t), u(t))dt + \sigma(\hat{\mathbf{y}}_k^{(i)}(t), u(t))d\boldsymbol{\omega}(t), \quad (67c)$$

for $i = 1, \dots, N_p$ and $t \in]t_k; t_{k+1}]$ with the approach described in Section III. The one-step ahead predictions are $\hat{x}_{k+1|k}^{(i)} = \hat{x}_{k,N_k}^{(i)}$, $\hat{y}_{k+1|k}^{(i)} = \hat{y}_{k,N_k}^{(i)}$, and $\hat{z}_{k+1|k}^{(i)} = \hat{z}_{k,N_k}^{(i)}$.

VIII. THE DYNAMIC UV FLASH SEPARATION PROCESS

We consider the separation of a mixture of N_C components. The mixture is separated into a vapor phase (v) and a liquid phase (l). The two phases are in thermodynamic equilibrium. The separator is supplied by a feed stream. The vapor and liquid phases exit the separator from two separate streams. Furthermore, the unit is either heated or cooled. We model the process with 1) vapor-liquid equilibrium conditions and 2) mass and energy conservation equations. We use an open-source thermodynamic software, ThermoLib [47], [48], to evaluate thermodynamic functions based on the Peng-Robinson equation of state.

A. Vapor-liquid equilibrium

The UV flash problem is a mathematical statement of the second law of thermodynamics, i.e. that the entropy of a closed system in equilibrium is maximal. The internal energy, U , the volume, V , and the total composition (in moles), n , are specified in the UV flash. The equilibrium temperature, T , pressure, P , and vapor-liquid composition (in moles), n^v and n^l , are the solution to the optimization problem,

$$\max_{T, P, n^v, n^l} S = S^v(T, P, n^v) + S^l(T, P, n^l), \quad (68a)$$

$$\text{s.t.} \quad U^v(T, P, n^v) + U^l(T, P, n^l) = U, \quad (68b)$$

$$V^v(T, P, n^v) + V^l(T, P, n^l) = V, \quad (68c)$$

$$n_i^v + n_i^l = n_i, \quad i = 1, \dots, N_C. \quad (68d)$$

The UV flash is also called the UVn flash or the isoenergetic-isochoric (constant energy - constant volume) flash. The solution to (68) is characterized by the first-order optimality conditions which the algebraic equations (1a) represent. The optimization variables are the algebraic variables, and the Lagrange multipliers associated with (68) are the adjoint algebraic variables. The state variables are U and n .

B. Conservation of mass and energy

The internal energy, U , and the total mixture composition, n , are determined by the conservation equations,

$$\dot{U}(t) = H_F^v(t) + H_F^l(t) - H_V(t) - H_L(t) + Q(t), \quad (69a)$$

$$\dot{n}_i(t) = f_{F,i}^v(t) + f_{F,i}^l(t) - v_i(t) - l_i(t), \quad i = 1, \dots, N_C. \quad (69b)$$

H_F^v and H_F^l are the enthalpies, and $f_{F,i}^v$ and $f_{F,i}^l$ are the molar flow rates, of the vapor and liquid phases of the feed stream. Similarly, H_V and H_L are the enthalpies, and v_i and l_i are the flow rates, of the vapor stream and the liquid stream. Q refers to heating if it is positive and to cooling if it is negative. The inputs to the system, e.g. the feed stream and the vapor-liquid output streams, can be uncertain. Similarly, there can be uncertainty related to the thermodynamic parameters. That is what we model with the stochastic part of the differential equations (1b) [49].

TABLE I

ACCURACY AND COMPUTATION TIMES FOR A UV FLASH PROCESS WITH 6 DIFFERENTIAL EQUATIONS AND 19 ALGEBRAIC EQUATIONS.

	EKF	UKF	PF	EnKF
Avg. NRMSD	0.0216	0.0166	0.0161	0.0199
Avg. meas. upd. CPU (ms)	1.27	12.74	2.13	3.22
Avg. time upd. CPU (ms)	8.44	161.81	773.75	805.78

IX. NUMERICAL EXAMPLE

We consider the separation of a hydrocarbon mixture in a 0.2 m³ separator. The mixture contains 60% C₁, 8% C₂, 5% C₃, 25% n-C₇, and 2% CO₂. We estimate the states over a 72 h period with the EKF, UKF, PF, and EnKF. We use the parameter values $\alpha = 0.1$ and $\beta = 2$ in the UKF. We sample 100 particles in both the PF and the EnKF. We measure temperature and pressure every 30 min. All filters take $N_k = 6$ time steps of 5 min between the measurements. The separator is cooled with $Q = -9$ MJ/h for $t \in [0 \text{ h}; 24 \text{ h}]$ and with $Q = -4$ MJ/h for the remaining 48 h. The flow rates of the feed, the vapor stream, and the liquid stream are 1000 mol/h, 400 mol/h, and 600 mol/h, respectively. The temperature and pressure measurement noises have standard deviations of 10 K and $10^{-1/2} \approx 0.3$ MPa. We consider a constant diffusion coefficient, i.e. $\sigma(y(t), u(t)) = \sigma = \text{diag}([\sigma_U; \sigma_{C_1}; \sigma_{C_2}; \sigma_{C_3}; \sigma_{n-C_7}; \sigma_{CO_2}])$. The diagonal elements are $\sigma_U = 1$ MJ, $\sigma_{C_1} = \sigma_{C_2} = \sigma_{n-C_7} = 1$ mol, and $\sigma_{C_3} = \sigma_{CO_2} = 0.1$ mol. x_0 is a steady-state of the process (without process noise), and $P_0 = \sigma\sigma'$.

Fig. 1 shows the state estimates of the four filters together with the true states (blue). The estimates of all four filters are close to the true states. The root-mean-square deviation (RMSD) of the i 'th state variable is

$$\text{RMSD}_i = \left(\frac{1}{N+1} \sum_{k=0}^N (\hat{x}_{i,k|k} - x_{i,k})^2 \right)^{1/2}, \quad (70)$$

where $N = 144$ is the number of sampling intervals. The state variables have different units and orders of magnitude. We therefore compute the normalized RMSD (NRMSD). It is $\text{NRMSD}_i = \text{RMSD}_i / \bar{x}_i$, where \bar{x}_i is the average of the true states, $x_{i,k}$, over the index k . Table I shows the average NRMSD over the state variables for each filter together with the average computation times for a single measurement-update and time-update. The EKF is significantly faster than the other filters while the PF estimates have the lowest average NRMSD. Fig. 2 illustrates that the state estimation algorithms can be used for soft sensing of the vapor-liquid compositions. It shows the PF estimates of the total mole fractions, the vapor-liquid mole fractions, and the vapor fraction of the mixture.

X. CONCLUSIONS

We describe four nonlinear filters for state estimation of UV flash processes, i.e. the EKF, UKF, PF, and EnKF. We model the UV flash processes with stochastic DAEs in a semi-explicit index-1 form. We describe a model of a UV flash separation process and compare the accuracy and

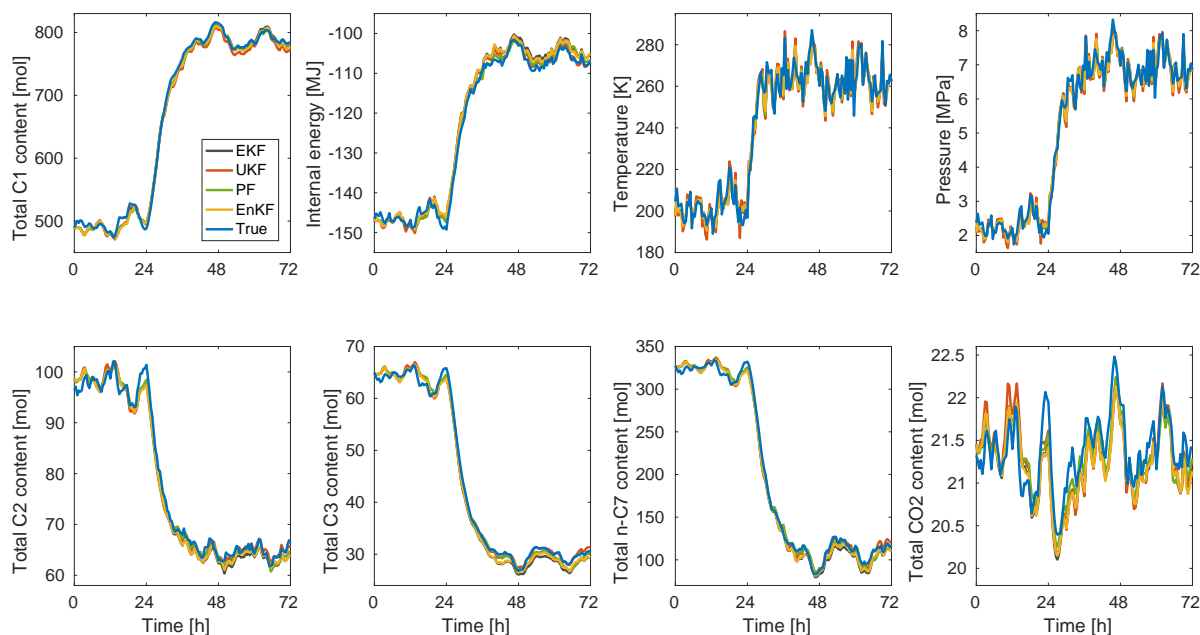


Fig. 1. Filtered estimates of total composition, internal energy, temperature, and pressure.

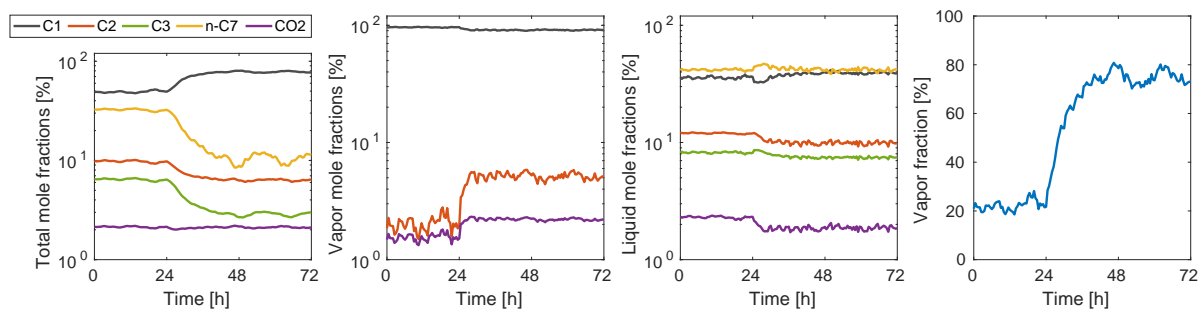


Fig. 2. PF estimates of the total mole fractions, vapor mole fractions, liquid mole fractions, and the vapor fraction. The estimates are based on the model and measurements of temperature and pressure. We omit the graphs of the C_3 and $n-C_7$ vapor mole fractions because they are below 1%.

efficiency of the filters with a numerical example. The PF is slightly more accurate than the other filters in terms of the average NRMSD of the estimates. However, all four filters provide estimates that are very close to the true states of the process. The EKF is significantly faster than the other filters. Finally, we demonstrate that the algorithms can be used for soft sensing of vapor-liquid compositions based on temperature and pressure measurements.

REFERENCES

- [1] J. M. Ali, N. H. Hoang, M. A. Hussain, and D. Dochain, "Review and classification of recent observers applied in chemical process systems," *Computers and Chemical Engineering*, vol. 76, pp. 27–41, 2015.
- [2] M. Geetha, J. Jerome, and P. A. Kumar, "Critical evaluation of nonlinear filter configurations for the state estimation of continuous stirred tank reactor," *Applied Soft Computing*, vol. 25, pp. 452–460, 2014.
- [3] M. Mansouri, H. Nounou, and M. Nounou, "State estimation of a chemical reactor process model - a comparative study," in *Proceedings of the 10th International Multi-Conference on Systems Signals & Devices*, Hammamet, Tunisia, Mar. 2013, pp. 1–6.
- [4] J. B. Jørgensen, P. G. Thomsen, H. Madsen, and M. R. Kristensen, "A computationally efficient and robust implementation of the continuous-discrete extended Kalman filter," in *Proceedings of the 2007 American Control Conference*, 2007, pp. 3706–3712.
- [5] S. Biagiola and J. Solsona, "State estimation in batch processes using a nonlinear observer," *Mathematical and Computer Modelling*, vol. 44, no. 11-12, pp. 1009–1024, 2006.
- [6] M.-J. Park, S.-M. Hur, and H.-K. Rhee, "Online estimation and control of polymer quality in a copolymerization reactor," *AIChE Journal*, vol. 48, no. 5, pp. 1013–1021, 2002.
- [7] D. I. Wilson, M. Agarwal, and D. W. T. Rippin, "Experiences implementing the extended Kalman filter on an industrial batch reactor," *Computers and Chemical Engineering*, vol. 22, no. 11, pp. 1653–1672, 1998.
- [8] P. de Vallière and D. Bonvin, "Application of estimation techniques to batch reactors - II. experimental studies in state and parameter estimation," *Computers and Chemical Engineering*, vol. 13, no. 1-2, pp. 11–20, 1989.
- [9] R. Hashemi, D. Kohlmann, and S. Engell, "Optimizing control and state estimation of a continuous polymerization process in a tubular reactor with multiple side-streams," *Macromolecular Reaction Engineering*, vol. 10, no. 4, pp. 415–434, 2016.
- [10] N. Barje, F. Barje, M. E. Aallaoui, and A. Kamal, "State estimators for isothermal plug-flow (bio) reactors," *International Journal of Applied Mathematics*, vol. 28, no. 2, pp. 1230–1235, 2013.
- [11] E. C. Rivera, D. I. P. Atala, F. M. Filho, A. C. da Costa, and R. M. Filho, "Development of real-time state estimators for reaction-separation processes: A continuous flash fermentation as a study case," *Chemical Engineering and Processing: Process Intensification*, vol. 49, no. 4, pp. 402–409, 2010.
- [12] R. D. Gudi, S. L. Shah, and M. R. Gray, "Multirate state and parameter

- estimation in an antibiotic fermentation with delayed measurements,” *Biotechnology and Bioengineering*, vol. 44, no. 11, pp. 1271–1278, 1994.
- [13] G. Kataria, K. Singh, and R. K. Dohare, “ANN based soft sensor model for reactive distillation column,” *International Journal of Advanced Technology and Engineering Exploration*, vol. 3, no. 24, pp. 182–186, 2016.
- [14] S. Pan, H. Su, P. Li, and Y. Gu, “State estimation for batch distillation operations with a novel extended Kalman filter approach,” in *Proceedings of the Joint 48th Conference on Decision and Control and 28th Chinese Control Conference*, Shanghai, China, December 2009, pp. 1884–1889.
- [15] A. K. Singh and J. Hahn, “State estimation for high-dimensional chemical processes,” *Computers and Chemical Engineering*, vol. 29, no. 11–12, pp. 2326–2334, 2005.
- [16] B. J. T. Binder, A. Pavlov, and T. A. Johansen, “Estimation of flow rate and viscosity in a well with an electric submersible pump using moving horizon estimation,” in *Proceedings of the 2nd IFAC Workshop on Automatic Control in Offshore Oil and Gas Production*, Florianópolis, Brazil, May 2015, pp. 140–146.
- [17] D. S. Oliver and Y. Chen, “Recent progress on reservoir history matching: a review,” *Computational Geosciences*, vol. 15, no. 1, pp. 185–221, 2011.
- [18] M. Castier, “Dynamic simulation of fluids in vessels via entropy maximization,” *Journal of Industrial and Engineering Chemistry*, vol. 16, no. 1, pp. 122–129, Jan. 2010.
- [19] A. R. J. Arendsen and G. F. Versteeg, “Dynamic thermodynamics with internal energy, volume, and amount of moles as states: Application to liquefied gas tank,” *Industrial & Engineering Chemistry Research*, vol. 48, no. 6, pp. 3167–3176, Jan. 2009.
- [20] E. R. A. Lima, M. Castier, and E. C. Biscaia Jr., “Differential-algebraic approach to dynamic simulations of flash drums with rigorous evaluation of physical properties,” *Oil & Gas Science and Technology*, vol. 63, no. 5, pp. 677–686, 2008.
- [21] P. Flatby, S. Skogestad, and P. Lundström, “Rigorous dynamic simulation of distillation columns based on UV-flash,” in *IFAC Symposium on Advanced Control of Chemical Processes (ADCHEM '94)*, 1994, pp. 261–266.
- [22] M. Hammer and A. Morin, “A method for simulating two-phase pipe flow with real equations of state,” *Computers & Fluids*, vol. 100, pp. 45–58, 2014.
- [23] L. Qiu, Y. Wang, and R. D. Reitz, “Multiphase dynamic flash simulations using entropy maximization and application to compressible flow with phase change,” *AIChE Journal*, vol. 60, no. 8, pp. 3013–3024, Aug. 2014.
- [24] M. L. Michelsen, “State function based flash specifications,” *Fluid Phase Equilibria*, vol. 158–160, pp. 617–626, June 1999.
- [25] T. K. S. Ritschel, A. Capolei, J. Gaspar, and J. B. Jørgensen, “An algorithm for gradient-based dynamic optimization of UV flash processes,” *Computers and Chemical Engineering*, 2017, in Press. DOI: <https://doi.org/10.1016/j.compchemeng.2017.10.007>.
- [26] D. Simon, *Optimal State Estimation: Kalman, H Infinity, and Nonlinear Approaches*. John Wiley & Sons, 2006.
- [27] G. Evensen, *Data Assimilation. The Ensemble Kalman Filter*, 2nd ed. Springer-Verlag Berlin Heidelberg, 2009.
- [28] —, “The ensemble Kalman filter for combined state and parameter estimation,” *IEEE Control Systems*, vol. 29, no. 3, pp. 83–104, 2009.
- [29] S. Gillijns, O. B. Mendoza, J. Chandrasekar, B. L. R. De Moor, D. S. Bernstein, and A. Ridley, “What is the ensemble Kalman filter and how well does it work?” in *Proceedings of the 2006 American Control Conference*, Minneapolis, Minnesota, USA, June 2006, pp. 4448–4453.
- [30] A. Alessandri, M. Baglietto, G. Battistelli, and V. Zavala, “Advances in moving horizon estimation for nonlinear systems,” in *Proceedings of the 49th IEEE Conference on Decision and Control*, Atlanta, Georgia, USA, Dec. 2010.
- [31] H. A. Talebi, F. Abdollahi, R. V. Patel, and K. Khorasani, *Neural Network-Based State Estimation of Nonlinear Systems. Application to Fault Detection and Isolation*, ser. Lecture Notes in Control and Information Sciences. Springer-Verlag New York, 2010, vol. 395.
- [32] P. Mobed, S. Munusamy, D. Bhattacharyya, and R. Rengaswamy, “State and parameter estimation in distributed constrained systems. I. extended Kalman filtering of a special class of differential-algebraic equation systems,” *Industrial & Engineering Chemistry Research*, vol. 56, no. 1, pp. 206–215, 2016.
- [33] J. B. Jørgensen, M. R. Kristensen, P. G. Thomsen, and H. Madsen, “New extended Kalman filter algorithms for stochastic differential algebraic equations,” in *Assessment and Future Directions of Non-linear Model Predictive Control*, ser. Lecture Notes in Control and Information Sciences. Springer-Verlag Berlin Heidelberg, 2007, vol. 358, pp. 359–366.
- [34] V. M. Becerra, P. D. Roberts, and G. W. Griffiths, “Applying the extended Kalman filter to systems described by nonlinear differential-algebraic equations,” *Control Engineering Practice*, vol. 9, no. 3, pp. 267–281, 2001.
- [35] J. L. Purohit, S. C. Patwardhan, and S. M. Mahajani, “Performance evaluation of bayesian state estimators for nonlinear DAE systems using a moderately high dimensional reactive distillation column model,” in *Proceedings of the 12th International Symposium on Process Systems Engineering and 25th European Symposium on Computer Aided Process Engineering*, Copenhagen, Denmark, May - June 2015, pp. 1763–1768.
- [36] —, “DAE-EKF-based nonlinear predictive control of reactive distillation systems exhibiting input and output multiplicities,” *Industrial & Engineering Chemistry Research*, vol. 52, no. 38, pp. 13 699–13 716, 2013.
- [37] R. Pastorino, D. Richiedei, J. Cuadrado, and A. Trevisani, “State estimation using multibody models and non-linear Kalman filters,” *International Journal of Non-Linear Mechanics*, vol. 53, pp. 83–90, 2013.
- [38] R. K. Mandela, R. Rengaswamy, S. Narasimhan, and L. N. Sridhar, “Recursive state estimation techniques for nonlinear differential algebraic systems,” *Chemical Engineering Science*, vol. 65, no. 16, pp. 4548–4556, August 2010.
- [39] R. K. Mandela, R. Rengaswamy, and S. Narasimhan, “Nonlinear state estimation of differential algebraic systems,” in *Proceedings of the 7th IFAC Symposium on Advanced Control of Chemical Processes*, 2009, pp. 792–797.
- [40] D. Haßkerl, S. Subramanian, R. Hashemi, M. Arshad, and S. Engell, “State estimation using a multi-rate particle filter for a reactive distillation column described by a DAE model,” in *Proceedings of the 25th Mediterranean Conference on Control and Automation*, Valletta, Malta, July 2017, pp. 876–881.
- [41] D. Haßkerl, M. Arshad, R. Hashemi, S. Subramanian, and S. Engell, “Simulation study of the particle filter and the EKF for state estimation of a large-scale DAE-system with multi-rate sampling,” in *Proceedings of the 11th IFAC symposium on Dynamics and Control of Process Systems Including Biosystems*, Trondheim, Norway, June 2016, pp. 490–495.
- [42] Y. Puranik, V. A. Bavdekar, S. C. Patwardhan, and S. L. Shah, “An ensemble Kalman filter for systems governed by differential algebraic equations (DAEs),” in *Proceedings of the 8th IFAC Symposium on Advanced Control of Chemical Processes*, Singapore, July 2012, pp. 531–536.
- [43] S. Julier, J. Uhlmann, and H. F. Durrant-Whyte, “A new method for the nonlinear transformation of means and covariances in filters and estimators,” *IEEE Transactions on Automatic Control*, vol. 45, no. 3, pp. 477–482, 2000.
- [44] S. J. Julier, “The scaled unscented transformation,” in *Proceedings of the American Control Conference*, Anchorage, Alaska, May 2002.
- [45] J. D. Hol, T. B. Schön, and F. Gustafsson, “On resampling algorithms for particle filters,” in *Proceedings of the 2006 IEEE Nonlinear Statistical Signal Processing Workshop*, Cambridge, UK, Sept. 2006, pp. 79–82.
- [46] R. Douc, O. Cappé, and E. Moulines, “Comparison of resampling schemes for particle filtering,” in *Proceedings of the 4th International Symposium on Image and Signal Processing and Analysis*, Zagreb, Croatia, Sept. 2005, pp. 64–69.
- [47] T. K. S. Ritschel, J. Gaspar, and J. B. Jørgensen, “A thermodynamic library for simulation and optimization of dynamic processes,” in *Proceedings of the 20th World Congress of the International Federation of Automatic Control*, 2017.
- [48] T. K. S. Ritschel, J. Gaspar, A. Capolei, and J. B. Jørgensen, “An open-source thermodynamic software library,” Department of Applied Mathematics and Computer Science, Technical University of Denmark, Tech. Rep. DTU Compute Technical Report-2016-12, 2016.
- [49] S. Kolás, B. A. Foss, and T. S. Schei, “Noise modeling concepts in nonlinear state estimation,” *Journal of Process Control*, vol. 19, no. 7, pp. 1111–1125, 2009.

Paper X

Production Optimization of Thermodynamically Rigorous Isothermal and Compositional Models

Authors:

Tobias K. S. Ritschel and John Bagterp Jørgensen

Published in:

Proceedings of the 16th European Conference on the Mathematics of Oil Recovery (ECMOR), September 3-6, 2018, Barcelona, Spain.

Introduction

The objective of production optimization is to maximize a financial measure of the long-term production from oil and gas reservoirs based on a model of the subsurface reservoir flow. Production optimization can be applied to both traditional recovery processes, e.g. waterflooding, and to enhanced oil recovery processes, e.g. chemical, biological, and thermal. Furthermore, production optimization is central to closed-loop reservoir management (Jansen et al., 2009) where an optimized production strategy is repeatedly updated when new data becomes available. An accurate subsurface reservoir flow model is a key component in production optimization because unmodeled physical phenomena can affect the utility of the optimized production strategy significantly.

In this work, we consider isothermal and compositional models of the subsurface reservoir flow, i.e. models of recovery processes where the temperature is constant. Such models combine two main principles: 1) conservation of mass and 2) phase equilibrium. The phase equilibrium conditions are derived from the second law of thermodynamics which states that the entropy of a closed system in equilibrium is maximal. However, isothermal systems are not closed because they exchange energy with their surroundings. In order to derive the phase equilibrium conditions, the condition of maximal entropy is applied to the combination of the isothermal system and its surroundings which constitutes a closed system. That condition is equivalent to a condition of minimal Helmholtz energy of the isothermal system alone (Callen, 1985) which is the phase equilibrium condition for isothermal systems. We formulate the phase equilibrium problem as a minimization problem where the objective function is the Helmholtz energy (Michelsen, 1999). It is called the VT (or VTn) flash optimization problem because volume, V , temperature, T , and the total amount of moles, n , are specified parameters in the problem. The VT flash optimization problem is part of the model and is thus an inner optimization problem in the outer production optimization problem. The solution to the VT flash optimization problem is the equilibrium pressure and phase compositions that minimize Helmholtz energy while satisfying the constraints on the volume and the total amount of moles. The specified total amount of moles, n , are given by the conservation of mass. It is common to use the equality of fugacities as a condition for phase equilibrium in the reservoir simulation and optimization literature (Garipov et al., 2018; Zaydullin et al., 2014; Kourounis et al., 2014). That condition is derived from the PT flash optimization problem in which Gibbs energy is minimized, and the temperature, pressure, and total amount of moles are specified parameters. In this paper, we discuss the relation between the VT flash and the equality of fugacities.

Production optimization is often applied to two-phase flow models or polymer flooding (Lei et al., 2012; Zhang and Li, 2007). However, Kourounis et al. (2014) use a single-shooting algorithm for production optimization of an isothermal and compositional model in which the phase equilibrium condition is the equality of fugacities. The single-shooting algorithm is often used to solve production optimization problems (Bukshytynov et al., 2015; Forouzanfar et al., 2013; Capolei et al., 2012) although alternative algorithms such as multiple-shooting (Cudas et al., 2017; Capolei and Jørgensen, 2012), simultaneous collocation (Heirung et al., 2011), gradient-free methods (Zhao et al., 2016), and artificial intelligence methods (Onwunalu and Durlofsky, 2010; Saputelli et al., 2002) are also used. The VT flash was previously considered by Polívka and Mikyška (2014) in the context of isothermal and compositional subsurface reservoir flow, and it has also been used to model several other types of processes (Santori and Luberti, 2016; Kou et al., 2016; Jindrová and Mikyška, 2015b; Castier and Tavares, 2005; Cabral et al., 2005; Benjelloun-Dabaghi et al., 2002; Espósito et al., 2000). Furthermore, algorithms have been developed for efficient solution of the VT flash (Jindrová and Mikyška, 2015a, 2013; Mikyška and Firoozabadi, 2011), and for simulation, state estimation, and dynamic optimization of phase equilibrium processes (Ritschel and Jørgensen, 2018b,c; Ritschel et al., 2017a) including VT flash processes. Ritschel and Jørgensen (2018d) apply production optimization to a model of thermal and compositional oil recovery. However, production optimization has not yet been applied to isothermal and compositional models that are based on the VT flash.

In this work, we use the gradient-based single-shooting algorithm described by Ritschel et al. (2017a) for production optimization of isothermal and compositional models based on the VT flash. We describe the isothermal and compositional model in terms of the VT flash optimization problem and the mass

conservation equations. We use the open-source thermodynamic software ThermoLib to evaluate thermodynamic properties based on the Peng-Robinson equation of state. ThermoLib also provides first and second order derivatives which are necessary in the single-shooting algorithm. We discretize the mass conservation equations with a finite volume method, and we demonstrate that the resulting model is in a semi-explicit differential-algebraic form. We implement the single-shooting algorithm in C/C++ and present a numerical example of optimized isothermal and compositional waterflooding.

Model

In this section, we present a thermodynamically rigorous model of isothermal and compositional oil reservoir flow during a waterflooding process. The model consists of 1) mass conservation equations which are partial differential equations and 2) phase equilibrium conditions that are derived from an optimization problem based on the second law of thermodynamics. We discretize the mass conservation equations with the finite volume method and enforce the condition of phase equilibrium in each grid cell of the discretized reservoir. The model is an adaptation of the thermal and compositional model described by Ritschel and Jørgensen (2018d) to isothermal waterflooding processes. The condition of constant temperature requires that the surroundings have infinite heat capacity such that their temperature, T , remains fixed. It also requires that all involved thermal conductivities are infinite such that energy is transferred instantly from any part of the reservoir to the surroundings until they are in thermal equilibrium.

Phase equilibrium

The fluid consists of a water phase (w), an oil phase (o), and a gas phase (g). The water phase is not miscible with the oil and the gas phase. The oil phase and the gas phase both contain N_C chemical components. All fluid phases are in thermal and mechanical equilibrium with each other and with the rock (r), i.e. $T^\alpha = T$ and $P^\alpha = P$ for $\alpha \in \{w, o, g, r\}$, and the oil and gas phase are in chemical equilibrium. Because the recovery process is isothermal, the system consisting of the fluid and the rock in the grid cell is not closed. However, the combined system of the fluid and the rock in the grid cell and their surroundings is closed. The surroundings of the grid cell consist of other grid cells and the surroundings of the reservoir. The second law of thermodynamics states that the entropy of the combined system is maximal when it is in equilibrium. That condition is equivalent to a condition of minimal Helmholtz energy of the fluid and the rock in the grid cell (Callen, 1985), i.e. not including the surroundings. The phase equilibrium conditions are derived from the VT flash optimization problem:

$$\min_{P, n^w, n^o, n^g} A^w + A^o + A^g + A^r, \quad (1a)$$

$$\text{subject to } V^w + V^o + V^g + V^r = V, \quad (1b)$$

$$n^w = n_w, \quad (1c)$$

$$n_k^o + n_k^g = n_k, \quad k = 1, \dots, N_C. \quad (1d)$$

n^w is the moles of water, and n^o and n^g are vectors of mole numbers in the oil and gas phase. $A^\alpha = A^\alpha(T, P, n^\alpha)$ and $V^\alpha = V^\alpha(T, P, n^\alpha)$ are the Helmholtz energy and the volume of phase $\alpha \in \{w, o, g, r\}$, and V is the volume of the grid cell. n_w and $n = [n_1; \dots; n_{N_C}]$ are the total amount of moles of water and the chemical components. They are given by the mass conservation equations, and they are parameters in the optimization problem (1). The solution to (1) is the equilibrium pressure, P , and phase compositions, n^α for $\alpha \in \{w, o, g\}$, that minimize Helmholtz energy while satisfying the volume constraint (1b) and the mole balance constraints (1c)-(1d). The phase equilibrium conditions are the first-order optimality conditions which are a set of algebraic equations because (1) does not contain inequality constraints (Nocedal and Wright, 2006).

Special case (incompressible rock): If the rock is incompressible, it can be considered part of the surroundings of the fluid in the grid cell in the derivation of the condition of minimal Helmholtz energy.

In that case, the relevant VT flash optimization problem is

$$\min_{P, n^w, n^o, n^g} A^w + A^o + A^g, \quad (2a)$$

$$\text{subject to } V^w + V^o + V^g = V^f, \quad (2b)$$

$$n^w = n_w, \quad (2c)$$

$$n_k^o + n_k^g = n_k, \quad k = 1, \dots, N_C, \quad (2d)$$

where $V^f = V - V^r$ is the volume that is available for fluid in the cell.

Mass conservation equations

The fluid flow is advective. The mass conservation equations are

$$\partial_t C_w = -\nabla \cdot \mathbf{N}^w + Q^w, \quad (3a)$$

$$\partial_t C_k = -\nabla \cdot \mathbf{N}_k + Q_k, \quad k = 1, \dots, N_C, \quad (3b)$$

where C_w and C_k are the molar concentrations of water and the k 'th component, and \mathbf{N}^w is the molar flux of the water phase. The molar flux of the k 'th component is

$$\mathbf{N}_k = x_k \mathbf{N}^o + y_k \mathbf{N}^g, \quad k = 1, \dots, N_C, \quad (4)$$

where \mathbf{N}^α is the molar flux of phase $\alpha \in \{o, g\}$, x_k is the oil mole fraction, and y_k is the gas mole fraction. The source terms are

$$Q^w = Q^{w, \text{inj}} - Q^{w, \text{prod}}, \quad (5a)$$

$$Q_k = -(x_k Q^{o, \text{prod}} + y_k Q^{g, \text{prod}}), \quad k = 1, \dots, N_C. \quad (5b)$$

The injection/production terms are zero in cells that are not perforated by a well. The molar injection rate of water is $Q^{w, \text{inj}}$ and the fluid phases are produced at molar rates of $Q^{\alpha, \text{prod}}$.

Darcy's law, relative permeability, and viscosity

The molar flux of phase $\alpha \in \{w, o, g\}$ is $\mathbf{N}^\alpha = \rho^\alpha \mathbf{u}^\alpha$ where $\rho^\alpha = \rho^\alpha(T, P, n^\alpha)$ is the phase density, and \mathbf{u}^α is the volumetric phase flux which is given by Darcy's law:

$$\mathbf{u}^\alpha = -\frac{k_r^\alpha}{\mu^\alpha} \mathbf{K} (\nabla P - \rho^\alpha g \nabla z), \quad \alpha \in \{w, o, g\}. \quad (6)$$

k_r^α is the relative permeability, μ^α is the viscosity, and \mathbf{K} is a permeability tensor. ∇P is the spatial gradient of P , g is the gravity acceleration, and ∇z is the spatial gradient of the depth, z . We use Stone's model II (Delshad and Pope, 1989) to model the relative permeabilities which are functions of the phase saturations, i.e. $k_r^\alpha = k_r^\alpha(\hat{S}^\alpha)$ where $\hat{S}^\alpha = V^\alpha / (V^w + V^o + V^g)$. Consequently, the relative permeability of phase $\alpha \in \{w, o, g\}$ is a function of temperature, pressure, and all fluid phase compositions:

$$k_r^\alpha = k_r^\alpha(T, P, n^w, n^o, n^g), \quad \alpha \in \{w, o, g\}. \quad (7)$$

We use the model of phase viscosity by Lohrenz et al. (1964) who describe viscosity as a function of the temperature, pressure, and phase composition:

$$\mu^\alpha = \mu^\alpha(T, P, n^\alpha), \quad \alpha \in \{w, o, g\}. \quad (8)$$

Well terms

The injection/production wells are perforated in certain cells in the discrete grid. The molar phase injection/production rates are

$$Q^{w, \text{inj}} = \frac{1}{V} \text{WI} \rho^w \frac{k_r^w}{\mu^w} (P^{\text{bhp}} - P), \quad (9a)$$

$$Q^{\alpha, \text{prod}} = \frac{1}{V} \text{WI} \rho^\alpha \frac{k_r^\alpha}{\mu^\alpha} (P - P^{\text{bhp}}), \quad \alpha \in \{w, o, g\}, \quad (9b)$$

where V is the volume of the grid cell, WI is the well index, and P^{bhp} is the well bottom-hole pressure.

Thermodynamics

We use the thermodynamic software library ThermoLib (Ritschel et al., 2017b, 2016) to evaluate thermodynamic functions of the fluid phases. ThermoLib provides routines for evaluating the enthalpy, entropy, and volume:

$$H^\alpha = H^\alpha(T, P, n^\alpha), \quad \alpha \in \{w, o, g\}, \quad (10a)$$

$$S^\alpha = S^\alpha(T, P, n^\alpha), \quad \alpha \in \{w, o, g\}, \quad (10b)$$

$$V^\alpha = V^\alpha(T, P, n^\alpha), \quad \alpha \in \{w, o, g\}. \quad (10c)$$

Other thermodynamic functions are given by the fundamental thermodynamic relations, $U^\alpha = H^\alpha - PV^\alpha$, $G^\alpha = H^\alpha - TS^\alpha$, and $A^\alpha = U^\alpha - TS^\alpha$. ThermoLib uses data and correlations from the DIPPR database (Thomson, 1996) together with cubic equations of state, e.g. the Soave-Redlich-Kwong or Peng-Robinson equations of state. The phase equilibrium conditions contain first-order derivatives of thermodynamic functions with respect to pressure, P , and the phase composition, n^α . In this work, we use the gradient-based dynamic optimization algorithm described by Ritschel et al. (2017a) which requires the Jacobian matrix of the phase equilibrium conditions. That matrix will contain second-order derivatives of thermodynamic properties with respect to pressure and phase compositions. Such derivatives are provided by ThermoLib based on analytical expressions described by Ritschel et al. (2016). We use an equation of state based on $(1/V^r)(\partial V^r/\partial P) = c_r$ for the rock. We assume that the compressibility of the rock, c_r , is constant.

Discretization

We use the same discretization scheme as Ritschel and Jørgensen (2018d) for the mass conservation equations. They use the finite volume method and a two-point flux approximation. The mass conservation equations (3) are in the form $\partial_t C = -\nabla \cdot \mathbf{N} + Q$. The discretized reservoir consists of a set of grid cells, Ω_i for $i \in \mathcal{N}$ where \mathcal{N} is the set of indices of the grid cells. In the finite volume method, we integrate the mass conservation equations over each grid cell:

$$\partial_t \int_{\Omega_i} C dV = - \int_{\Omega_i} \nabla \cdot \mathbf{N} dV + \int_{\Omega_i} Q dV, \quad i \in \mathcal{N}. \quad (11)$$

We apply Gauss' divergence theorem to the flux term, and split up the resulting surface integral over the faces of the grid cell:

$$\int_{\Omega_i} \nabla \cdot \mathbf{N} dV = \int_{\partial\Omega_i} \mathbf{N} \cdot \mathbf{n} dA = \sum_{j \in \mathcal{N}^{(i)}} \int_{\gamma_{ij}} \mathbf{N} \cdot \mathbf{n} dA, \quad i \in \mathcal{N}. \quad (12)$$

$\partial\Omega_i$ is the boundary of Ω_i , and γ_{ij} is the shared faced between the i 'th and j 'th grid cell. \mathbf{n} is the outward normal vector, and $\mathcal{N}^{(i)}$ contains the indices of the neighboring cells of the i 'th grid cell, i.e. cells that share a face with the i 'th grid cell. The left-hand side integrals in (11) can be evaluated exactly:

$$n_{w,i} = \int_{\Omega_i} C_w dV, \quad i \in \mathcal{N}, \quad (13a)$$

$$n_{k,i} = \int_{\Omega_i} C_k dV, \quad i \in \mathcal{N}. \quad (13b)$$

We use quadrature to approximate the remaining integrals:

$$\int_{\gamma_{ij}} \mathbf{N} \cdot \mathbf{n} dA \approx (AN \cdot \mathbf{n})_{ij}, \quad i \in \mathcal{N}, \quad j \in \mathcal{N}^{(i)}, \quad (14a)$$

$$\int_{\Omega_i} Q dV \approx (QV)_i, \quad i \in \mathcal{N}. \quad (14b)$$

The resulting differential equations are

$$\dot{n}_{w,i} = - \sum_{j \in \mathcal{N}^{(i)}} (\mathbf{AN}^w \cdot \mathbf{n})_{ij} + (Q^w V)_i, \quad i \in \mathcal{N}, \quad (15a)$$

$$\dot{n}_{k,i} = - \sum_{j \in \mathcal{N}^{(i)}} (\mathbf{AN}_k \cdot \mathbf{n})_{ij} + (Q_k V)_i, \quad i \in \mathcal{N}. \quad (15b)$$

We use the two-point flux approximation described by Lie (2014) to approximate the fluxes on the cell faces in (15).

Differential-algebraic equations

The state variables related to the i 'th grid cell are $x_i = [n_w; n]_i \in \mathbb{R}^{1+N_c}$, the algebraic variables are $y_i = [P; n^w; n^o; n^g]_i \in \mathbb{R}^{2+2N_c}$, the manipulated input variable is $u_i = P_i^{\text{bhp}} \in \mathbb{R}$, and there are no disturbance variables, d_i , in the model. Consequently, the VT flash optimization problem (1) is in the form

$$\min_{y_i} f(y_i), \quad (16a)$$

$$\text{subject to } g(y_i) = x_i, \quad (16b)$$

$$h(y_i) = 0. \quad (16c)$$

$f(y_i)$ is the Helmholtz energy of the fluid and the rock in (1a), (16b) is the mole balance constraints (1c)-(1d), and (16c) is the volume balance constraint (1b). The first-order optimality conditions of (16) are a set of algebraic equations, $G(x_i, y_i, z_i) = 0$, where $z_i \in \mathbb{R}^{2+2N_c}$ are Lagrange multipliers. The left-hand side of the discretized mass conservation equations (15) contains derivatives of the state variables, x_i , while both the fluxes and the well terms on the right-hand side depend only on the manipulated inputs, u_i , and the algebraic variables in the cell itself, y_i , and in neighboring cells, $\{y_j\}_{j \in \mathcal{N}^{(i)}}$. Consequently, the discretized mass conservation equations (15) are in the form $\dot{x}_i(t) = F(y_i(t), \{y_j(t)\}_{j \in \mathcal{N}^{(i)}}, u_i(t), d_i(t))$, and the entire discretized reservoir flow model is in the form

$$G(x(t), y(t), z(t)) = 0, \quad (17a)$$

$$\dot{x}(t) = F(y(t), u(t), d(t)). \quad (17b)$$

Relation to equality of fugacities

It is common, in compositional models of oil reservoir flow, to formulate the phase equilibrium conditions as the equality of fugacities:

$$f_k^o = f_k^g, \quad k = 1, \dots, N_c. \quad (18)$$

The fugacity of component k in phase $\alpha \in \{o, g\}$, $f_k^\alpha = f_k^\alpha(T, P, n^\alpha)$, is derived from the partial derivatives of Gibbs energy, $G^\alpha = G^\alpha(T, P, n^\alpha)$, and the ideal gas Gibbs energy, $G^{\text{ig}, \alpha} = G^{\text{ig}, \alpha}(T, P, n^\alpha)$, with respect to n_k^α (Ritschel and Jørgensen, 2018a, 2017). In order to relate the VT flash optimization problem (1) to the condition of equality of fugacities (18), we first describe the relation between the VT and the PT flash optimization problems. The VT flash optimization problem is equivalent to the combination of a PT flash optimization problem and a volume constraint. The PT flash optimization problem is

$$\min_{n^w, n^o, n^g} G^w + G^o + G^g + G^r, \quad (19a)$$

$$\text{subject to } n^w = n_w, \quad (19b)$$

$$n_k^o + n_k^g = n_k, \quad k = 1, \dots, N_c. \quad (19c)$$

The PT flash (19) is different from the VT flash (1) because 1) Gibbs energy is minimized 2) there is no volume constraint, and 3) pressure is not a decision variable. The volume constraint is

$$V^w + V^o + V^g + V^r = V. \quad (20)$$

The equilibrium pressure, P , and phase compositions, n^α for $\alpha \in \{w, o, g\}$, satisfy the first-order optimality conditions of (19) as well as the volume balance constraint (20). We prove that the VT flash optimization problem (1) is equivalent to the combination of (19) and (20) in the appendix. The volume constraint (20) is often reformulated as a saturation constraint. The Gibbs energy of the rock, G^r , in (19a) is independent of the decision variables. Therefore, it does not change the solution to (19) and can be removed. Furthermore, the constraint (19b) can be eliminated in which case the Gibbs energy of the water phase, G^w , in (19a) is also independent of the decision variables and can also be removed. The resulting optimization problem is

$$\min_{n^o, n^g} \quad G^o + G^g, \quad (21a)$$

$$\text{subject to} \quad n_k^o + n_k^g = n_k, \quad k = 1, \dots, N_C. \quad (21b)$$

The condition of equal fugacities (18) can be derived from the first-order optimality conditions of (21) (Ritschel and Jørgensen, 2018a, 2017). In conclusion, the VT flash optimization problem (1) is equivalent to the combination of the condition of equal fugacities (18), the volume constraint (20), and the eliminated water mole balance constraint, $n^w = n_w$.

Production Optimization

The production optimization problem is in the form

$$\min_{[x(t);y(t);z(t)]_{t_0}^{t_f}, \{u_k\}_{k=0}^{N-1}} \quad \phi = \int_{t_0}^{t_f} \Phi(y(t), u(t), d(t)) dt, \quad (22a)$$

subject to

$$x(t_0) = \hat{x}_0, \quad (22b)$$

$$G(x(t), y(t), z(t)) = 0, \quad t \in [t_0, t_f], \quad (22c)$$

$$\dot{x}(t) = F(y(t), u(t), d(t)), \quad t \in [t_0, t_f], \quad (22d)$$

$$u(t) = u_k, \quad t \in [t_k, t_{k+1}[, \quad k = 0, \dots, N-1, \quad (22e)$$

$$d(t) = \hat{d}_k, \quad t \in [t_k, t_{k+1}[, \quad k = 0, \dots, N-1, \quad (22f)$$

$$\{u_k\}_{k=0}^{N-1} \in \mathcal{U}. \quad (22g)$$

The decision variables are the state variables, $x(t)$, the algebraic variables, $y(t)$, and the adjoint algebraic variables, $z(t)$, as well as the manipulated inputs, $\{u_k\}_{k=0}^{N-1}$, where N is the number of control intervals. The objective is to optimize a financial measure, e.g. total oil production or net present value, over the time interval $[t_0, t_f]$. The financial measure is represented by the objective function, ϕ , in (22a). $x(t)$, $y(t)$, and $z(t)$ must satisfy the phase equilibrium conditions (22c) and the discretized mass conservation equations (22d) with the initial condition (22b) where \hat{x}_0 is an estimated initial state. (22e)-(22f) are zero-order hold parametrizations of the manipulated inputs and the disturbance variables, and $\{\hat{d}_k\}_{k=0}^{N-1}$ are predictions of the disturbance variables. (22g) represents constraints on the manipulated inputs. $x(t)$, $y(t)$, and $z(t)$ are dependent decision variables because they are determined by the model equations (22b)-(22d) when \hat{x}_0 , $\{u_k\}_{k=0}^{N-1}$, and $\{\hat{d}_k\}_{k=0}^{N-1}$ are given.

We solve the production optimization problem (22) with a gradient-based dynamic optimization algorithm described by Ritschel et al. (2017a). The algorithm uses a single-shooting method together with an adjoint method for computing gradients. The differential equations (22d) are discretized with Euler's implicit method, and the discretized differential equations are solved simultaneously with the algebraic equations (22c) using Newton's method. The algorithm uses a simplified version of the time step selection algorithm described by Völcker et al. (2010).

We implement the dynamic optimization algorithm in C++, and we use the open-source software library DUNE for solving linear systems with iterative methods (Blatt and Bastian, 2007) and for grid management (Bastian et al., 2008a,b). We use a GMRES method with a block ILU(1) preconditioner to solve

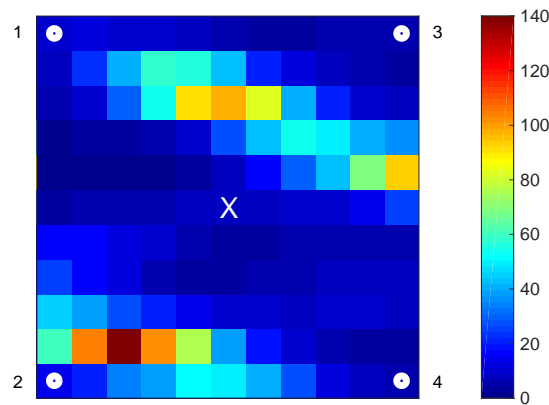


Figure 1 Permeability field [mD]. The white circles indicate the locations of the injectors, and the white X indicates the location of the producer.

the linear systems of equations, and the thermodynamic functions are evaluated with C routines from ThermoLib (Ritschel et al., 2017b). The single-shooting algorithm requires the solution of a numerical optimization problem which we solve with an SLQP algorithm (Nocedal and Wright, 2006, Chap. 18) that is implemented in the commercial optimization software KNITRO 10.2. We use a 64-bit workstation with 15.6 GB memory and four Intel Core i7 3.60 GHz cores to produce the numerical results in the following section. Each core has 64 KB of level 1 cache and 256 KB of level 2 cache. The workstation has a shared level 3 cache of 8192 KB.

Numerical Example

We consider a numerical example of optimized waterflooding of a $110 \times 110 \times 10$ m square reservoir at 50°C . We discretize the reservoir with a rectangular grid consisting of $11 \times 11 \times 1$ cells. The objective of the production optimization problem is to maximize the total oil recovery over a period of three years. The wells are placed in a five spot pattern which is shown together with the heterogeneous permeability field in Fig. 1. The porosity is 0.25, i.e. the porosity field is homogeneous. There are 12 control intervals per year, i.e. 36 control intervals in total. For simplicity, we consider the rock to be incompressible. The injector bottom-hole pressures (BHPs) are constrained to the interval $[11.0 \text{ MPa}, 12.0 \text{ MPa}]$, and the producer BHP is constrained to the interval $[8.5 \text{ MPa}, 11.0 \text{ MPa}]$.

Fig. 2 shows the optimized injector and producer BHPs together with the corresponding cumulative recovery of water, oil, and gas. For comparison, it also shows the cumulative oil recovery for a reference strategy where the injectors are operated at maximal BHP, and the producer is operated at minimal BHP. Fig. 3 shows the pressure and the oil and gas saturations during the three years of production. The area between injector 1 and the producer is highly permeable. Therefore, injector 1 is operated at the minimal BHP for the first two years. There is both a highly permeable area and an area with low permeability between injector 2 and the producer. Therefore, injector 2 is only operated at the maximal BHP during part of the production period. Injector 3 and 4 are located in areas with low permeability, and they are operated at the maximal BHP during the entire production period. The total oil recovery is 682 m^3 with the optimized strategy. For comparison, the total oil recovery is 521 m^3 with the reference strategy.

Conclusions

In this work, we consider production optimization of a thermodynamically rigorous isothermal and compositional subsurface reservoir flow model that is based on the VT flash. We formulate the VT flash as a minimization problem where the objective function is the Helmholtz energy. We describe the relation between the VT flash and the condition of equal fugacities which is often used as the phase equilibrium condition in models of subsurface reservoir flow. We show that the isothermal and compositional model is in a semi-explicit index-1 differential-algebraic form, and we use the gradient-based single-shooting algorithm by Ritschel et al. (2017a) to solve production optimization problems. We implement the al-

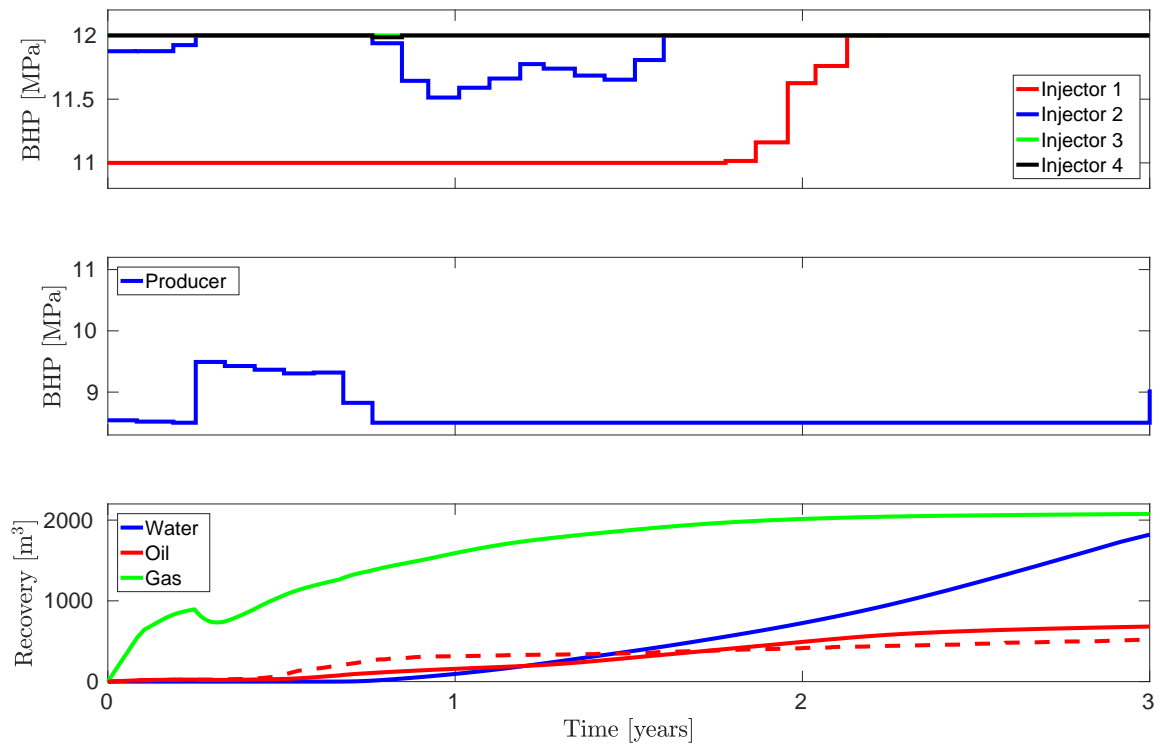


Figure 2 Top: Optimized BHPs of injectors (the graphs for injector 3 and 4 coincide). Middle: Optimized BHP of the producer. Bottom: Cumulative recovery of water, oil, and gas for the optimized strategy (solid), and cumulative oil recovery for the reference strategy (dashed).

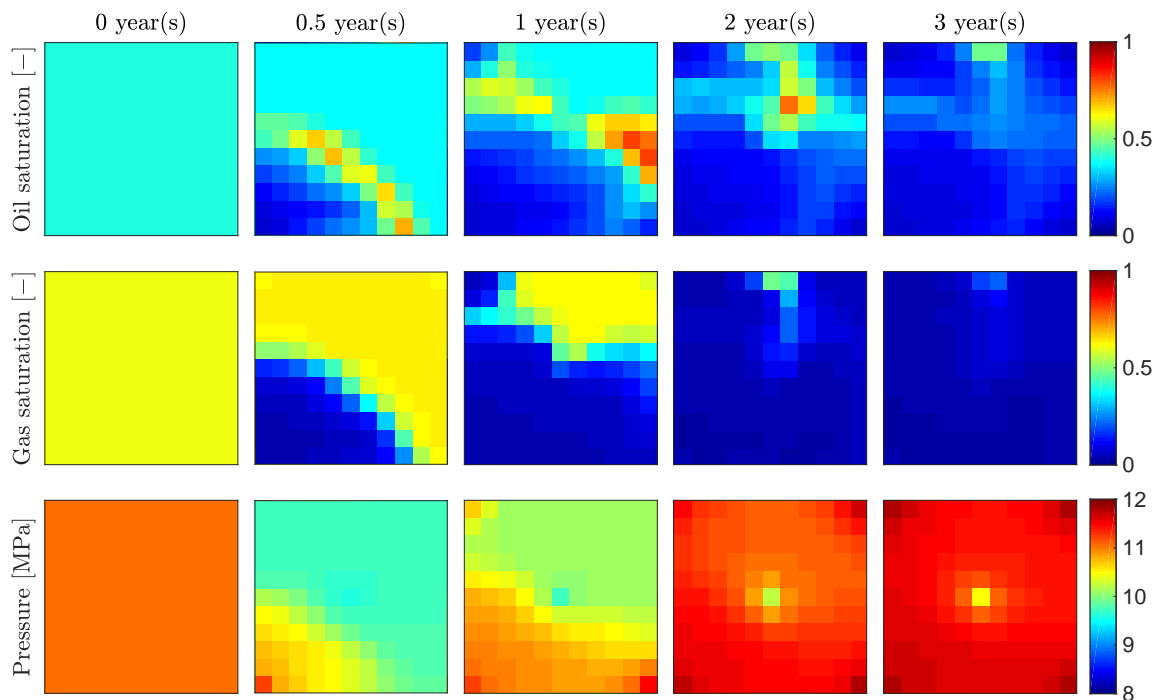


Figure 3 Optimized waterflooding.

gorithm in C/C++ using the open-source software packages DUNE and ThermoLib, and we present a numerical example of optimized isothermal waterflooding.

References

- Bastian, P., Blatt, M., Dedner, A., Engwer, C., Klöforn, R., Kornhuber, R., Ohlberger, M. and Sander, O. [2008a] A generic grid interface for parallel and adaptive scientific computing. Part II: implementation and tests in DUNE. *Computing*, **82**(2-3), 121–138.
- Bastian, P., Blatt, M., Dedner, A., Engwer, C., Klöforn, R., Ohlberger, M. and Sander, O. [2008b] A generic grid interface for parallel and adaptive scientific computing. Part I: abstract framework. *Computing*, **82**(2-3), 103–119.
- Benjelloun-Dabaghi, Z., de Hemptinne, J.C., Jarrin, J., Leroy, J.M., Aubry, J.C., Saas, J.N. and Tavel-Condard, C. [2002] MOLDITM: a fluid permeation model to calculate the annulus composition in flexible pipes. *Oil & Gas Science and Technology - Rev. IFP*, **57**(2), 177–192.
- Blatt, M. and Bastian, P. [2007] The iterative solver template library. In: B. Kågström et al. (Ed.) *Applied Parallel Computing. State of the Art in Scientific Computing. PARA 2006, Lecture Notes in Computer Science*, 4699. Springer, Berlin, Heidelberg, 666–675.
- Bukshtynov, V., Volkov, O., Durlafsky, L.J. and Aziz, K. [2015] Comprehensive framework for gradient-based optimization in closed-loop reservoir management. *Computational Geosciences*, **19**(4), 877–897.
- Cabral, V.F., Castier, M. and Tavares, F.W. [2005] Thermodynamic equilibrium in systems with multiple adsorbed and bulk phases. *Chemical Engineering Science*, **60**, 1773–1782.
- Callen, H.B. [1985] *Thermodynamics and an introduction to thermostatistics*. John Wiley & Sons, 2nd edn.
- Capolei, A. and Jørgensen, J.B. [2012] Solution of constrained optimal control problems using multiple shooting and ESDIRK methods. In: *Proceedings of the 2012 American Control Conference*. Fairmont Queen Elizabeth, Montréal, Canada, 295–300.
- Capolei, A., Völcker, C., Frydendall, J. and Jørgensen, J.B. [2012] Oil reservoir production optimization using single shooting and ESDIRK methods. In: *Proceedings of the 2nd IFAC Workshop on Automatic Control in Offshore Oil and Gas Production*. Trondheim, Norway, 286–291.
- Castier, M. and Tavares, F.W. [2005] Centrifugation equilibrium of natural gas. *Chemical Engineering Science*, **60**, 2927–2935.
- Codas, A., Hanssen, K.G., Foss, B., Capolei, A. and Jørgensen, J.B. [2017] Multiple shooting applied to robust reservoir control optimization including output constraints on coherent risk measures. *Computational Geosciences*, **21**(3), 479–497.
- Delshad, M. and Pope, G.A. [1989] Comparison of the three-phase oil relative permeability models. *Transport in Porous Media*, **4**(1), 59–83.
- Espósito, R.O., Castier, M. and Tavares, F.W. [2000] Calculations of thermodynamic equilibrium in systems subject to gravitational fields. *Chemical Engineering Science*, **55**, 3495–3504.
- Forouzanfar, F., Rossa, E.D., Russo, R. and Reynolds, A.C. [2013] Life-cycle production optimization of an oil field with an adjoint-based gradient approach. *Journal of Petroleum Science and Engineering*, **112**, 351–358.
- Garipov, T.T., Tomin, P., Rin, R., Voskov, D.V. and Tchepeli, H.A. [2018] Unified thermo-compositional-mechanical framework for reservoir simulation. *Computational Geosciences*, 1–19.
- Heirung, T.A.N., Wartmann, M.R., Jansen, J.D., Ydstie, B.E. and Foss, B.A. [2011] Optimization of the water-flooding process in a small 2D horizontal oil reservoir by direct transcription. In: *Proceedings of the 18th IFAC World Congress*. Milano, Italy, 10863–10868.
- Jansen, J.D., Douma, S.D., Brouwer, D.R., Van den Hof, P.M.J., Bosgra, O.H. and Heemink, A.W. [2009] Closed-loop reservoir management. In: *Proceedings of the 2009 SPE Reservoir Simulation Symposium*. Society of Petroleum Engineers, The Woodlands, Texas, USA.
- Jindrová, T. and Mikyška, J. [2013] Fast and robust algorithm for calculation of two-phase equilibria at given volume, temperature, and moles. *Fluid Phase Equilibria*, **353**, 101–114.
- Jindrová, T. and Mikyška, J. [2015a] General algorithm for multiphase equilibria calculation at given volume, temperature, and moles. *Fluid Phase Equilibria*, **393**, 7–25.
- Jindrová, T. and Mikyška, J. [2015b] Phase equilibria calculation of CO₂-H₂O system at given volume,

- temperature, and moles in CO₂ sequestration. *International Journal of Applied Mathematics*, **45**(3), 20–29.
- Kou, J., Sun, S. and Wang, X. [2016] An energy stable evolution method for simulating two-phase equilibria of multi-component fluids at constant moles, volume and temperature. *Computational Geosciences*, **20**, 283–295.
- Kourounis, D., Durlofsky, L.J., Jansen, J.D. and Aziz, K. [2014] Adjoint formulation and constraint handling for gradient-based optimization of compositional reservoir flow. *Computational Geosciences*, **18**(2), 117–137.
- Lei, Y., Li, S., Zhang, X. and Zhang, Q. [2012] Optimal control of polymer flooding for high temperature and high salinity reservoir. *International Journal of Advancements in Computing Technology*, **4**(12), 52–60.
- Lie, K.A. [2014] *An introduction to reservoir simulation using MATLAB*. Sintef ICT, Oslo, Norway.
- Lohrenz, J., Bray, B.G. and Clark, C.R. [1964] Calculating viscosities of reservoir fluids from their compositions. *Journal of Petroleum Engineers*, **16**(10), 1171–1176.
- Michelsen, M.L. [1999] State function based flash specifications. *Fluid Phase Equilibria*, **158-160**, 617–626.
- Michelsen, M.L. and Mollerup, J.M. [2007] *Thermodynamic models: fundamentals and computational aspects*. Tie-Line Publications, 2nd edn.
- Mikyška, J. and Firoozabadi, A. [2011] A new thermodynamic function for phase-splitting at constant temperature, moles, and volume. *AIChE Journal*, **57**(7), 1897–1904.
- Nocedal, J. and Wright, S.J. [2006] *Numerical optimization*. Springer Science & Business Media, 2nd edn.
- Onwunalu, J.E. and Durlofsky, L.J. [2010] Application of a particle swarm optimization algorithm for determining optimum well location and type. *Computational Geosciences*, **14**(1), 183–198.
- Polívka, O. and Mikyška, J. [2014] Compositional modeling in porous media using constant volume flash and flux computation without the need for phase identification. *Journal of Computational Physics*, **272**, 149–169.
- Ritschel, T.K.S., Capolei, A., Gaspar, J. and Jørgensen, J.B. [2017a] An algorithm for gradient-based dynamic optimization of UV flash processes. *Computers and Chemical Engineering*. In Press. DOI: <https://doi.org/10.1016/j.compchemeng.2017.10.007>.
- Ritschel, T.K.S., Gaspar, J., Capolei, A. and Jørgensen, J.B. [2016] An open-source thermodynamic software library. Tech. Rep. DTU Compute Technical Report-2016-12, Department of Applied Mathematics and Computer Science, Technical University of Denmark.
- Ritschel, T.K.S., Gaspar, J. and Jørgensen, J.B. [2017b] A thermodynamic library for simulation and optimization of dynamic processes. In: *Proceedings of the 20th IFAC World Congress*.
- Ritschel, T.K.S. and Jørgensen, J.B. [2017] Computation of phase equilibrium and phase envelopes. Tech. Rep. DTU Compute Technical Report-2017-11, Department of Applied Mathematics and Computer Science, Technical University of Denmark.
- Ritschel, T.K.S. and Jørgensen, J.B. [2018a] Computation of phase equilibrium in reservoir simulation and optimization. In: *Proceedings of the 3rd IFAC Workshop on Automatic Control in Offshore Oil and Gas Production*. Esbjerg, Denmark. Accepted.
- Ritschel, T.K.S. and Jørgensen, J.B. [2018b] The extended Kalman filter for state estimation of dynamic UV flash processes. In: *Proceedings of the 3rd IFAC Workshop on Automatic Control in Offshore Oil and Gas Production*. Esbjerg, Denmark. Accepted.
- Ritschel, T.K.S. and Jørgensen, J.B. [2018c] Nonlinear filters for state estimation of UV flash processes. In: *Proceedings of the 2nd IEEE Conference on Control Technology and Applications*. Copenhagen, Denmark. Accepted.
- Ritschel, T.K.S. and Jørgensen, J.B. [2018d] Production optimization of a rigorous thermal and compositional reservoir flow model. In: *Proceedings of the 3rd IFAC Workshop on Automatic Control in Offshore Oil and Gas Production*. Esbjerg, Denmark. Accepted.
- Santori, G. and Luberti, M. [2016] Thermodynamics of thermally-driven adsorption compression. *Sustainable Materials and Technologies*, **10**, 1–9.
- Saputelli, L., Malki, H., Canelon, J. and Nikolaou, M. [2002] A critical overview of artificial neural network applications in the context of continuous oil field optimization. In: *SPE Annual Technical Conference and Exhibition*. Society of Petroleum Engineers, San Antonio, Texas.

- Thomson, G.H. [1996] The DIPPR® databases. *International Journal of Thermophysics*, **17**(1), 223–232.
- Völcker, C., Jørgensen, J.B., Thomsen, P.G. and Stenby, E.H. [2010] Explicit singly diagonally implicit Runge-Kutta methods and adaptive stepsize control for reservoir simulation. In: *Proceedings of the 12th European Conference on the Mathematics of Oil Recovery*. Oxford, England.
- Zaydullin, R., Voskov, D.V., James, S.C., Henley, H. and Lucia, A. [2014] Fully compositional and thermal reservoir simulation. *Computers and Chemical Engineering*, **63**, 51–65.
- Zhang, X. and Li, S. [2007] Optimal control solving of polymer flooding in enhanced oil recovery with 2-D models. In: *Proceedings of the 2007 IEEE International Conference on Control and Automation*. Guangzhou, China, 1981–1986.
- Zhao, H., Tang, Y.W., Li, Y., Shi, Y.B., Cao, L., Gong, R.X. and Shang, G.H. [2016] Reservoir production optimization using general stochastic approximate algorithm under the mixed-linear-nonlinear constraints. *Journal of Residuals Science & Technology*, **13**(8).

Appendix: Proof of relation between the VT flash and the PT flash

In this appendix, we show that the VT flash optimization problem (1) is equivalent to the combination of the PT flash optimization problem (19) and the volume balance (20). First, we present the optimality conditions for the VT flash. Next, we present the optimality conditions for the PT flash. Finally, we show the equivalence.

First-order optimality conditions for the VT flash optimization problem

In order to derive the first-order optimality conditions for the VT flash optimization problem (1), we introduce the Lagrangian (Nocedal and Wright, 2006),

$$\mathcal{L}^{\text{VT}} = A^w + A^o + A^g + A^r - \lambda (V^w + V^o + V^g + V^r - V) - \bar{\mu}_w (n^w - n_w) - \sum_{k=1}^{N_C} \bar{\mu}_k (n_k^o + n_k^g - n_k). \quad (23)$$

λ , $\bar{\mu}_w$, and $\bar{\mu}_k$ are Lagrange multipliers. The first-order optimality conditions are the stationarity conditions,

$$\frac{\partial \mathcal{L}^{\text{VT}}}{\partial P} = \frac{\partial A^w}{\partial P} + \frac{\partial A^o}{\partial P} + \frac{\partial A^g}{\partial P} + \frac{\partial A^r}{\partial P} - \lambda \left(\frac{\partial V^w}{\partial P} + \frac{\partial V^o}{\partial P} + \frac{\partial V^g}{\partial P} + \frac{\partial V^r}{\partial P} \right) = 0, \quad (24a)$$

$$\frac{\partial \mathcal{L}^{\text{VT}}}{\partial n^w} = \frac{\partial A^w}{\partial n^w} - \lambda \frac{\partial V^w}{\partial n^w} - \bar{\mu}_w = 0, \quad (24b)$$

$$\frac{\partial \mathcal{L}^{\text{VT}}}{\partial n_k^o} = \frac{\partial A^o}{\partial n_k^o} - \lambda \frac{\partial V^o}{\partial n_k^o} - \bar{\mu}_k = 0, \quad k = 1, \dots, N_C, \quad (24c)$$

$$\frac{\partial \mathcal{L}^{\text{VT}}}{\partial n_k^g} = \frac{\partial A^g}{\partial n_k^g} - \lambda \frac{\partial V^g}{\partial n_k^g} - \bar{\mu}_k = 0, \quad k = 1, \dots, N_C, \quad (24d)$$

and the feasibility conditions,

$$V^w + V^o + V^g + V^r = V, \quad (24e)$$

$$n^w = n_w, \quad (24f)$$

$$n_k^o + n_k^g = n_k, \quad k = 1, \dots, N_C. \quad (24g)$$

First-order optimality conditions for the PT flash optimization problem

We introduce the Lagrangian associated with the PT flash optimization problem (19),

$$\mathcal{L}^{\text{PT}} = G^w + G^o + G^g + G^r - \mu_w (n^w - n_w) - \sum_{k=1}^{N_C} \mu_k (n_k^o + n_k^g - n_k). \quad (25)$$

μ_w and μ_k are Lagrange multipliers. The first-order optimality conditions are the stationarity conditions,

$$\frac{\partial \mathcal{L}^{\text{PT}}}{\partial n^w} = \frac{\partial G^w}{\partial n^w} - \mu_w = 0, \quad (26a)$$

$$\frac{\partial \mathcal{L}^{\text{PT}}}{\partial n_k^o} = \frac{\partial G^o}{\partial n_k^o} - \mu_k = 0, \quad k = 1, \dots, N_C, \quad (26b)$$

$$\frac{\partial \mathcal{L}^{\text{PT}}}{\partial n_k^g} = \frac{\partial G^g}{\partial n_k^g} - \mu_k = 0, \quad k = 1, \dots, N_C, \quad (26c)$$

and the feasibility conditions,

$$n^w = n_w, \quad (26d)$$

$$n_k^o + n_k^g = n_k, \quad k = 1, \dots, N_C. \quad (26e)$$

Equivalence

In order to show equivalence between the VT flash optimization problem (1) and the combination of the PT flash optimization problem (19) and the volume constraint (1), we need to show that (24) is equivalent to the combination of (26) and the volume balance (20). Clearly, the feasibility conditions (24e)-(24g) are equivalent to the feasibility conditions (26d)-(26e) combined with the volume balance (20). We use the fundamental thermodynamic relation $G^\alpha = A^\alpha + PV^\alpha$ to rewrite the stationarity conditions (26a)-(26c):

$$\frac{\partial \mathcal{L}^{\text{PT}}}{\partial n^w} = \frac{\partial A^w}{\partial n^w} + P \frac{\partial V^w}{\partial n^w} - \mu_w = 0, \quad (27a)$$

$$\frac{\partial \mathcal{L}^{\text{PT}}}{\partial n_k^o} = \frac{\partial A^o}{\partial n_k^o} + P \frac{\partial V^o}{\partial n_k^o} - \mu_k = 0, \quad k = 1, \dots, N_C, \quad (27b)$$

$$\frac{\partial \mathcal{L}^{\text{PT}}}{\partial n_k^g} = \frac{\partial A^g}{\partial n_k^g} + P \frac{\partial V^g}{\partial n_k^g} - \mu_k = 0, \quad k = 1, \dots, N_C. \quad (27c)$$

The stationarity conditions (27) are equivalent to (24b)-(24d) provided that

$$\lambda = -P, \quad (28a)$$

$$\bar{\mu}_w = \mu_w, \quad (28b)$$

$$\bar{\mu}_k = \mu_k, \quad k = 1, \dots, N_C. \quad (28c)$$

Finally, the fundamental thermodynamic relation $(\partial A^\alpha / \partial P) = -P(\partial V^\alpha / \partial P)$ (Michelsen and Mollerup, 2007, Table 3) implies that (24a) is satisfied for $\lambda = -P$. In conclusion, we have shown that the first-order optimality conditions (24) are equivalent to the first-order optimality conditions (26) combined with the volume constraint (20). This implies that the VT flash optimization problem (1) is equivalent to the combination of the PT flash optimization problem (19) and the volume balance (20).

Paper XI

Dynamic Optimization of Thermodynamically Rigorous Models of Multiphase Flow in Porous Subsurface Oil Reservoirs

Authors:

Tobias K. S. Ritschel and John Bagterp Jørgensen

Submitted to:

Journal of Process Control, 2018.

Dynamic Optimization of Thermodynamically Rigorous Models of Multiphase Flow in Porous Subsurface Oil Reservoirs

Tobias K. S. Ritschel, John Bagterp Jørgensen*

*Department of Applied Mathematics and Computer Science & Center for Energy Resources Engineering (CERE),
Technical University of Denmark, DK-2800 Kgs. Lyngby, Denmark*

Abstract

In this paper, we consider dynamic optimization of thermal and isothermal oil recovery processes which involve multicomponent three-phase flow in porous media. We present thermodynamically rigorous models of these processes based on 1) conservation of mass and energy, and 2) phase equilibrium. The conservation equations are partial differential equations. The phase equilibrium problems that are relevant to thermal and isothermal models are called the UV and the VT flash, and they are based on the second law of thermodynamics. We formulate these phase equilibrium problems as optimization problems and the phase equilibrium conditions as the corresponding first order optimality conditions. We demonstrate that the thermal and isothermal flow models are in a semi-explicit differential-algebraic form, and we solve the dynamic optimization problems with a previously developed gradient-based algorithm implemented in C/C++. We present numerical examples of optimized thermal and isothermal oil recovery strategies and discuss the computational performance of the dynamic optimization algorithm in these examples.

Keywords: Dynamic optimization, Single-shooting, The adjoint method, Thermal and isothermal oil recovery, Multicomponent multiphase flow, Phase equilibrium, UV flash, VT flash

1. Introduction

Dynamic optimization is concerned with the computation of an optimal open-loop control strategy for a dynamical process. The objective of the optimization is either to 1) minimize the distance to predefined setpoints or 2) optimize the economics of the process. Dynamic optimization of multiphase flow processes in porous media is relevant to numerous engineering applications, e.g. production of oil from subsurface reservoirs [1–6], geothermal energy systems [7], groundwater contamination and remediation [8], trickle bed reactors [9], fuel cells [10, 11], food processing [12], and several others [13].

In this work, we consider dynamic optimization of thermal (varying temperature) and isothermal (constant temperature) oil recovery processes which involve multicomponent multiphase flow in porous rock. Oil recovery processes are described as primary, secondary, or tertiary [14, 15]. In primary recovery processes, the oil is recovered by means of the initial pressure in the reservoir. In secondary recovery processes, water is injected into the reservoir in order to maintain a high pressure. Tertiary recovery involves chemical, biological, or thermal injection with the purpose of mobilizing and recovering the oil that remains after the primary and secondary recovery processes. We are concerned with dynamic optimization of the secondary recovery process (also called waterflooding). However,

dynamic optimization is equally applicable to the tertiary recovery processes (also called enhanced oil recovery processes). The objective of the dynamic optimization is to compute a field-wide production strategy that optimizes a long-term financial measure of the oil production, e.g. the total recovery or the net present value over the life-time of the oil reservoir.

Models of thermal and isothermal reservoir flow are based on two main principles: 1) conservation of mass and energy and 2) phase equilibrium. The conservation of energy is related to the first law of thermodynamics, while the equilibrium between phases is related to the second law of thermodynamics. The conservation equations are partial differential equations, and we formulate the phase equilibrium problems as inner optimization problems [16]. Consequently, the dynamic optimization problem that we consider belongs to the class of bilevel optimization problems [17] as well as the closely related class of mathematical programs with equilibrium constraints [18, 19]. We use the method of lines and discretize the conservation equations with a finite volume method in which the reservoir is represented by a discrete grid. The result of the discretization is a set of differential equations for each cell in the grid. Furthermore, we enforce the condition of phase equilibrium in each grid cell. The phase equilibrium conditions in the thermal model are different from the phase equilibrium conditions in the isothermal model. However, both sets of conditions are derived from the second law of thermodynamics which states that the entropy of a closed system in equilibrium is maximal [20–22]. The phase equilibrium optimization problem in the thermal model is the UV (or UVn) flash which is a direct statement of the second law

*Corresponding author.

Email addresses: tobk@dtu.dk (Tobias K. S. Ritschel), jbj@dtu.dk (John Bagterp Jørgensen)

of thermodynamics. The objective of the UV flash optimization problem is to maximize entropy subject to constraints on the internal energy, U , the volume, V , and the total composition, n , i.e. total amount of moles of each component. The internal energy and the total composition are determined by the conservation equations while the volume is the size of the grid cell. The solution to the UV flash is the equilibrium temperature, pressure, and phase compositions. Isothermal systems are not closed. Consequently, the condition of maximal entropy does not apply directly. Instead, the Helmholtz energy is minimal for isothermal systems in equilibrium [20]. The phase equilibrium optimization problem in the isothermal model is the VT (or VTn) flash which involves minimization of the Helmholtz energy subject to constraints on the volume, V , the temperature, T , and the total composition, n . The solution to the VT flash is the equilibrium pressure and phase compositions. In the reservoir simulation and optimization literature, it is common to formulate the phase equilibrium conditions as the isofugacity condition [1, 2, 6, 14, 15] which is derived from the PT (or PTn) flash [23, 24]. The PT flash is relevant to isothermal and isobaric (constant pressure) systems. For such systems, the Gibbs energy is minimal at equilibrium. Consequently, the PT flash involves minimization of the Gibbs energy subject to constraints on the temperature, T , pressure, P , and total composition, n . The solution to the PT flash is the equilibrium phase compositions. The UV and the VT flash are related to the PT flash [5, 25], and the condition of isofugacity in thermal and isothermal compositional reservoir flow models is derived from the UV and the VT flash.

Dynamic optimization of reservoir flow models most often involves models of immiscible two-phase flow [26–29], partially miscible two-phase flow [30], or polymer flooding [31, 32]. Garipov et al. [1] and Zaydullin et al. [6] consider the simulation of thermal and compositional reservoir flow models, and Kourounis et al. [2] present a gradient-based algorithm for dynamic optimization of isothermal and compositional reservoir flow models. However, none of the above models involve thermodynamically rigorous phase equilibrium conditions based on the UV or the VT flash. Polívka and Míkyška [3] consider simulation of an isothermal and compositional model that involves the VT flash. Dynamic optimization of UV flash processes was first addressed by Ritschel et al. [25, 33, 34]. Furthermore, dynamic optimization of thermal and isothermal compositional reservoir flow models based on the UV and the VT flash was first considered by Ritschel and Jørgensen [4, 5].

There exists a number of algorithms for dynamic optimization of nonlinear systems [35]. Single-shooting algorithms involve the solution of numerical optimization problems in which the number of decision variables is independent of the number of state variables in the model. Therefore, single-shooting algorithms are often used for dynamic optimization of reservoir flow models [26, 27, 36] which typically involve a large number of state variables, i.e. on the order of 10^4 - 10^7 state variables. Alternative algorithms include multiple-shooting and simultaneous collocation which both require the solution of numerical optimization problems in which the number of decision vari-

ables does depend on the number of state variables. For reservoir models, the solution of such optimization problems can be intractable due to both high computation time and excessive memory requirements. However, both multiple-shooting [28] and simultaneous collocation [29] have been used for dynamic optimization of reservoir flow models. Efficient algorithms for the solution of numerical optimization problems require the gradients of the objective function. For single- and multiple-shooting algorithms, such gradients can be computed efficiently with either an adjoint method [27, 37, 38] or a forward method [39]. Alternatives to gradient-based optimization algorithms include stochastic approximation methods [40] and metaheuristic methods [41].

In this work, we present thermodynamically rigorous models of thermal and isothermal waterflooding processes. We use the method of lines and discretize the conservation equations with a finite volume method. We demonstrate that the resulting equations are in a semi-explicit index-1 differential-algebraic form. Ritschel et al. [25] describe a gradient-based dynamic optimization algorithm for such systems. The algorithm uses a single-shooting method together with an adjoint method for the computation of gradients. We implement the algorithm in C/C++ based on the open-source software DUNE [42–44], the open-source software ThermoLib [45, 46], and the commercial software KNITRO (IPOPT [47] is an open-source alternative to KNITRO). We use the thermodynamic software ThermoLib to evaluate thermodynamic properties based on the Peng-Robinson equation of state. The ThermoLib routines also evaluate the first and second order derivatives of the thermodynamic functions which are necessary in the gradient-based dynamic optimization algorithm. Finally, we present numerical examples of optimized thermal and isothermal waterflooding strategies, and we discuss the computational performance of the C/C++ implementation.

The remainder of this paper is organized as follows. We present the thermal and compositional model in Section 2, and we present the isothermal and compositional model in Section 3. We formulate the dynamic optimization problem and discuss the C/C++ implementation in Section 4. In Section 5, we present the numerical examples, and we present conclusions in Section 6.

2. Thermal and compositional reservoir flow model

In this section, we describe the thermal and compositional model. The waterflooding process is illustrated in Fig. 1 for a rectangular (and discretized) reservoir. The model consists of a set of mass conservation equations, one energy balance equation, and a set of phase equilibrium conditions. The flow of mass in the reservoir is due to advection while the flow of energy is due to both advection and conduction. The phase equilibrium problem is the UV flash, and we assume that the fluid and the rock reach thermal, mechanical, and chemical equilibrium instantaneously, i.e. that they are in equilibrium at all times. We use a finite volume method to discretize the conservation equations, and we enforce the phase equilibrium in each cell of the discretized reservoir. Finally, we demonstrate

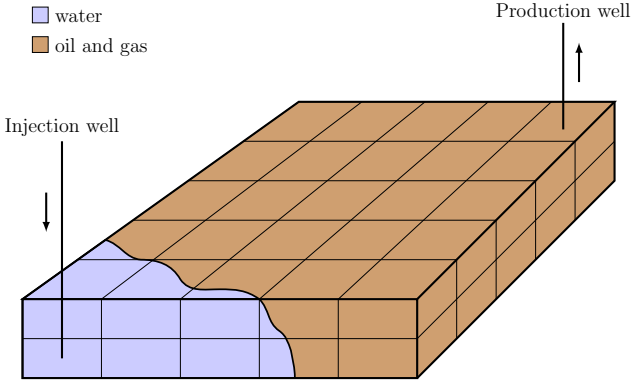


Figure 1: Illustration of the waterflooding process.

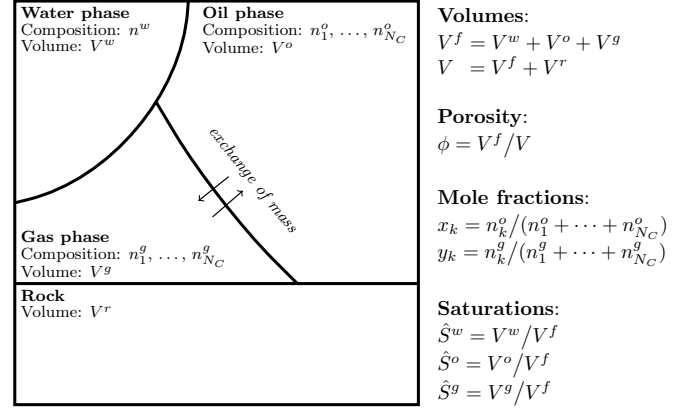


Figure 2: Illustration of the fluid phases and the rock in each grid cell.

that the thermal and compositional model is in the semi-explicit index-1 differential-algebraic form.

2.1. Phase equilibrium

Each grid cell in the discretized reservoir contains a water phase (w), an oil phase (o), a gas phase (g), and a (solid) rock phase (r) as illustrated in Fig. 2. The water phase is immiscible with the oil and the gas phase. The water phase contains only water while the oil and the gas phase contain N_C chemical components. The fluid phases and the rock are in thermal and mechanical equilibrium, i.e. $T^\alpha = T$ and $P^\alpha = P$ for $\alpha \in \{w, o, g, r\}$. Furthermore, the oil and the gas phase are in chemical equilibrium. The UV flash optimization problem describing the isoenergetic-isochoric chemical equilibrium is

$$\max_{T, P, n^w, n^o, n^g} S^w + S^o + S^g + S^r, \quad (1a)$$

$$\text{subject to } U^w + U^o + U^g + U^r = U, \quad (1b)$$

$$V^w + V^o + V^g + V^r = V, \quad (1c)$$

$$n^w = n_w, \quad (1d)$$

$$n_k^o + n_k^g = n_k, \quad k = 1, \dots, N_C. \quad (1e)$$

$S^\alpha = S^\alpha(T, P, n^\alpha)$, $U^\alpha = U^\alpha(T, P, n^\alpha)$, and $V^\alpha = V^\alpha(T, P, n^\alpha)$ are the entropy, internal energy, and volume of phase $\alpha \in \{w, o, g, r\}$, respectively. U and V are the specified internal energy and volume. n_w and n_k are the specified total amount of moles of water and of component k . n_w , n_k , and U are determined by the mass and energy conservation equations, and V is the size of the grid cell in the discretized reservoir. The solution to (1) is the equilibrium temperature, T , pressure, P , and phase compositions, n^α for $\alpha \in \{w, o, g\}$.

2.2. Conservation of mass

Each mass conservation equation contains 1) a flux term related to the flow in the reservoir and 2) a source term related to the injection of water and the production of the reservoir fluid:

$$\partial_t C_w = -\nabla \cdot \mathbf{N}^w + Q^w, \quad (2a)$$

$$\partial_t C_k = -\nabla \cdot \mathbf{N}_k + Q_k, \quad k = 1, \dots, N_C. \quad (2b)$$

C_w and C_k are the molar concentrations of water and component k . \mathbf{N}^w is the molar flux of the water phase, and the molar flux of the k 'th component is

$$\mathbf{N}_k = x_k \mathbf{N}^o + y_k \mathbf{N}^g, \quad k = 1, \dots, N_C. \quad (3)$$

\mathbf{N}^α is the molar flux of phase $\alpha \in \{o, g\}$. x_k and y_k are the oil and gas mole fractions of component k , i.e. the moles of component k in the oil and the gas phase divided by the total amount of moles in the respective phase. The source terms describe the molar well flow rates:

$$Q^w = Q^{w, \text{inj}} - Q^{w, \text{prod}}, \quad (4a)$$

$$Q_k = -(x_k Q^{o, \text{prod}} + y_k Q^{g, \text{prod}}), \quad k = 1, \dots, N_C. \quad (4b)$$

In Section 2.4, we provide expressions for the molar injection rate of the water phase, $Q^{w, \text{inj}}$, and the molar production rate of phase $\alpha \in \{w, o, g\}$, $Q^{\alpha, \text{prod}}$.

2.3. Conservation of energy

First, we describe the conservation of energy of the fluid (f) and the rock (r) separately, i.e. without assuming thermal equilibrium between the fluid and the rock. Consequently, we distinguish between the temperature of the fluid phases, T^f , and the temperature of the rock, T^r . Next, we incorporate the assumption of thermal equilibrium and present the energy conservation equation for the combined rock-fluid system. The energy conservation equations for the fluid and the rock are

$$\partial_t u^f = -\nabla \cdot \mathbf{N}_u^f + Q_u^f, \quad (5a)$$

$$\partial_t u^r = -\nabla \cdot \mathbf{N}_u^r + Q_u^r. \quad (5b)$$

u^f and u^r are the internal energies per unit volume of the fluid and the rock. The heat flux of the fluid is due to the advective flow of the three fluid phases:

$$\mathbf{N}_u^f = h^w \mathbf{N}^w + h^o \mathbf{N}^o + h^g \mathbf{N}^g. \quad (6)$$

$h^\alpha = h^\alpha(T, P, n^\alpha)$ is the molar enthalpy of phase $\alpha \in \{w, o, g\}$. The heat flux of the rock is due to conduction, and we describe it using Fourier's law of thermal conduction [48, Chap. 1]:

$$\mathbf{N}_u^r = -k_T^r \nabla T^r. \quad (7)$$

k_T^r is the thermal conductivity of the rock. The source term in the fluid energy balance (5a) describes 1) the transfer of energy through the wells and 2) the transfer of energy through the rock-fluid interface:

$$Q_u^f = h^{w,\text{inj}} Q^{w,\text{inj}} - \sum_{\alpha \in \{w,o,g\}} h^\alpha Q^{\alpha,\text{prod}} + Q^{rf}. \quad (8)$$

$h^{w,\text{inj}}$ is the molar enthalpy of the injected water. We describe the thermal conduction through the rock-fluid interface using Newton's law of cooling [48, Chap. 1]:

$$Q^{rf} = -k_T^{rf} (T^f - T^r). \quad (9)$$

k_T^{rf} is the thermal conductivity of the rock-fluid interface. The source term in the energy balance for the rock contains terms describing 1) the transfer of energy through the rock-fluid interface and 2) the transfer of energy to the surroundings (s) of the reservoir:

$$Q_u^r = -Q^{rf} - Q^{rs}. \quad (10)$$

We describe the thermal conduction through the interface between the rock and the surroundings of the reservoir using Newton's law of cooling:

$$Q^{rs} = -k_T^{rs} (T^s - T^r). \quad (11)$$

T^s is the temperature of the surroundings, and k_T^{rs} is the thermal conductivity of the interface between the rock and surroundings. Now, we assume that energy is transferred instantaneously between the fluid and the rock, i.e. that the thermal conductivity of the rock-fluid interface, k_T^{rf} , is infinite. Consequently, the temperature of the fluid and the rock are equal, i.e. $T^f = T^r = T$. In order to obtain a conservation equation for the internal energy of the combined rock-fluid system, $u = u^f + u^r$, we add (5a) and (5b):

$$\partial_t u = -\nabla \cdot \mathbf{N}_u + Q_u. \quad (12)$$

The heat flux, \mathbf{N}_u , and the source term, Q_u , are

$$\mathbf{N}_u = h^w \mathbf{N}^w + h^o \mathbf{N}^o + h^g \mathbf{N}^g - k_T^r \nabla T, \quad (13a)$$

$$Q_u = h^{w,\text{inj}} Q^{w,\text{inj}} - \sum_{\alpha \in \{w,o,g\}} h^\alpha Q^{\alpha,\text{prod}} - Q^{rs}. \quad (13b)$$

2.4. Well equations

The injection and the production wells are perforated in certain places in the reservoir, i.e. the injection and production source terms will only be nonzero in a few locations. The model of the well flow depends on the discretization of the reservoir. For a given grid cell, the molar injection and production phase flow rates are

$$Q^{w,\text{inj}} = \frac{1}{V} \text{WI} \rho^w \frac{k_r^w}{\mu^w} (P^{\text{bhp}} - P), \quad (14a)$$

$$Q^{\alpha,\text{prod}} = \frac{1}{V} \text{WI} \rho^\alpha \frac{k_r^\alpha}{\mu^\alpha} (P - P^{\text{bhp}}), \quad \alpha \in \{w, o, g\}. \quad (14b)$$

V is the volume of the perforated grid cell, and WI is the well index which is a scalar quantity that describes the ability of the perforation to transmit fluid. $\rho^\alpha = \rho^\alpha(T, P, n^\alpha)$, $k_r^\alpha = k_r^\alpha(T, P, n^w, n^o, n^g)$, and $\mu^\alpha = \mu^\alpha(T, P, n^\alpha)$ are the molar density, the relative permeability, and the viscosity of phase $\alpha \in \{w, o, g\}$. P^{bhp} is the bottom-hole pressure (BHP) in the well.

2.5. Darcy's law

The molar phase flux is the product of the molar density and the volumetric phase flux:

$$\mathbf{N}^\alpha = \rho^\alpha \mathbf{u}^\alpha, \quad \alpha \in \{w, o, g\}. \quad (15)$$

We describe the volumetric phase flux with Darcy's law:

$$\mathbf{u}^\alpha = -\frac{k_r^\alpha}{\mu^\alpha} \mathbf{K} (\nabla P - \rho^\alpha g \nabla z), \quad \alpha \in \{w, o, g\}. \quad (16)$$

\mathbf{K} is a permeability tensor, g is the gravity acceleration, and z is the depth.

2.6. Relative permeability

We use Stone's model II to describe the relative permeabilities [49]. The relative permeabilities depend on the phase saturations, $\hat{S}^\alpha = V^\alpha / (V^w + V^o + V^g)$ for $\alpha \in \{w, o, g\}$. Consequently, the relative permeability of phase $\alpha \in \{w, o, g\}$ depends on the temperature, pressure, and the compositions of all phases:

$$k_r^\alpha = k_r^\alpha(T, P, n^w, n^o, n^g), \quad \alpha \in \{w, o, g\}. \quad (17)$$

In Appendix A, we present the expressions for the relative permeabilities in detail.

2.7. Viscosity

We describe the viscosities of the oil and the gas phase with the model by Lohrenz et al. [50], and we model the water viscosity by $(1/\mu^w)(\partial \mu^w / \partial P) = c_\mu^w$ where c_μ^w is the viscosibility of the water phase. Consequently, the viscosities are functions of temperature, pressure, and the phase compositions:

$$\mu^\alpha = \mu^\alpha(T, P, n^\alpha), \quad \alpha \in \{w, o, g\}. \quad (18)$$

In Appendix B, we describe the viscosity of the oil and gas phases in detail.

2.8. Thermodynamics

The phase equilibrium optimization problem (1), the fluid heat flux (6), and the fluid heat source (8) involve thermodynamical functions. We use the open-source thermodynamic software ThermoLib [45, 46] to evaluate the enthalpy, entropy, and volume of the fluid phases:

$$H^\alpha = H^\alpha(T, P, n^\alpha), \quad \alpha \in \{w, o, g\}, \quad (19a)$$

$$S^\alpha = S^\alpha(T, P, n^\alpha), \quad \alpha \in \{w, o, g\}, \quad (19b)$$

$$V^\alpha = V^\alpha(T, P, n^\alpha), \quad \alpha \in \{w, o, g\}. \quad (19c)$$

The thermodynamic model in ThermoLib is based on data and correlations from the DIPPR database [51] as well as cubic equations of state [21, 22, 52]. We use the Peng-Robinson equation of state [53]. The first order optimality conditions of the phase equilibrium problem (1) are algebraic equations. The gradient-based dynamic optimization algorithm described by Ritschel et al. [25] requires the Jacobian matrices of these algebraic equations. Consequently, the algorithm requires both the first and second order derivatives of the thermodynamic functions with respect to temperature, pressure, and mole numbers. The ThermoLib routines evaluate such derivatives based on the analytical expressions described by Ritschel et al. [45]. The thermodynamic properties of the rock, $H^r = H^r(T, P)$, $S^r = S^r(T, P)$, and $V^r = V^r(T, P)$ are also computed from an equation of state. We use a temperature-independent equation of state, $(1/V^r)(\partial V^r/\partial P) = c^r$, and we assume that the rock compressibility, c^r , is constant. We compute other thermodynamic functions with the fundamental thermodynamic relations $U^\alpha = H^\alpha - PV^\alpha$, $G^\alpha = H^\alpha - TS^\alpha$, and $A^\alpha = U^\alpha - TS^\alpha$ for $\alpha \in \{w, o, g, r\}$.

2.9. Finite volume discretization

The mass conservation equations (2) and the energy conservation equation (12) are in the form

$$\partial_t C = -\nabla \cdot \mathbf{N} + Q. \quad (20)$$

In this section, we describe the finite volume discretization of (20) and use it to discretize the mass and energy conservation equations. We consider a discretized reservoir that consists of a set of grid cells, $\{\Omega_i\}_{i \in \mathcal{N}}$, where \mathcal{N} is a set of grid cell indices. We assume that each grid cell is a polyhedron and that each face of the polyhedron is shared by exactly two cells. We integrate (20) over each of the grid cells and interchange integration and differentiation on the left-hand side:

$$\partial_t \int_{\Omega_i} C dV = - \int_{\Omega_i} \nabla \cdot \mathbf{N} dV + \int_{\Omega_i} Q dV, \quad i \in \mathcal{N}. \quad (21)$$

We apply Gauss' divergence theorem to the first integral on the right-hand side:

$$\int_{\Omega_i} \nabla \cdot \mathbf{N} dV = \int_{\partial\Omega_i} \mathbf{N} \cdot \mathbf{n} dA, \quad i \in \mathcal{N}. \quad (22)$$

$\partial\Omega_i$ is the boundary of the i 'th grid cell, and \mathbf{n} is the outward normal vector. We split up the boundary integral over each of the faces of the cell:

$$\int_{\partial\Omega_i} \mathbf{N} \cdot \mathbf{n} dA = \sum_{j \in \mathcal{N}^{(i)}} \int_{\gamma_{ij}} \mathbf{N} \cdot \mathbf{n} dA, \quad i \in \mathcal{N}. \quad (23)$$

$\mathcal{N}^{(i)}$ is the set of cells that share a face with the i 'th grid cell, and γ_{ij} is the face that is shared by the i 'th and the j 'th grid cell. We use quadrature to approximate the integral of the source term in (21) and the integral over γ_{ij} in (23):

$$\int_{\Omega_i} Q dV \approx (QV)_i, \quad i \in \mathcal{N}, \quad (24a)$$

$$\int_{\gamma_{ij}} \mathbf{N} \cdot \mathbf{n} dA \approx (\mathbf{AN} \cdot \mathbf{n})_{ij}, \quad i \in \mathcal{N}, \quad j \in \mathcal{N}^{(i)}. \quad (24b)$$

The subscript i indicates that a quantity is related to the i 'th grid cell while the subscript ij indicates that it is related to the face γ_{ij} . V_i is the volume of Ω_i , and A_{ij} is the area of γ_{ij} . We now apply the finite volume discretization to the mass and energy conservation equations. The integrals of the internal energy per unit volume and the concentrations are evaluated exactly:

$$\int_{\Omega_i} u dV = U_i, \quad i \in \mathcal{N}, \quad (25a)$$

$$\int_{\Omega_i} C_w dV = n_{w,i}, \quad i \in \mathcal{N}, \quad (25b)$$

$$\int_{\Omega_i} C_k dV = n_{k,i}, \quad i \in \mathcal{N}. \quad (25c)$$

The right-hand side of (24b) involves the flux evaluated at the center of the face which we approximate with a two-point flux approximation [54]. The resulting approximation of the right-hand side of (24b) for the heat and mass fluxes are

$$(\mathbf{AN}_u \cdot \mathbf{n})_{ij} \approx - \sum_{\alpha \in \{w, o, g\}} (h^\alpha \Gamma \hat{H}^\alpha \Delta \Phi^\alpha)_{ij} + (\Gamma_T \Delta T)_{ij}, \quad (26a)$$

$$(\mathbf{AN}^w \cdot \mathbf{n})_{ij} \approx -(\Gamma \hat{H}^w \Delta \Phi^w)_{ij}, \quad (26b)$$

$$(\mathbf{AN}_k \cdot \mathbf{n})_{ij} \approx -(x_k \Gamma \hat{H}^o \Delta \Phi^o + y_k \Gamma \hat{H}^g \Delta \Phi^g)_{ij}, \quad (26c)$$

for $i \in \mathcal{N}$ and $j \in \mathcal{N}^{(i)}$. The difference in temperature is $\Delta T_{ij} = T_j - T_i$. Γ_{ij} is the geometric part of the transmissibilities:

$$\Gamma_{ij} = A_{ij} (\hat{\Gamma}_{ij}^{-1} + \hat{\Gamma}_{ji}^{-1})^{-1}, \quad i \in \mathcal{N}, \quad j \in \mathcal{N}^{(i)}, \quad (27a)$$

$$\hat{\Gamma}_{ij} = \left(\mathbf{K}_i \frac{c_{ij} - c_i}{|c_{ij} - c_i|^2} \right) \cdot \mathbf{n}_{ij}, \quad i \in \mathcal{N}, \quad j \in \mathcal{N}^{(i)}. \quad (27b)$$

c_{ij} is the center of γ_{ij} , c_i is the center of Ω_i , and $\hat{\Gamma}_{ij}$ is the one-sided transmissibility. The expression for $\Gamma_{T,ij}$ is analogous to (27). However, the thermal conductivity of the rock replaces the permeability tensor:

$$\Gamma_{T,ij} = A_{ij} (\hat{\Gamma}_{T,ij}^{-1} + \hat{\Gamma}_{T,ji}^{-1})^{-1}, \quad i \in \mathcal{N}, \quad j \in \mathcal{N}^{(i)}, \quad (28a)$$

$$\hat{\Gamma}_{T,ij} = \left(k_{T,i}^r \frac{c_{ij} - c_i}{|c_{ij} - c_i|^2} \right) \cdot \mathbf{n}_{ij}, \quad i \in \mathcal{N}, \quad j \in \mathcal{N}^{(i)}. \quad (28b)$$

The difference in potential and the fluid part of the transmissibilities are

$$\Delta \Phi_{ij}^\alpha = (\Delta P - \rho^\alpha g \Delta z)_{ij}, \quad i \in \mathcal{N}, \quad j \in \mathcal{N}^{(i)}, \quad (29a)$$

$$\hat{H}_{ij}^\alpha = \begin{cases} (\rho^\alpha k_r^\alpha / \mu^\alpha)_i, & \Delta \Phi_{ij}^\alpha < 0, \\ (\rho^\alpha k_r^\alpha / \mu^\alpha)_j, & \Delta \Phi_{ij}^\alpha \geq 0, \end{cases} \quad i \in \mathcal{N}, \quad j \in \mathcal{N}^{(i)}. \quad (29b)$$

The differences in pressure and depth are $\Delta P_{ij} = P_j - P_i$ and $\Delta z_{ij} = z_j - z_i$. We approximate the density on the face center by $\rho_{ij}^\alpha \approx (\rho_i^\alpha + \rho_j^\alpha)/2$. In (29b), we have upwinded the fluid part of the transmissibilities to ensure numerical stability. Similarly, we upwind h^α in (26a) as well as x_k and y_k in (26c). The differential equations that result from the finite volume discretization

of the mass and energy conservation equations are

$$\dot{U}_i = \sum_{j \in \mathcal{N}^{(i)}} \left(\sum_{\alpha \in \{w,o,g\}} (h^\alpha \Gamma \hat{H}^\alpha \Delta \Phi^\alpha)_{ij} + (\Gamma_T \Delta T)_{ij} \right) + (Q_u V)_i, \quad (30a)$$

$$\dot{n}_{w,i} = \sum_{j \in \mathcal{N}^{(i)}} (\Gamma \hat{H}^w \Delta \Phi^w)_{ij} + (Q^w V)_i, \quad (30b)$$

$$\dot{n}_{k,i} = \sum_{j \in \mathcal{N}^{(i)}} (x_k \Gamma \hat{H}^o \Delta \Phi^o + y_k \Gamma \hat{H}^g \Delta \Phi^g)_{ij} + (Q_k V)_i, \quad (30c)$$

for $i \in \mathcal{N}$. The internal energy and the total amounts of moles on the left-hand side of (30) appear as specified quantities in the phase equilibrium problem described in Section 2.1.

2.10. The semi-explicit differential-algebraic form

The phase equilibrium problem described in Section 2.1 is in the form

$$\min_{y_i} f(y_i), \quad (31a)$$

$$\text{subject to } g(y_i) = x_i, \quad (31b)$$

$$h(y_i) = 0. \quad (31c)$$

$x_i = [U; n_w; n]_i \in \mathbb{R}^{2+N_c}$ is the state vector, and $y_i = [T; P; n^w; n^o; n^g]_i \in \mathbb{R}^{3+2N_c}$ is a vector of algebraic variables. The phase equilibrium conditions are the first order optimality conditions (also called Karush-Kuhn-Tucker or KKT conditions) of (31). The first order optimality conditions are a set of algebraic equations, $G_i(x_i, y_i, z_i) = 0$, because (31) does not contain inequality constraints [55]. $z_i \in \mathbb{R}^{3+N_c}$ are Lagrange multipliers.

The left-hand sides of the differential equations (30) are the time derivatives of the state variables while the quantities on the right-hand sides depend exclusively on the algebraic variables in the i 'th cell, y_i , and in neighbouring cells, $\{y_j\}_{j \in \mathcal{N}^{(i)}}$, as well as the manipulated inputs, $u_i = P_i^{\text{bhp}} \in \mathbb{R}$, and the disturbance variables, $d_i = T_i^{\text{inj}} \in \mathbb{R}$. The temperature of the injected water, T^{inj} , is used to evaluate the enthalpy of the injected water, $h^{w,\text{inj}}$. The manipulated inputs and the disturbance variables are only nonempty for cells that are perforated by a well. The differential equations (30) are thus in the form $\dot{x}_i(t) = F_i(y_i(t), \{y_j(t)\}_{j \in \mathcal{N}^{(i)}}, u_i(t), d_i(t))$. Consequently, the collection of the differential equations and phase equilibrium conditions for all grid cells is in the semi-explicit differential-algebraic form,

$$G(x(t), y(t), z(t)) = 0, \quad (32a)$$

$$\dot{x}(t) = F(y(t), u(t), d(t)). \quad (32b)$$

Furthermore, the algebraic equations are of index 1, i.e. $G_i(x_i, y_i, z_i) = 0$ can be solved for y_i and z_i when x_i is specified.

3. Isothermal and compositional reservoir flow model

In this section, we adapt the thermal and compositional model presented in Section 2 to isothermal systems. In isothermal systems, all involved thermal conductivities are infinite such that energy is transferred instantaneously between 1) the fluid and the rock, and 2) the rock and the surroundings, until thermal equilibrium is reached. Furthermore, the heat capacity of the surroundings is infinite such that their temperature is constant despite the supply or removal of energy. The key difference between the thermal and the isothermal model is the phase equilibrium problem which, for isothermal systems, is the VT flash. Furthermore, the isothermal model does not involve an energy conservation equation. However, the mass conservation equations in the two models are identical. Therefore, we only discuss 1) the phase equilibrium problem and 2) the semi-explicit differential-algebraic form of the model.

3.1. Phase equilibrium

The VT flash optimization problem describing the isochoric-isothermal chemical equilibrium is

$$\min_{P, n^w, n^o, n^g} A^w + A^o + A^g + A^r, \quad (33a)$$

$$\text{subject to } V^w + V^o + V^g + V^r = V, \quad (33b)$$

$$n^w = n_w, \quad (33c)$$

$$n_k^o + n_k^g = n_k, \quad k = 1, \dots, N_c. \quad (33d)$$

$A^\alpha = A^\alpha(T, P, n^\alpha)$ is the Helmholtz energy of phase $\alpha \in \{w, o, g, r\}$. The main differences between the VT flash (33) and the UV flash (1) are that in the VT flash 1) the Helmholtz energy is minimized, 2) there is no constraint on the internal energy, and 3) temperature is not an optimization variable.

3.2. The semi-explicit differential-algebraic form

The VT flash optimization problem is in the same form as the UV flash optimization problem, i.e. (31). The state variables are $x_i = [n_w; n]_i \in \mathbb{R}^{1+N_c}$, and the algebraic variables are $y_i = [P; n^w; n^o; n^g]_i \in \mathbb{R}^{2+2N_c}$. The VT flash contains one less equality constraint than the UV flash. Consequently, there is also one less Lagrange multiplier, i.e. $z_i \in \mathbb{R}^{2+N_c}$. The manipulated inputs remain unchanged. However, there are no disturbance variables in the isothermal model because the temperature of the injected water, T^{inj} , is constant. Consequently, the isothermal and compositional model is also in the semi-explicit differential-algebraic form (32).

4. Dynamic optimization

We consider the dynamic optimization problem

$$\min_{\{x(t); y(t); z(t)\}_0^{t_f}, \{u_k\}_{k=0}^{N-1}} \phi = \int_0^{t_f} \Phi(y(t), u(t), d(t)) dt, \quad (34a)$$

subject to

$$x(t_0) = \hat{x}_0, \quad (34b)$$

$$G(x(t), y(t), z(t)) = 0, \quad t \in [t_0, t_f], \quad (34c)$$

$$\dot{x}(t) = F(y(t), u(t), d(t)), \quad t \in [t_0, t_f], \quad (34d)$$

$$u(t) = u_k, \quad t \in [t_k, t_{k+1}[, \quad k = 0, \dots, N-1, \quad (34e)$$

$$d(t) = \hat{d}_k, \quad t \in [t_k, t_{k+1}[, \quad k = 0, \dots, N-1, \quad (34f)$$

$$\{u_k\}_{k=0}^{N-1} \in \mathcal{U}. \quad (34g)$$

$[x(t); y(t); z(t)]_{t_0}^{t_f}$ is a vector of dependent decision variables, and $\{u_k\}_{k=0}^{N-1}$ are independent decision variables. \hat{x}_0 is an estimate of the initial states, and $\{\hat{d}_k\}_{k=0}^{N-1}$ are predictions of the disturbance variables. Both \hat{x}_0 and $\{\hat{d}_k\}_{k=0}^{N-1}$ are parameters in the optimization problem. t_0 is the initial time, and $t_N = t_f$ is the final time. N is the number of control intervals. (34b) is a set of initial conditions for the semi-explicit differential-algebraic model equations (34c)-(34d). (34e)-(34f) are zero-order-hold (ZOH) parametrizations of the manipulated inputs and the disturbance variables. The constraints on the manipulated inputs (34g) are often bounds or linear constraints.

4.1. The dynamic optimization algorithm

We solve the dynamic optimization problem (34) with the gradient-based algorithm described by Ritschel et al. [25]. The algorithm is based on the single-shooting method which exploits that the initial value problem (34b)-(34d), subject to the ZOH parametrizations (34e)-(34f), determines the dependent decision variables, $[x(t); y(t); z(t)]_{t_0}^{t_f}$, when $\{u_k\}_{k=0}^{N-1}$, \hat{x}_0 , and $\{\hat{d}_k\}_{k=0}^{N-1}$ are specified. Consequently, the single-shooting method transcribes the infinite-dimensional dynamic optimization problem (34) to a finite-dimensional numerical optimization problem in which the objective function requires the solution of the initial value problem (also referred to as a simulation). Efficient algorithms for solving numerical optimization problems require the gradients of the objective function. The dynamic optimization algorithm computes these gradients with the adjoint method. Furthermore, it uses Euler's implicit method to discretize the differential equations (34d), and it solves the discretized differential equations and the algebraic equations in a simultaneous manner. The algorithm implements a simplified version of the time step selection scheme described by Völcker et al. [56], and it solves the involved linear systems with a block ILU(1) preconditioned GMRES method.

4.2. Implementation

We implement the dynamic optimization algorithm in C/C++. The implementation uses the open-source software DUNE for 1) grid management [42, 43] and 2) solution of linear systems with the preconditioned GMRES method [44]. The involved thermodynamic functions (and their first and second order derivatives) are computed with C routines from ThermoLib. We use an SLQP algorithm [55, Chap. 18], from the commercial optimization software KNITRO 10.2, to solve the involved numerical optimization problem. Furthermore, we use

C/C++ compilers from GCC. In Section 5, we present performance tests which are carried out on a 64-bit workstation with 15.6 GB memory and four Intel Core i7 3.60 GHz cores. The workstation uses the Ubuntu 16.04 operating system. Furthermore, it has a shared level 3 cache of 8192 KB, and each core has a 256 KB level 2 cache and a 64 KB level 1 cache.

5. Numerical examples

In this section, we present numerical examples of optimized thermal and isothermal waterflooding strategies computed with the dynamic optimization algorithm described in Section 4. Furthermore, we discuss the computational performance of the algorithm in terms of various key performance indicators (KPIs).

5.1. Optimized waterflooding strategies

We consider a $110 \times 110 \times 10$ m reservoir which is initially at 50°C . The oil and the gas phases consist of methane, ethane, propane, n-heptane, and hydrogen sulfide. We discretize the reservoir with an $11 \times 11 \times 1$ grid. The objective in the dynamic optimization of the thermal and isothermal waterflooding strategies is to maximize the total oil production over a three-year period. The decision variables are the BHPs of four injection wells and a single production well. There are 12 control intervals per year which results in a total of 36 control intervals, i.e. 36 decision variables per well. The locations of the wells are shown in Fig. 3 together with the heterogeneous (and isotropic) permeability field. For simplicity, we assume that there is no heat loss to the surroundings, that the rock is incompressible, and that the porosity field is homogeneous. The porosity is 0.25. The thermal heat capacity of the rock is $0.92 \text{ kJ}/(\text{kg} \cdot \text{K})$, and the thermal conductivity is $2.5 \text{ W}/(\text{m} \cdot \text{K})$ which resemble the properties of sandstone [57, Chap. 2]. The BHPs of the injectors are constrained to the interval $[10 \text{ MPa}, 12 \text{ MPa}]$, and the BHP of the producer is constrained to the interval $[9 \text{ MPa}, 10 \text{ MPa}]$. The injected water is at 90°C in the thermal strategy and at 50°C in the isothermal strategy.

Fig. 4 shows the injector and producer BHPs of the optimized thermal and isothermal strategies together with the cumulative volumetric injection of water, production of oil, and production of gas. Both strategies operate the producer close to the minimum BHP. Furthermore, they operate injector 3 and 4 close to the maximum BHP because they are located in very impermeable areas. The most significant differences between the two strategies are the BHPs of injector 1 and 2 which they both vary significantly. Compared to the isothermal strategy, the thermal strategy 1) injects slightly less water, 2) produces slightly less oil, and 3) produces slightly more gas. Fig. 5 and 6 illustrate the thermal and isothermal waterflooding processes in terms of the pressure and the oil and gas saturations in the reservoir. Fig. 5 also shows the temperature in the reservoir. The two figures suggest that it is challenging for the dynamic optimization algorithm to compute strategies that completely deplete the upper half of the reservoir during the three years of production.

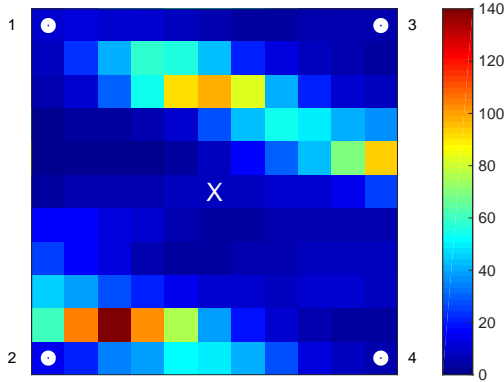


Figure 3: The permeability field [mD] and the locations of the injection and production wells. The white circles indicate the locations of the injectors, and the white X indicates the location of the producer.

In this example, we have considered an optimized thermal waterflooding strategy. However, thermal oil recovery strategies most often involve the injection of steam, e.g. steam-assisted gravity drainage (SAGD) is used to produce heavy oil in Canada and Venezuela [58]. In such cases, the temperature of the injected steam would be a manipulated input together with the well BHP, and the objective function should include the cost of heating the steam.

5.2. Key performance indicators

Table 1 shows a number of key problem characteristics and KPIs for the dynamic optimization of the thermal and isothermal waterflooding strategies. The number of manipulated inputs, i.e. independent decision variables, is the same for both strategies. However, the thermal model contains more differential equations and algebraic equations. The optimization of the thermal strategy involves 15 iterations in the SLQP optimization algorithm, whereas the optimization of the isothermal strategy only requires 6. However, the computation time per iteration is slightly higher for the isothermal strategy because each iteration, on average, involves more simulations. The number of gradient evaluations per iteration is close to 1 for both strategies.

The average number of time steps in the thermal and isothermal simulations are close to each other. However, the simulations of the thermal waterflooding strategies require, on average, close to one more Newton iteration. Consequently, the number of evaluations of the functions F and G in the semi-explicit differential-algebraic equations (34c)-(34d), and their Jacobians, is also approximately one higher. Furthermore, the GMRES algorithm requires close to 0.8 more iterations, on average, to solve the linear systems in the simulations of the thermal strategy. In conclusion, it is more computationally demanding to simulate, and therefore also optimize, the thermal waterflooding strategy than the isothermal strategy.

6. Conclusions

In this work, we consider dynamic optimization of thermal and isothermal oil recovery processes. Therefore, we present

Table 1: Problem characteristics and KPIs for the dynamic optimization of the thermal and isothermal waterflooding strategies. The function evaluations refer to the evaluation of the functions F and G in the semi-explicit differential-algebraic equations (34c)-(34d), and the Jacobian evaluations refer to the evaluation of the Jacobians of F and G . The iterations per linear system refer to the iterations in the GMRES algorithm. The KPIs related to the simulations are averaged over all the simulations involved in the dynamic optimization.

	Thermal	Isothermal
Problem		
Manipulated inputs	180	180
Differential equations	847	726
Algebraic equations	2541	2299
Optimization		
Iterations	15	6
Simulations	27	20
Gradient evaluations	16	7
CPU time (s)	661.16	287.18
CPU time per iteration (s)	44.08	47.86
Simulation		
Time steps per simulation	218.37	212.55
Newton iterations per time step	4.04	3.02
Function evaluations per time step	6.07	5.07
Jacobian evaluations per time step	5.67	4.41
Iterations per linear system	9.03	8.21

thermodynamically rigorous models of thermal and isothermal multicomponent three-phase flow in subsurface oil reservoirs. The involved phase equilibrium problems, i.e. the UV and the VT flash, are based on the second law of thermodynamics. Furthermore, we formulate the UV and VT flash problems as optimization problems. We demonstrate that the thermal and the isothermal reservoir flow models are in a semi-explicit index-1 differential-algebraic form, and we use a gradient-based algorithm to solve the dynamic optimization problems. We implement the algorithm in C/C++ using the software libraries DUNE, ThermoLib, and KNITRO. Finally, we present numerical examples of optimized thermal and isothermal oil recovery strategies, and we discuss the computational performance of the dynamic optimization algorithm.

7. Acknowledgements

We would like to acknowledge the financial support from Innovation Fund Denmark in the OPTION project (63-2013-3).

References

- [1] T. T. Garipov, P. Tomin, R. Rin, D. V. Voskov, H. A. Tchelepi, Unified thermo-compositional-mechanical framework for reservoir simulation, *Computational Geosciences* 22 (4) (2018) 1039–1057.
- [2] D. Kourounis, L. J. Durlofsky, J. D. Jansen, K. Aziz, Adjoint formulation and constraint handling for gradient-based optimization of compositional reservoir flow, *Computational Geosciences* 18 (2) (2014) 117–137.
- [3] O. Polívka, J. Mikyška, Compositional modeling in porous media using constant volume flash and flux computation without the need for phase identification, *Journal of Computational Physics* 272 (2014) 149–169.

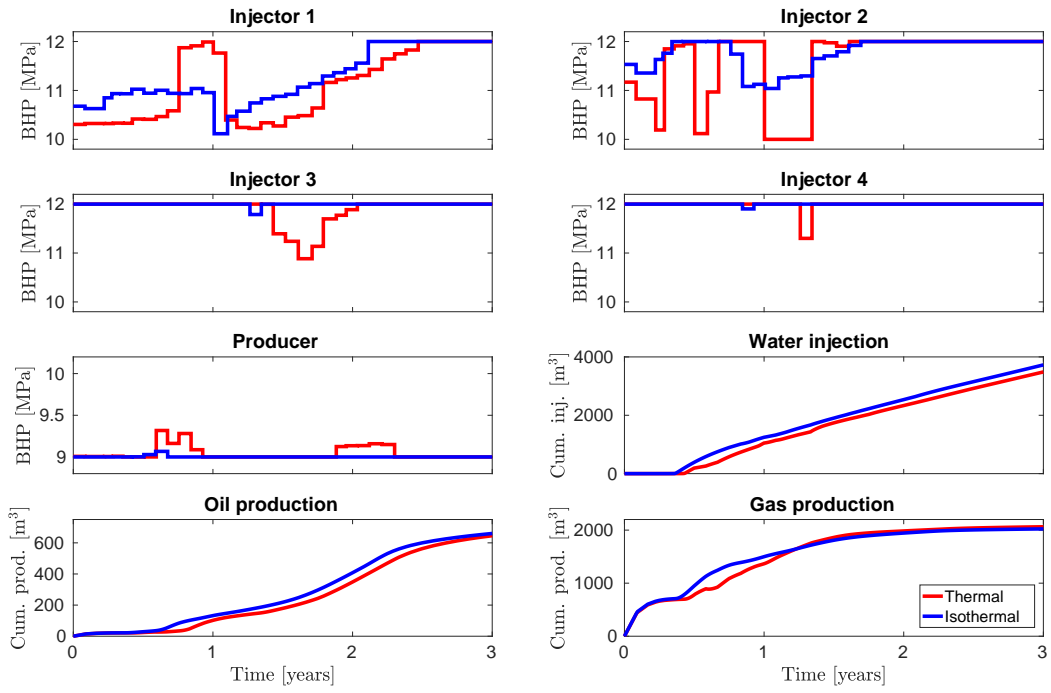


Figure 4: Optimized BHPs of the injectors and the cumulative volumetric water injection, oil production, and gas production.

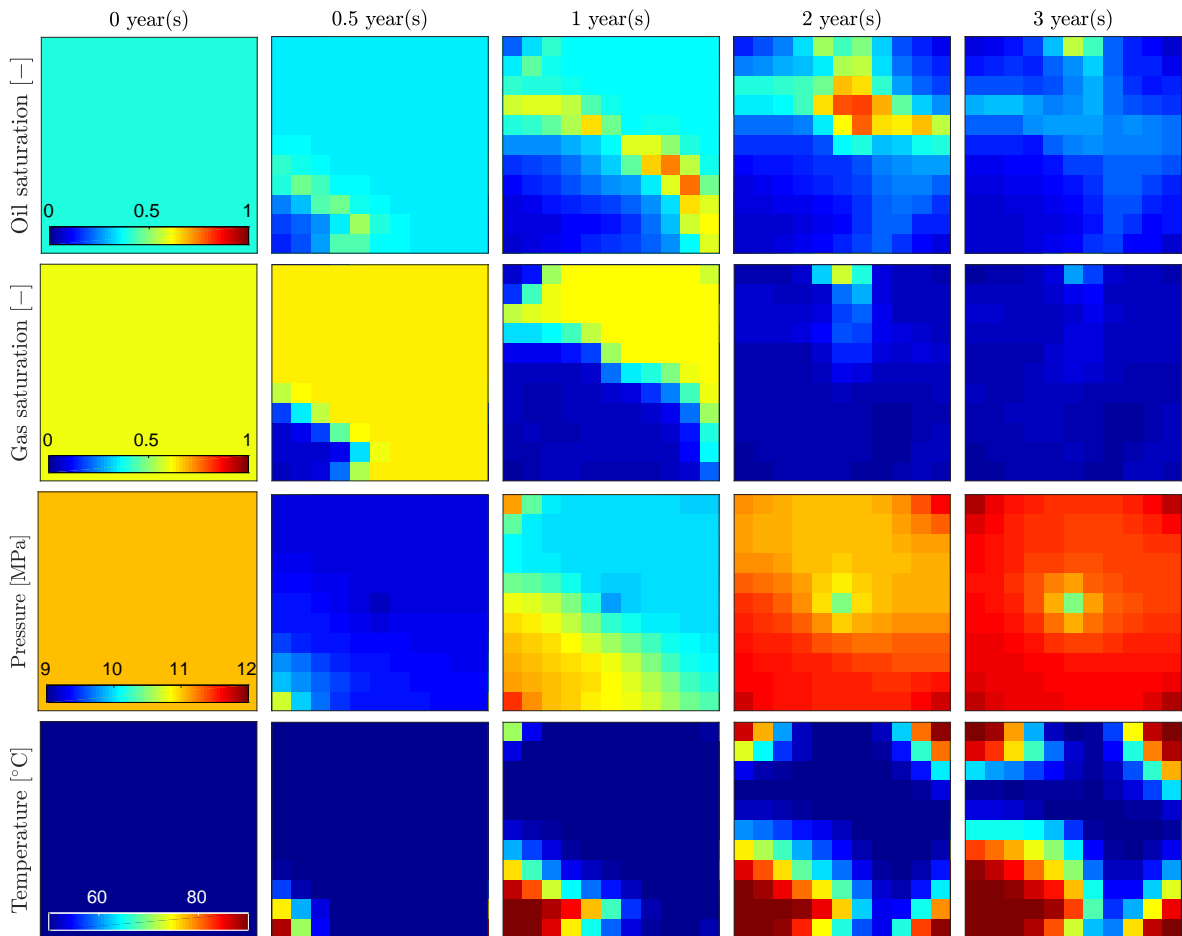


Figure 5: Optimized thermal waterflooding (90°C water injected).

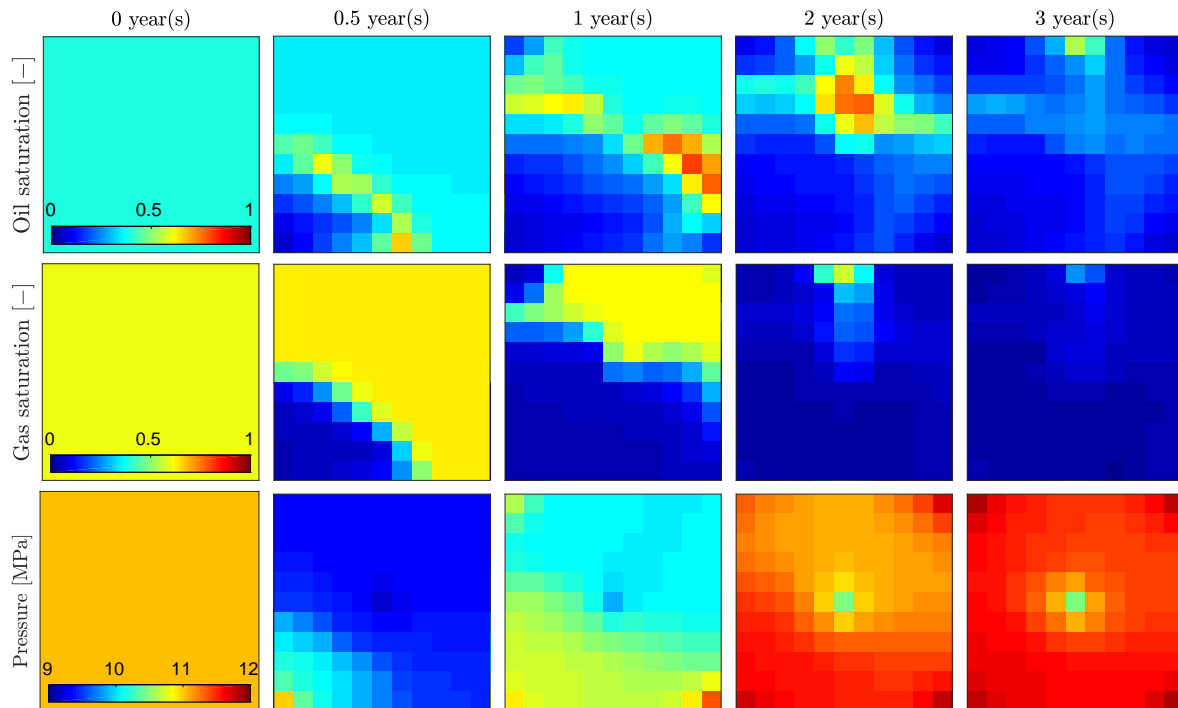


Figure 6: Optimized isothermal waterflooding (50°C water injected).

- [4] T. K. S. Ritschel, J. B. Jørgensen, Production optimization of a rigorous thermal and compositional reservoir flow model, *IFAC-PapersOnLine* 51 (8) (2018) 76–81.
- [5] T. K. S. Ritschel, J. B. Jørgensen, Production optimization of thermodynamically rigorous isothermal and compositional models, in: *Proceedings of the 16th European Conference on the Mathematics of Oil Recovery*, Barcelona, Spain, 2018.
- [6] R. Zaydullin, D. V. Voskov, S. C. James, H. Henley, A. Lucia, Fully compositional and thermal reservoir simulation, *Computers and Chemical Engineering* 63 (2014) 51–65.
- [7] L. Thorvaldsson, H. Palsson, Modeling liquid dominated two phase flow in geothermal reservoirs in vicinity to, and inside wells, in: *Proceedings of the 37th Workshop on Geothermal Reservoir Engineering*, Stanford, California, USA, 2012.
- [8] W. Jang, M. M. Aral, Multiphase flow fields in in-situ air sparging and its effect on remediation, *Transport in Porous Media* 76 (2009) 99–119.
- [9] R. Hannaoui, P. Horgue, F. Larachi, Y. Haroun, F. Augier, M. Quintard, M. Prat, Pore-network modeling of trickle bed reactors: pressure drop analysis, *Chemical Engineering Journal* 262 (2015) 334–343.
- [10] T. Berning, M. Odgaard, S. K. Kær, A computational analysis of multiphase flow through PEMFC cathode porous media using the multifluid approach, *Journal of the Electrochemical Society* 156 (11) (2009) B1301–B1311.
- [11] J.-P. Kone, X. Zhang, Y. Yan, G. Hu, G. Ahmadi, Three-dimensional multiphase flow computational fluid dynamics models for proton exchange membrane fuel cell: a theoretical development, *The Journal of Computational Multiphase Flows* 9 (1) (2017) 3–25.
- [12] M. I. H. Khan, M. U. H. Joardder, C. Kumar, M. A. Karim, Multiphase porous media modelling: a novel approach to predicting food processing performance, *Critical Reviews in Food Science and Nutrition* 58 (4) (2018) 528–546.
- [13] F. Pesavento, B. A. Schrefler, G. Sciumè, Multiphase flow in deforming porous media: a review, *Archives of Computational Methods in Engineering* 24 (2017) 423–448.
- [14] Z. Chen, *Reservoir simulation: mathematical techniques in oil recovery*, SIAM, 2007.
- [15] Z. Chen, G. Huan, Y. Ma, *Computational methods for multiphase flows in porous media*, Computational Science & Engineering, SIAM, 2006.
- [16] M. L. Michelsen, State function based flash specifications, *Fluid Phase Equilibria* 158-160 (1999) 617–626.
- [17] B. Colson, P. Marcotte, G. Savard, An overview of bilevel optimization, *Annals of Operations Research* 153 (2007) 235–256.
- [18] Z.-Q. Luo, J.-S. Pang, D. Ralph, *Mathematical programs with equilibrium constraints*, Cambridge University Press, 1996.
- [19] J. Outrata, M. Kočvara, J. Zowe, *Nonsmooth approach to optimization problems with equilibrium constraints: theory, applications and numerical results*, Vol. 28 of *Nonconvex Optimization and Its Applications*, Springer Science & Business Media, 1998.
- [20] H. B. Callen, *Thermodynamics and an introduction to thermostatistics*, 2nd Edition, John Wiley & Sons, 1985.
- [21] M. D. Koretsky, *Engineering and chemical thermodynamics*, 2nd Edition, Wiley, 2013.
- [22] J. M. Smith, H. C. Van Ness, M. M. Abbott, *Introduction to chemical engineering thermodynamics*, 7th Edition, McGraw-Hill, 2005.
- [23] T. K. S. Ritschel, J. B. Jørgensen, Computation of phase equilibrium and phase envelopes, Tech. Rep. DTU Compute Technical Report-2017-11, Department of Applied Mathematics and Computer Science, Technical University of Denmark (2017).
- [24] T. K. S. Ritschel, J. B. Jørgensen, Computation of phase equilibrium in reservoir simulation and optimization, *IFAC-PapersOnLine* 51 (8) (2018) 94–101.
- [25] T. K. S. Ritschel, A. Capolei, J. Gaspar, J. B. Jørgensen, An algorithm for gradient-based dynamic optimization of UV flash processes, *Computers and Chemical Engineering* 114 (2018) 281–295.
- [26] V. Bukshynov, O. Volkov, L. J. Durlofsky, K. Aziz, Comprehensive framework for gradient-based optimization in closed-loop reservoir management, *Computational Geosciences* 19 (4) (2015) 877–897.
- [27] A. Capolei, C. Völcker, J. Frydendall, J. B. Jørgensen, Oil reservoir production optimization using single shooting and ESDIRK methods, *IFAC Proceedings Volumes* 45 (8) (2012) 286–291.
- [28] A. Cudas, K. G. Hanssen, B. Foss, A. Capolei, J. B. Jørgensen, Multiple shooting applied to robust reservoir control optimization including output constraints on coherent risk measures, *Computational Geosciences* 21 (3) (2017) 479–497.
- [29] T. A. N. Heirung, M. R. Wartmann, J. D. Jansen, B. E. Ydstie, B. A. Foss, Optimization of the water-flooding process in a small 2D horizontal oil reservoir by direct transcription, *IFAC Proceedings Volumes* 44 (1) (2011) 10863–10868.

- [30] M. Simon, M. Ulbrich, Adjoint based optimal control of partially miscible two-phase flow in porous media with applications to CO₂ sequestration in underground reservoirs, *Optimization and Engineering* 16 (2015) 103–130.
- [31] Y. Lei, S. Li, X. Zhang, Q. Zhang, Optimal control of polymer flooding for high temperature and high salinity reservoir, *International Journal of Advancements in Computing Technology* 4 (12) (2012) 52–60.
- [32] Z. Xiao-Dong, L. Shu-rong, Optimal control solving of polymer flooding in enhanced oil recovery with 2-D models, in: *Proceedings of the 2007 IEEE International Conference on Control and Automation*, Guangzhou, China, 2007, pp. 1981–1986.
- [33] T. K. S. Ritschel, A. Capolei, J. B. Jørgensen, The adjoint method for gradient-based dynamic optimization of UV flash processes, *Computer Aided Chemical Engineering* 40 (2017) 2071–2076.
- [34] T. K. S. Ritschel, A. Capolei, J. B. Jørgensen, Dynamic optimization of UV flash processes, in: *FOCAPO / CPC 2017*, Tucson, Arizona, 2017.
- [35] T. Binder, L. Blank, H. G. Bock, R. Bulirsch, W. Dahmen, M. Diehl, T. Kroneder, W. Marquardt, J. P. Schlöder, O. von Stryk, Introduction to model based optimization of chemical processes on moving horizons, in: *Online Optimization of Large Scale Systems*, Springer-Verlag Berlin Heidelberg, 2001, pp. 295–339.
- [36] F. Forouzanfar, E. D. Rossa, R. Russo, A. C. Reynolds, Life-cycle production optimization of an oil field with an adjoint-based gradient approach, *Journal of Petroleum Science and Engineering* 112 (2013) 351–358.
- [37] J. B. Jørgensen, Adjoint sensitivity results for predictive control, state- and parameter-estimation with nonlinear models, in: *Proceedings of the 2007 European Control Conference*, Kos, Greece, 2007, pp. 3649–3656.
- [38] C. Völcker, J. B. Jørgensen, E. H. Stenby, Oil reservoir production optimization using optimal control, in: *Proceedings of the 50th IEEE Conference on Decision and Control and European Control Conference*, Orlando, Florida, USA, 2011, pp. 7937–7943.
- [39] M. R. Kristensen, J. B. Jørgensen, P. G. Thomsen, M. L. Michelsen, S. B. Jørgensen, Sensitivity analysis in index-1 differential algebraic equations by ESDIRK methods, *IFAC Proceedings Volumes* 38 (1) (2005) 212–217.
- [40] H. Zhao, Y. W. Tang, Y. Li, Y. B. Shi, L. Cao, R. X. Gong, G. H. Shang, Reservoir production optimization using general stochastic approximate algorithm under the mixed-linear-nonlinear constraints, *Journal of Residuals Science & Technology* 13 (8) (2016) 154.1–154.15.
- [41] J. E. Onwunali, L. J. Durlafsky, Application of a particle swarm optimization algorithm for determining optimum well location and type, *Computational Geosciences* 14 (1) (2010) 183–198.
- [42] P. Bastian, M. Blatt, A. Dedner, C. Engwer, R. Klöfkor, R. Kornhuber, M. Ohlberger, O. Sander, A generic grid interface for parallel and adaptive scientific computing. Part II: implementation and tests in DUNE, *Computing* 82 (2-3) (2008) 121–138.
- [43] P. Bastian, M. Blatt, A. Dedner, C. Engwer, R. Klöfkor, M. Ohlberger, O. Sander, A generic grid interface for parallel and adaptive scientific computing. Part I: abstract framework, *Computing* 82 (2-3) (2008) 103–119.
- [44] M. Blatt, P. Bastian, The iterative solver template library, in: B. Kågström et al. (Ed.), *Applied Parallel Computing. State of the Art in Scientific Computing. PARA 2006*, Vol. 4699 of *Lecture Notes in Computer Science*, Springer-Verlag Berlin Heidelberg, 2007, pp. 666–675.
- [45] T. K. S. Ritschel, J. Gaspar, A. Capolei, J. B. Jørgensen, An open-source thermodynamic software library, *Tech. Rep. DTU Compute Technical Report-2016-12*, Department of Applied Mathematics and Computer Science, Technical University of Denmark (2016).
- [46] T. K. S. Ritschel, J. Gaspar, J. B. Jørgensen, A thermodynamic library for simulation and optimization of dynamic processes, *IFAC-PapersOnLine* 50 (1) (2017) 3542–3547.
- [47] A. Wächter, L. T. Biegler, On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming, *Mathematical Programming* 106 (1) (2006) 25–57.
- [48] J. P. Holman, *Heat transfer*, 10th Edition, McGraw-Hill, 2010.
- [49] M. Delshad, G. A. Pope, Comparison of the three-phase oil relative permeability models, *Transport in Porous Media* 4 (1) (1989) 59–83.
- [50] J. Lohrenz, B. G. Bray, C. R. Clark, Calculating viscosities of reservoir fluids from their compositions, *Journal of Petroleum Technology* 16 (10) (1964) 1171–1176.
- [51] G. H. Thomson, The DIPPR® databases, *International Journal of Thermophysics* 17 (1) (1996) 223–232.
- [52] J. Gmehling, B. Kolbe, M. Kleiber, J. Rarey, *Chemical thermodynamics for process simulation*, Wiley-VCH, 2012.
- [53] D.-Y. Peng, D. B. Robinson, A new two-constant equation of state, *Industrial & Engineering Chemistry Fundamentals* 15 (1) (1976) 59–64.
- [54] K.-A. Lie, *An introduction to reservoir simulation using MATLAB*, Sintef ICT, Oslo, Norway, 2014.
- [55] J. Nocedal, S. J. Wright, *Numerical optimization*, 2nd Edition, Springer Science & Business Media, 2006.
- [56] C. Völcker, J. B. Jørgensen, P. G. Thomsen, E. H. Stenby, Adaptive step-size control in implicit Runge-Kutta methods for reservoir simulation, *IFAC Proceedings Volumes* 43 (5) (2010) 523–528.
- [57] L. Eppelbaum, I. Kutasov, A. Pilchin, *Applied geothermics*, Lecture Notes in Earth System Sciences, Springer-Verlag Berlin Heidelberg, 2014.
- [58] V. Alvarado, E. Manrique, Enhanced oil recovery: an update review, *Energies* 3 (2010) 1529–1575.
- [59] H. M. Goda, P. Behrenbruch, Using a modified Brooks-Corey model to study oil-water relative permeability for diverse pore structures, in: *Proceedings of the 2004 SPE Asia Pacific Oil and Gas Conference and Exhibition*, Perth, Australia, 2004.
- [60] J. A. Jossi, L. I. Stiel, G. Thodos, The viscosity of pure substances in the dense gaseous and liquid phases, *AIChE Journal* 8 (1) (1962) 59–63.

Appendix A. Relative permeability

In this appendix, we use Stone’s model II [49] to describe the relative permeabilities. The model equations involve the relative permeabilities of 1) a hypothetical oil-water system and 2) a hypothetical oil-gas system. We express the relative permeabilities for these two hypothetical systems using the modified Brooks-Corey model [59]. We introduce the normalized saturations, $\bar{S}^w = \bar{S}^w(T, P, n^w, n^o, n^g)$ and $\bar{S}^g = \bar{S}^g(T, P, n^w, n^o, n^g)$:

$$\bar{S}^w = (\hat{S}^w - \hat{S}_c^w) / (1 - \hat{S}_c^w - \hat{S}_{\max}^w), \quad (\text{A.1a})$$

$$\bar{S}^g = (\hat{S}^g - \hat{S}_c^g) / (1 - \hat{S}_c^g - \hat{S}_{\max}^g). \quad (\text{A.1b})$$

$\hat{S}^w = \hat{S}^w(T, P, n^w, n^o, n^g)$ and $\hat{S}^g = \hat{S}^g(T, P, n^w, n^o, n^g)$ are the water and gas saturations, \hat{S}_c^w and \hat{S}_c^g are the connate water and gas saturations, and \hat{S}_{\max}^w and \hat{S}_{\max}^g are the maximum water and gas saturations. The relative permeabilities of the hypothetical oil-water system, $k_r^w = k_r^w(T, P, n^w, n^o, n^g)$ and $k_r^{ow} = k_r^{ow}(T, P, n^w, n^o, n^g)$, are

$$k_r^w = k_{r,0}^w (\bar{S}^w)^{m_w}, \quad (\text{A.2a})$$

$$k_r^{ow} = k_{r,0}^{ow} (1 - \bar{S}^w)^{m_{ow}}, \quad (\text{A.2b})$$

and similarly, the relative permeabilities of the hypothetical oil-gas system, $k_r^g = k_r^g(T, P, n^w, n^o, n^g)$ and $k_r^{og} = k_r^{og}(T, P, n^w, n^o, n^g)$, are

$$k_r^g = k_{r,0}^g (\bar{S}^g)^{m_g}, \quad (\text{A.2c})$$

$$k_r^{og} = k_{r,0}^{og} (1 - \bar{S}^g)^{m_{og}}, \quad (\text{A.2d})$$

where $k_{r,0}^w$, $k_{r,0}^{ow}$, $k_{r,0}^g$, and $k_{r,0}^{og}$ are the end-point relative permeabilities, and m_w , m_{ow} , m_g , and m_{og} are the Corey exponents. If one of the expressions in (A.2) become negative or larger than one, the corresponding relative permeability is set to zero or one, respectively. Finally, the relative permeability of the oil phase, $k_r^o = k_r^o(T, P, n^w, n^o, n^g)$, is

$$k_r^o = k_r^c ((k_r^{ow} / k_r^c + k_r^w) (k_r^{og} / k_r^c + k_r^g) - (k_r^w + k_r^g)), \quad (\text{A.3})$$

where k_c^c is a parameter. To summarize, the relative permeabilities of the water, oil, and gas phases are given by (A.2a), (A.3), and (A.2c), respectively.

Appendix B. Viscosity of oil and gas

In this appendix, we describe the model of the viscosity of reservoir fluids by Lohrenz et al. [50]. We use the expressions for the liquid phase viscosity to describe the viscosity of both the oil and the gas phase. The viscosity of phase $\alpha \in \{o, g\}$, $\mu^\alpha = \mu^\alpha(T, P, n^\alpha)$, is a function of temperature, T , pressure, P , and phase composition (in moles), n^α :

$$\mu^\alpha = \bar{\mu}^\alpha + \frac{1}{\tau^\alpha} \left((a^\alpha)^4 - 10^{-4} \right), \quad \alpha \in \{o, g\}. \quad (\text{B.1})$$

The auxiliary variables $\tau^\alpha = \tau^\alpha(n^\alpha)$ and $a^\alpha = a^\alpha(T, P, n^\alpha)$ are

$$\tau^\alpha = (T_c^\alpha)^{\frac{1}{6}} (M_w^\alpha)^{-\frac{1}{2}} (P_c^\alpha)^{-\frac{2}{3}}, \quad \alpha \in \{o, g\}, \quad (\text{B.2a})$$

$$a^\alpha = \sum_{i=0}^4 a_i (\rho_r^\alpha)^i, \quad \alpha \in \{o, g\}. \quad (\text{B.2b})$$

The coefficients in the polynomial in (B.2b) are $a_0 = 0.1023$, $a_1 = 0.023364$, $a_2 = 0.058533$, $a_3 = -0.040758$, and $a_4 = 0.0093324$ [60]. The auxiliary variables $T_c^\alpha = T_c^\alpha(n^\alpha)$, $P_c^\alpha = P_c^\alpha(n^\alpha)$, $V_c^\alpha = V_c^\alpha(n^\alpha)$, and $M_w^\alpha = M_w^\alpha(n^\alpha)$ are

$$T_c^\alpha = \sum_{k=1}^{N_C} x_k^\alpha T_{c,k}, \quad \alpha \in \{o, g\}, \quad (\text{B.3a})$$

$$P_c^\alpha = \sum_{k=1}^{N_C} x_k^\alpha P_{c,k}, \quad \alpha \in \{o, g\}, \quad (\text{B.3b})$$

$$V_c^\alpha = \sum_{k=1}^{N_C} x_k^\alpha V_{c,k}, \quad \alpha \in \{o, g\}, \quad (\text{B.3c})$$

$$M_w^\alpha = \sum_{k=1}^{N_C} x_k^\alpha M_{w,k}, \quad \alpha \in \{o, g\}. \quad (\text{B.3d})$$

We present the expression for V_c^α here, but we first use it in (B.6). We use values of the pure component critical temperature, $T_{c,k}$, critical pressure, $P_{c,k}$, critical volume, $V_{c,k}$, and molecular weight, $M_{w,k}$, from the DIPPR database [51]. In order to describe the viscosity of the oil and the gas phases in a unified manner, we have adopted a different notation for the mole fractions, $x_k^\alpha = x_k^\alpha(n^\alpha)$, than in previous sections:

$$x_k^\alpha = \frac{n_k^\alpha}{N^\alpha}, \quad k = 1, \dots, N_C, \quad \alpha \in \{o, g\}. \quad (\text{B.4})$$

The total amount of moles in phase α , $N^\alpha = N^\alpha(n^\alpha)$, is

$$N^\alpha = \sum_{k=1}^{N_C} n_k^\alpha, \quad \alpha \in \{o, g\}. \quad (\text{B.5})$$

The reduced density, $\rho_r^\alpha = \rho_r^\alpha(T, P, n^\alpha)$, is

$$\rho_r^\alpha = \rho^\alpha V_c^\alpha, \quad \alpha \in \{o, g\}, \quad (\text{B.6})$$

where the molar density, $\rho^\alpha = \rho^\alpha(T, P, n^\alpha)$, is

$$\rho^\alpha = \frac{N^\alpha}{V^\alpha}, \quad \alpha \in \{o, g\}. \quad (\text{B.7})$$

The reference viscosity, $\bar{\mu}^\alpha = \bar{\mu}^\alpha(T, n^\alpha)$, is

$$\bar{\mu}^\alpha = \frac{\bar{M}_{w\mu}^\alpha}{M_w^\alpha}, \quad \alpha \in \{o, g\}, \quad (\text{B.8})$$

where the auxiliary variables $\bar{M}_{w\mu}^\alpha = \bar{M}_{w\mu}^\alpha(T, n^\alpha)$ and $M_w^\alpha = M_w^\alpha(n^\alpha)$ are

$$\bar{M}_{w\mu}^\alpha = \sum_{k=1}^{N_C} x_k^\alpha \bar{\mu}_k \sqrt{M_{w,k}}, \quad \alpha \in \{o, g\}, \quad (\text{B.9a})$$

$$M_w^\alpha = \sum_{k=1}^{N_C} x_k^\alpha \sqrt{M_{w,k}}, \quad \alpha \in \{o, g\}. \quad (\text{B.9b})$$

The pure component reference viscosity, $\bar{\mu}_k = \bar{\mu}_k(T)$, is

$$\bar{\mu}_k = \begin{cases} 34 \cdot 10^{-5} \frac{T_{r,k}^{0.94}}{\tau_k}, & T_{r,k} < 1.5, \\ 17.78 \cdot 10^{-5} \frac{(4.58T_{r,k} - 1.67)^{\frac{5}{8}}}{\tau_k}, & T_{r,k} \geq 1.5, \end{cases} \quad (\text{B.10})$$

for $k = 1, \dots, N_C$ where the reduced temperature, $T_{r,k} = T_{r,k}(T)$, and the auxiliary variable τ_k are

$$T_{r,k} = \frac{T}{T_{c,k}}, \quad k = 1, \dots, N_C, \quad (\text{B.11a})$$

$$\tau_k = T_{c,k}^{\frac{1}{6}} M_{w,k}^{-\frac{1}{2}} P_{c,k}^{-\frac{2}{3}}, \quad k = 1, \dots, N_C. \quad (\text{B.11b})$$

Paper XII

Nonlinear Model Predictive Control for Disturbance Rejection in UV Flash Processes

Authors:

Tobias K. S. Ritschel and John Bagterp Jørgensen

To be submitted

Nonlinear Model Predictive Control for Disturbance Rejection in UV Flash Processes

Tobias K. S. Ritschel and John Bagterp Jørgensen

Abstract—We present a nonlinear model predictive control (NMPC) algorithm for semi-explicit index-1 stochastic differential-algebraic equations. It is natural to model UV flash processes with such equations. The algorithm uses the continuous-discrete extended Kalman filter (EKF) for state estimation, and it uses a single-shooting method to solve the involved optimal control problems. It computes the gradients with an adjoint method. The UV flash is important to rigorous models of phase equilibrium processes because it is a mathematical statement of the second law of thermodynamics. NMPC algorithms for UV flash processes are therefore relevant to both safe and economical operation of phase equilibrium processes such as flash separation, distillation, two-phase flow in pipes, and oil production. We design the NMPC algorithm for disturbance rejection, and we therefore augment the state vector with the unknown disturbance variables in the continuous-discrete EKF. We present a numerical example of economical NMPC of a UV flash separation process. It involves output constraints and the estimation of an unknown and unmeasured disturbance. The computation time of the NMPC algorithm does not exceed 74 s in any of the 5 min control intervals. This indicates that real-time NMPC of UV flash separation processes is computationally feasible.

I. INTRODUCTION

Nonlinear model predictive control (NMPC) algorithms compute a closed-loop feedback control strategy using the moving horizon optimization principle, i.e. by solving a sequence of open-loop optimal control problems (OCPs) [1]. The objective of NMPC algorithms is either to optimize the economics of the process or to minimize the distance to predefined setpoints. NMPC algorithms are used to control a variety of chemical processes [2] including stirred tank reactors [3], batch reactors [4], fermentors [5], distillation [6], [7], and production from oil reservoirs [8], [9].

Chemical processes often involve thermodynamic equilibrium between vapor and liquid phases. The conditions for equilibrium between phases are derived from the second law of thermodynamics. It states that the entropy of a closed system in equilibrium is maximal. The UV flash is a mathematical statement of the second law of thermodynamics and is therefore key to rigorous modeling of phase equilibrium processes. The UV flash has been used in models of flash separation [10]–[12], distillation [13], and computational fluid dynamical processes [14], [15]. The UV flash problem can be formulated as an optimization problem

with equality constraints [16]. The solution to the optimization problem consists of the temperature, pressure, and vapor-liquid compositions that maximize the entropy while satisfying constraints on the internal energy, U , the volume, V , and the total composition, n . The first-order optimality conditions of the UV flash optimization problem are a set of algebraic equations which constitute the phase equilibrium conditions. It is therefore natural to model dynamic UV flash processes with differential-algebraic equations (DAEs). Algorithms for dynamic optimization and state estimation of UV flash processes have been considered recently [17], [18], but NMPC of such processes has not yet been addressed.

NMPC algorithms combine state estimation algorithms with dynamic optimization algorithms. The extended Kalman filter (EKF) is commonly used for state estimation of nonlinear processes [19], [20]. Alternatives include the unscented Kalman filter and particle filters [21] as well as the ensemble Kalman filter [22], [23], moving-horizon estimation [24], and neural network-based algorithms [25]. Optimal control problems are often solved with direct methods [1], i.e. single-shooting, multiple-shooting, or simultaneous collocation. Computationally efficient implementation of such methods requires the evaluation of gradients. The gradients can be computed with an adjoint method [26] or with a forward method [27] in the single- and multiple-shooting approaches. NMPC algorithms for disturbance rejection either 1) estimate the disturbance variables [28]–[30] or 2) represent the uncertainty in the disturbance variables with a scenario-tree [31].

In this work, we present an NMPC algorithm for disturbance rejection in UV flash processes. The algorithm combines the continuous-discrete EKF with single-shooting and an adjoint method for computing gradients. We model the UV flash processes with semi-explicit index-1 stochastic DAEs. We augment the states with the disturbance variables and exploit the structure of the augmented DAE system to improve computational performance. We use the open-source thermodynamic software ThermoLib [32], [33] to evaluate thermodynamic functions. It is tailored for computation of first and second order derivatives which are needed for efficient implementation of the single-shooting method and in the adjoint method. We present a numerical example of output-constrained economical NMPC of a UV flash separation process that involves an unknown and unmeasured disturbance.

This paper is structured as follows. We describe the semi-explicit index-1 stochastic DAE system in Section II, and we discuss the numerical simulation of such systems in Section III. In Section IV, we discuss the estimation of the

*This work is funded by Innovation Fund Denmark in the OPTION project (63-2013-3). Tobias K. S. Ritschel and John Bagterp Jørgensen are with the Department of Applied Mathematics and Computer Science & the Center for Energy Resources Engineering (CERE), Technical University of Denmark, DK-2800 Kgs. Lyngby, Denmark {tobk, jbjjo}@dtu.dk

states and disturbance variables with the continuous-discrete EKF. We describe the open-loop optimal control problem and the single-shooting method in Section V. We describe the flash separation process in Section VI, and we present the numerical example in Section VII. Conclusions are given in Section VIII.

II. SEMI-EXPLICIT INDEX-1 STOCHASTIC DIFFERENTIAL-ALGEBRAIC EQUATIONS

We consider stochastic DAEs that are in the form

$$G(\mathbf{x}(t), \mathbf{y}(t), \mathbf{z}(t)) = 0, \quad (1a)$$

$$d\mathbf{x}(t) = F(\mathbf{y}(t), u(t), d(t))dt + \sigma(\mathbf{y}(t), u(t), d(t))d\boldsymbol{\omega}(t). \quad (1b)$$

$\mathbf{x}(t)$, $\mathbf{y}(t)$, and $\mathbf{z}(t)$ are vectors of state variables, algebraic variables, and adjoint algebraic variables, respectively. The algebraic equations (1a) represent phase equilibrium conditions, and the stochastic differential equations (1b) represent conservation equations. $u(t)$ are the manipulated inputs, and $d(t)$ are the disturbance variables. The initial states are normally distributed, i.e. $\mathbf{x}(t_0) \sim N(x_0, P_0)$. $\boldsymbol{\omega}(t)$ is a standard Wiener process, i.e. it has an incremental covariance of $\text{Id}t$. The algebraic equations (1a) are of index 1 for the processes that we consider. Therefore, they can be solved for $\mathbf{y}(t)$ and $\mathbf{z}(t)$ when $\mathbf{x}(t)$ is given. The measurements, $\mathbf{y}^m(t_k)$, of the outputs, $\mathbf{z}^m(t_k)$, are obtained at discrete times, t_k :

$$\mathbf{z}^m(t_k) = H(\mathbf{y}(t_k)), \quad (2a)$$

$$\mathbf{y}^m(t_k) = \mathbf{z}^m(t_k) + \mathbf{v}(t_k). \quad (2b)$$

$\mathbf{v}_k = \mathbf{v}(t_k)$ is the measurement noise. It is normally distributed, i.e. $\mathbf{v}_k \sim N(0, T_k)$.

III. NUMERICAL SIMULATION

We discretize the stochastic DAE (1) with a semi-implicit scheme, i.e. we discretize the drift and the diffusion terms in (1b) with Euler's implicit and explicit method, respectively. Between measurement k and $k+1$, we compute N_k time steps. In each time step, we solve $R_{k,n+1} = 0$ for $w_{k,n+1} = [x_{k,n+1}; y_{k,n+1}; z_{k,n+1}]$ where

$$\begin{aligned} R_{k,n+1} &= R_{k,n+1}(w_{k,n+1}; x_{k,n}, y_{k,n}, u_k, d_k) \\ &= R_{k,n+1}(x_{k,n+1}, y_{k,n+1}, z_{k,n+1}; x_{k,n}, y_{k,n}, u_k, d_k) \\ &= \begin{bmatrix} D_{k,n+1}(x_{k,n+1}, y_{k,n+1}; x_{k,n}, y_{k,n}, u_k, d_k) \\ G(x_{k,n+1}, y_{k,n+1}, z_{k,n+1}) \end{bmatrix}. \end{aligned} \quad (3)$$

The discretized stochastic differential equations are

$$\begin{aligned} D_{k,n+1} &= D_{k,n+1}(x_{k,n+1}, y_{k,n+1}; x_{k,n}, y_{k,n}, u_k, d_k) \\ &= x_{k,n+1} - F(y_{k,n+1}, u_k, d_k)\Delta t_{k,n} \\ &\quad - \sigma(y_{k,n}, u_k, d_k)\Delta\omega_{k,n} - x_{k,n}. \end{aligned} \quad (4)$$

We sample the increments, $\Delta\omega_{k,n}$, from $N(0, \text{Id}t_{k,n})$. We solve $R_{k,n+1} = 0$ with an inexact Newton method:

$$w_{k,n+1}^{l+1} = w_{k,n+1}^l + \Delta w_{k,n+1}^l. \quad (5)$$

In each Newton iteration, we solve

$$M\Delta w_{k,n+1}^l = -R_{k,n+1}(w_{k,n+1}^l), \quad (6)$$

for the Newton step. The iteration matrix, M , is

$$M \approx \frac{\partial R_{k,n+1}}{\partial w_{k,n+1}} = \begin{bmatrix} \mathbf{I} & -\frac{\partial F}{\partial y}\Delta t_{k,n} & 0 \\ \frac{\partial G}{\partial x} & \frac{\partial G}{\partial y} & \frac{\partial G}{\partial z} \end{bmatrix}. \quad (7)$$

IV. STATE AND DISTURBANCE ESTIMATION

In order to estimate the disturbances, we augment the stochastic DAE system (1) with a stochastic differential equation for the disturbance variables:

$$G(\mathbf{x}(t), \mathbf{y}(t), \mathbf{z}(t)) = 0, \quad (8a)$$

$$d\mathbf{x}(t) = F(\mathbf{y}(t), u(t), \mathbf{d}(t))dt + \sigma(\mathbf{y}(t), u(t), \mathbf{d}(t))d\boldsymbol{\omega}(t), \quad (8b)$$

$$d\mathbf{d}(t) = \sigma_d d\boldsymbol{\omega}_d(t). \quad (8c)$$

The measurement equations (2) remain unchanged. We use σ_d to tune the filter. We initialize the continuous-discrete EKF with $\hat{x}_{0|-1} = x_0$, $\hat{d}_{0|-1} = d_0$, $P_{0|-1}^{xx} = P_0^{xx}$, $P_{0|-1}^{dx} = 0$, and $P_{0|-1}^{dd} = P_0^{dd}$. d_0 is an initial estimate of the disturbance variables. We use P_0^{dd} for tuning of the filter.

A. Measurement-update

The one-step ahead prediction of the measurements, $\hat{\mathbf{y}}_{k|k-1}^m$, and its approximate covariance, $T_{k|k-1}$, are

$$\hat{\mathbf{y}}_{k|k-1}^m = H(\hat{\mathbf{y}}_{k|k-1}), \quad (9a)$$

$$T_{k|k-1} = C_k P_{k|k-1}^{xx} C_k' + T_k, \quad (9b)$$

where

$$C_k = \frac{\partial H}{\partial \mathbf{y}}(\hat{\mathbf{y}}_{k|k-1}) \frac{\partial \hat{\mathbf{y}}_{k|k-1}}{\partial \hat{\mathbf{x}}_{k|k-1}}. \quad (10)$$

The sensitivities, $\frac{\partial \hat{\mathbf{y}}_{k|k-1}}{\partial \hat{\mathbf{x}}_{k|k-1}}$ and $\frac{\partial \hat{\mathbf{z}}_{k|k-1}}{\partial \hat{\mathbf{x}}_{k|k-1}}$, satisfy

$$\begin{bmatrix} \frac{\partial G}{\partial \mathbf{y}} & \frac{\partial G}{\partial \mathbf{z}} \end{bmatrix} \begin{bmatrix} \frac{\partial \hat{\mathbf{y}}_{k|k-1}}{\partial \hat{\mathbf{x}}_{k|k-1}} \\ \frac{\partial \hat{\mathbf{z}}_{k|k-1}}{\partial \hat{\mathbf{x}}_{k|k-1}} \end{bmatrix} = -\frac{\partial G}{\partial \mathbf{x}}. \quad (11)$$

The innovation error is

$$e_k = \mathbf{y}_k^m - \hat{\mathbf{y}}_{k|k-1}^m, \quad (12)$$

and the Kalman filter gain matrices are

$$K_{fx,k} = P_{k|k-1}^{xx} C_k' T_{k|k-1}^{-1}, \quad (13a)$$

$$K_{fd,k} = P_{k|k-1}^{dx} C_k' T_{k|k-1}^{-1}. \quad (13b)$$

The filtered estimates of the states and disturbance variables and their covariance matrices are

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + K_{fx,k} e_k, \quad (14a)$$

$$\hat{\mathbf{d}}_{k|k} = \hat{\mathbf{d}}_{k|k-1} + K_{fd,k} e_k, \quad (14b)$$

$$P_{k|k}^{xx} = P_{k|k-1}^{xx} - K_{fx,k} T_{k|k-1} K_{fx,k}', \quad (14c)$$

$$P_{k|k}^{dx} = P_{k|k-1}^{dx} - K_{fd,k} T_{k|k-1} K_{fx,k}', \quad (14d)$$

$$P_{k|k}^{dd} = P_{k|k-1}^{dd} - K_{fd,k} T_{k|k-1} K_{fd,k}'. \quad (14e)$$

B. Time-update

We compute the one-step ahead predictions at time t_{k+1} by solving

$$\hat{x}_k(t_k) = \hat{x}_{k|k}, \quad (15a)$$

$$G(\hat{x}_k(t), \hat{y}_k(t), \hat{z}_k(t)) = 0, \quad t \in [t_k; t_{k+1}], \quad (15b)$$

$$d\hat{x}_k(t) = F(\hat{y}_k(t), u(t), \hat{d}_k(t))dt, \quad t \in [t_k; t_{k+1}], \quad (15c)$$

where $u(t) = u_{k|k}$ and $\hat{d}_k(t) = \hat{d}_{k|k}$. In order to compute the one-step ahead predictions of the covariance matrices, we compute the sensitivities, $\Phi_{xx}(t, s) = \frac{\partial \hat{x}_k(t)}{\partial \hat{x}_k(s)}$, $\Phi_{yx}(t, s) = \frac{\partial \hat{y}_k(t)}{\partial \hat{x}_k(s)}$, $\Phi_{zx}(t, s) = \frac{\partial \hat{z}_k(t)}{\partial \hat{x}_k(s)}$, $\Phi_{xd}(t, s) = \frac{\partial \hat{x}_k(t)}{\partial \hat{d}_k(s)}$, $\Phi_{yd}(t, s) = \frac{\partial \hat{y}_k(t)}{\partial \hat{d}_k(s)}$, and $\Phi_{zd}(t, s) = \frac{\partial \hat{z}_k(t)}{\partial \hat{d}_k(s)}$, by solving

$$\frac{\partial G}{\partial x} \Phi_{xx}(t, s) + \frac{\partial G}{\partial y} \Phi_{yx}(t, s) + \frac{\partial G}{\partial z} \Phi_{zx}(t, s) = 0, \quad (16a)$$

$$\frac{\partial G}{\partial x} \Phi_{xd}(t, s) + \frac{\partial G}{\partial y} \Phi_{yd}(t, s) + \frac{\partial G}{\partial z} \Phi_{zd}(t, s) = 0, \quad (16b)$$

$$\frac{d\Phi_{xx}(t, s)}{dt} = \frac{\partial F}{\partial y} \Phi_{yx}(t, s), \quad (16c)$$

$$\frac{d\Phi_{xd}(t, s)}{dt} = \frac{\partial F}{\partial y} \Phi_{yd}(t, s) + \frac{\partial F}{\partial d} \Phi_{dd}(t, s), \quad (16d)$$

where $\Phi_{xx}(s, s) = \mathbf{I}$, $\Phi_{xd}(s, s) = 0$, and $\Phi_{dd}(t, s) = \mathbf{I}$. The covariance matrices are given by [20]

$$\begin{aligned} P_k^{xx}(t) &= \Phi_{xx} P_{k|k}^{xx} \Phi'_{xx} + \Phi_{xd} P_{k|k}^{dx} \Phi'_{xx} \\ &+ \Phi_{xx} (P_{k|k}^{dx})' \Phi'_{xd} + \Phi_{xd} P_{k|k}^{dd} \Phi'_{xd} \\ &+ \int_{t_k}^t \Omega_{xx}(t, s) \Omega_{xx}(t, s)' + \Omega_{xd}(t, s) \Omega_{xd}(t, s)' ds, \end{aligned} \quad (17a)$$

$$\begin{aligned} P_k^{dx}(t) &= \Phi_{dd} P_{k|k}^{dx} \Phi'_{xx} + \Phi_{dd} P_{k|k}^{dd} \Phi'_{xd} \\ &+ \int_{t_k}^t \Omega_{dd}(t, s) \Omega_{xd}(t, s)' ds, \end{aligned} \quad (17b)$$

$$P_k^{dd}(t) = \Phi_{dd} P_{k|k}^{dd} \Phi'_{dd} + \int_{t_k}^t \Omega_{dd}(t, s) \Omega_{dd}(t, s)' ds, \quad (17c)$$

where the sensitivity matrices are evaluated at t and t_k , e.g. $\Phi_{xx} = \Phi_{xx}(t, t_k)$, and

$$\Omega_{xx}(t, s) = \Phi_{xx}(t, s) \sigma(\hat{y}_k(s), u(s), \hat{d}_k(s)), \quad (18a)$$

$$\Omega_{xd}(t, s) = \Phi_{xd}(t, s) \sigma_d, \quad (18b)$$

$$\Omega_{dd}(t, s) = \Phi_{dd}(t, s) \sigma_d. \quad (18c)$$

C. Numerical solution of the time-update equations

We solve the time-update equations (15) with Euler's implicit method, i.e. we solve the equations

$$\begin{bmatrix} D_{k,n+1}(\hat{x}_{k,n+1}, \hat{y}_{k,n+1}; \hat{x}_{k,n}, u_{k|k}, \hat{d}_{k|k}) \\ G(\hat{x}_{k,n+1}, \hat{y}_{k,n+1}, \hat{z}_{k,n+1}) \end{bmatrix} = 0, \quad (19)$$

where the discretized differential equations are

$$D_{k,n+1} = \hat{x}_{k,n+1} - F(\hat{y}_{k,n+1}, u_{k|k}, \hat{d}_{k|k}) \Delta t_{k,n} - \hat{x}_{k,n}. \quad (20)$$

The sensitivities satisfy

$$\begin{bmatrix} \mathbf{I} & -\frac{\partial F}{\partial y} \Delta t_{k,n} & 0 \\ \frac{\partial G}{\partial x} & \frac{\partial G}{\partial y} & \frac{\partial G}{\partial z} \end{bmatrix} \begin{bmatrix} \Phi_{xx} & \Phi_{xd} \\ \Phi_{yx} & \Phi_{yd} \\ \Phi_{zx} & \Phi_{zd} \end{bmatrix} = \begin{bmatrix} \mathbf{I} & \frac{\partial F}{\partial d} \Delta t_{k,n} \\ 0 & 0 \end{bmatrix}, \quad (21)$$

where the sensitivity matrices are evaluated at $t_{k,n+1}$ and $t_{k,n}$, e.g. $\Phi_{xx} = \Phi_{xx}(t_{k,n+1}, t_{k,n})$. We discretize the integrals in (17) with a right rectangle quadrature rule:

$$\begin{aligned} P_{k,n+1}^{xx} &= \Phi_{xx} P_{k,n}^{xx} \Phi'_{xx} + \Phi_{xd} P_{k,n}^{dx} \Phi'_{xx} + \Phi_{xx} (P_{k,n}^{dx})' \Phi'_{xd} \\ &+ \Phi_{xd} P_{k,n}^{dd} \Phi'_{xd} + (\Omega_{xx} \Omega'_{xx} + \Omega_{xd} \Omega'_{xd}) \Delta t_{k,n}, \end{aligned} \quad (22a)$$

$$P_{k,n+1}^{dx} = P_{k,n}^{dx} \Phi'_{xx} + P_{k,n}^{dd} \Phi'_{xd} + \Omega_{dd} \Omega'_{xd} \Delta t_{k,n}, \quad (22b)$$

$$P_{k,n+1}^{dd} = P_{k,n}^{dd} + \Omega_{dd} \Omega'_{dd} \Delta t_{k,n}. \quad (22c)$$

The sensitivities and the matrices Ω_{xx} , Ω_{xd} , and Ω_{dd} , in (22) are evaluated at $t_{k,n+1}$ and $t_{k,n}$. The sensitivity matrices in (22) are therefore the ones that we solve (21) for. We have exploited that $\Phi_{dd}(t, s) = \mathbf{I}$ in (21) and (22).

V. DYNAMIC OPTIMIZATION

We assume that there is one control interval in between measurements. At sample time t_k , we solve the OCP

$$\min_{[x(t); y(t); z(t)]_{t_k}^{t_k+N_h}, \{u_{j|k}\}_{j=k}^{k+N_h-1}} \phi \left([y(t); u(t); d(t)]_{t_k}^{t_k+N_h} \right), \quad (23a)$$

subject to

$$x(t_k) = \hat{x}_{k|k}, \quad (23b)$$

$$G(x(t), y(t), z(t)) = 0, \quad t \in [t_k, t_k+N_h], \quad (23c)$$

$$\dot{x}(t) = F(y(t), u(t), d(t)), \quad t \in [t_k, t_k+N_h], \quad (23d)$$

$$u(t) = u_{j|k}, \quad t \in [t_j, t_{j+1}[], \quad j = k, \dots, k+N_h-1, \quad (23e)$$

$$d(t) = \hat{d}_{j|k}, \quad t \in [t_j, t_{j+1}[], \quad j = k, \dots, k+N_h-1, \quad (23f)$$

$$\{u_{j|k}\}_{j=k}^{k+N_h-1} \in \mathcal{U}, \quad (23g)$$

where the objective function, ϕ , is in Lagrange form:

$$\phi \left([y(t); u(t); d(t)]_{t_k}^{t_k+N_h} \right) = \int_{t_k}^{t_k+N_h} \Phi(y(t), u(t), d(t)) dt. \quad (24)$$

We consider a control and prediction horizon of N_h control intervals. $[x(t); y(t); z(t)]_{t_k}^{t_k+N_h}$ is a vector of dependent optimization variables, and $\{u_{j|k}\}_{j=k}^{k+N_h-1}$ are independent optimization variables. The DAE system (23c)-(23d) is deterministic, i.e. there is no process noise. (23e)-(23f) are zero-order-hold parametrizations of the manipulated inputs and the disturbance variables, and (23g) are constraints on the manipulated inputs. We assume that $\hat{d}_{j|k} = \hat{d}_{k|k}$ for $j = k+1, \dots, k+N_h-1$. The OCP (23) needs the current estimates of the states, $\hat{x}_{k|k}$, and disturbance variables, $\hat{d}_{k|k}$. They are computed by the continuous-discrete EKF.

A. Single-shooting

We use the single-shooting algorithm described in [18] to solve the OCP (23). We define the objective function, ψ , as

$$\begin{aligned} \psi &= \psi(\{u_{j|k}\}_{j=k}^{k+N_h-1}; \hat{x}_{k|k}, \{\hat{d}_{j|k}\}_{j=k}^{k+N_h-1}) \\ &= \left\{ \phi : (23b)-(23f) \right\}. \end{aligned} \quad (25)$$

That is, ψ is the objective function ϕ in (24) evaluated using the solution of the DAE system (23c)-(23d) with the initial condition (23b) and the zero-order-hold parametrizations of the manipulated inputs and the disturbance variables (23e)-(23f). In the single-shooting approach, we solve the finite dimensional nonlinear program

$$\min_{\{u_{j|k}\}_{j=k}^{k+N_h-1}} \psi = \psi(\{u_{j|k}\}_{j=k}^{k+N_h-1}; \hat{x}_{k|k}, \{\hat{d}_{j|k}\}_{j=k}^{k+N_h-1}), \quad (26a)$$

$$\text{s.t.} \quad \{u_{j|k}\}_{j=k}^{k+N_h-1} \in \mathcal{U}. \quad (26b)$$

We solve the DAE system (23c)-(23d) with Euler's implicit method. For each time step, we solve the residual equations, $R_{j,n+1} = 0$, for $w_{j,n+1}$ where

$$\begin{aligned} R_{j,n+1} &= R_{j,n+1}(w_{j,n+1}; x_{j,n}, u_{j|k}, \hat{d}_{j|k}) \\ &= R_{j,n+1}(x_{j,n+1}, y_{j,n+1}, z_{j,n+1}; x_{j,n}, u_{j|k}, \hat{d}_{j|k}) \\ &= \begin{bmatrix} D_{j,n+1}(x_{j,n+1}, y_{j,n+1}; x_{j,n}, u_{j|k}, \hat{d}_{j|k}) \\ G(x_{j,n+1}, y_{j,n+1}, z_{j,n+1}) \end{bmatrix}, \end{aligned} \quad (27)$$

and the discretized differential equations are

$$\begin{aligned} D_{j,n+1} &= D_{j,n+1}(x_{j,n+1}, y_{j,n+1}; x_{j,n}, u_{j|k}, \hat{d}_{j|k}) \\ &= x_{j,n+1} - F(y_{j,n+1}, u_{j|k}, \hat{d}_{j|k})\Delta t_{j,n} - x_{j,n}. \end{aligned} \quad (28)$$

Furthermore, $w_{j+1,0} = w_{j,N_j}$. We substitute the discretized DAEs (27) into (25) and approximate the integral in (24) with a right rectangle quadrature rule:

$$\psi = \psi(\{u_{j|k}\}_{j=k}^{k+N_h-1}; \hat{x}_{k|k}, \{\hat{d}_{j|k}\}_{j=k}^{k+N_h-1}) \quad (29a)$$

$$= \left\{ \phi = \sum_{j=k}^{k+N_h-1} \sum_{n=0}^{N_j-1} \Phi_{j,n}(y_{j,n+1}, u_{j|k}, \hat{d}_{j|k}) : \right. \quad (29b)$$

$$x_{k,0} = \hat{x}_{k|k}, \quad (29c)$$

$$\left. R_{j,n+1}(w_{j,n+1}; x_{j,n}, u_{j|k}, \hat{d}_{j|k}) = 0 \right\}. \quad (29d)$$

$n = 0, \dots, N_j - 1$ and $j = k, \dots, k + N_h - 1$ in (29d) and

$$\Phi_{j,n}(y_{j,n+1}, u_{j|k}, \hat{d}_{j|k}) = \Phi(y_{j,n+1}, u_{j|k}, \hat{d}_{j|k})\Delta t_{j,n}. \quad (30)$$

Proposition 1: Consider the function, ψ , defined in (29). The gradients, $\nabla_{u_{j|k}} \psi$, can be computed by

$$\nabla_{u_{j|k}} \psi = \sum_{n=0}^{N_j-1} \left(\nabla_{u_{j|k}} \Phi_{j,n} + \left(\frac{\partial R_{j,n+1}}{\partial u_{j|k}} \right)' \lambda_{j,n+1} \right), \quad (31)$$

where the adjoints satisfy

$$\left(\frac{\partial R_{j,N_j}}{\partial w_{j,N_j}} \right)' \lambda_{j,N_j} = -\nabla_{w_{j,N_j}} \Phi_{j,N_j-1}, \quad (32)$$

for $j = k + N_h - 1$,

$$\left(\frac{\partial R_{j,n}}{\partial w_{j,n}} \right)' \lambda_{j,n} = -\left(\frac{\partial R_{j,n+1}}{\partial w_{j,n}} \right)' \lambda_{j,n+1} - \nabla_{w_{j,n}} \Phi_{j,n-1}, \quad (33)$$

for $n = N_j - 1, \dots, 1$ and $j = k + N_h - 1, \dots, k$, and

$$\begin{aligned} \left(\frac{\partial R_{j,N_j}}{\partial w_{j,N_j}} \right)' \lambda_{j,N_j} &= -\left(\frac{\partial R_{j+1,1}}{\partial w_{j+1,0}} \right)' \lambda_{j+1,1} \\ &\quad - \nabla_{w_{j,N_j}} \Phi_{j,N_j-1}, \end{aligned} \quad (34)$$

for $j = k + N_h - 2, \dots, k$.

VI. DYNAMIC UV FLASH SEPARATION

We consider the flash separation of a mixture of N_C components into a vapor phase (v) and a liquid phase (l). The two phases are in thermodynamic equilibrium. The vapor and liquid phases exit the separator from two separate streams. The separator is supplied with a vapor-liquid mixture through a feed stream. The separator is cooled, $Q(t) \leq 0$. The internal energy, U , and the total amount of moles of each component, n , are described by the conservation equations

$$\dot{U}(t) = H_F^v(t) + H_F^l(t) - H_V(t) - H_L(t) + Q(t), \quad (35a)$$

$$\dot{n}_i(t) = f_{F,i}^v(t) + f_{F,i}^l(t) - v_i(t) - l_i(t), \quad i = 1, \dots, N_C. \quad (35b)$$

H_F^v and H_F^l are the vapor-liquid enthalpies of the feed stream, and H_V and H_L are the enthalpies of the vapor and liquid streams. Q is the heat flux from the external cooling. f_F^v and f_F^l are vectors of the molar flow rates of the feed stream, and v and l are vectors of the molar flow rates of the vapor and liquid streams. The volume, V , of the separator is fixed. The equilibrium temperature, T , pressure, P , and vapor-liquid compositions, n^v and n^l , are the solution to the UV flash optimization problem:

$$\max_{T, P, n^v, n^l} S = S^v(T, P, n^v) + S^l(T, P, n^l), \quad (36a)$$

$$\text{s.t.} \quad U^v(T, P, n^v) + U^l(T, P, n^l) = U, \quad (36b)$$

$$V^v(T, P, n^v) + V^l(T, P, n^l) = V, \quad (36c)$$

$$n_i^v + n_i^l = n_i, \quad i = 1, \dots, N_C. \quad (36d)$$

The UV flash optimization problem (36) is a mathematical statement of the second law of thermodynamics, i.e. the entropy of a closed system in equilibrium is maximal. That condition can be adapted to isothermal-isobaric (constant temperature - constant pressure) systems in which case it states that Gibbs energy, G , is minimal. That is the condition that determines the molar flow rates of the feed stream, f_F^v and f_F^l , based on the feed temperature, T_F , pressure, P_F , and total molar flow rates, f_F :

$$\min_{f_F^v, f_F^l} G = G^v(T_F, P_F, f_F^v) + G^l(T_F, P_F, f_F^l), \quad (37a)$$

$$f_{F,i}^v + f_{F,i}^l = f_{F,i}, \quad i = 1, \dots, N_C. \quad (37b)$$

(37) is called the PT flash optimization problem. The first-order optimality conditions of (36) constitute the algebraic equations (1a). We solve (35) and (36) simultaneously. However, we nest the solution of (37) into the evaluation of the right-hand side of (35). The properties in the right-hand side of the conservation equations (35) are uncertain, e.g. because of limited accuracy in the thermodynamic parameters. The differential equations (35) are therefore stochastic, and the process noise can be non-additive depending on how the uncertain parameters enter into the model equations [34].

VII. NUMERICAL EXAMPLE

We consider the flash separation of a hydrocarbon mixture in a 1 m^3 separator. The feed mixture contains 60% C_1 , 10% C_2 , 5% C_3 , 23% n- C_7 , and 2% H_2S . There are 6 states, 12 algebraic variables, and 7 adjoint algebraic variables. We control the separation process over a time period of 8 h. The prediction and control horizon in the NMPC algorithm is 6 h. The manipulated inputs are the heat flux, Q , and the total flow rates of the vapor and liquid streams, F_V and F_L . The total feed flow rate is $F_F = 12 \text{ kmol/h}$. The objective of the NMPC algorithm is to minimize the amount of energy spent on cooling, i.e. to maximize $\int_{0^h}^{8^h} Q(t)dt$ (which is negative). There are upper bounds on the H_2S vapor mole fraction, $y_{H_2S} \leq 2\%$, and the pressure, $P \leq 7.5 \text{ MPa}$. We incorporate those bounds into the objective function with logarithmic barrier functions. We incorporate bounds on the vapor fraction, $\beta \in [0\%, 100\%]$, in a similar manner. The feed temperature, T_F , is an unknown disturbance variable that we estimate. The initial estimate of T_F is equal to its true value. However, T_F increases with 5 K after 2 h. This increase is not known by the NMPC algorithm. We use $\sigma_d = 0.5 \text{ K}$ and $P_0^{dd} = 0.25^2 \text{ K}^2$ in the EKF. We measure the temperature and pressure every 5 min. This is also the length of the control intervals. The standard deviations of the temperature and pressure measurement noises are 2.5 K and 0.05 MPa. The diffusion coefficient is constant, i.e. $\sigma(y(t), u(t), d(t)) = \sigma = \text{diag}([\sigma_U; \sigma_{C_1}; \sigma_{C_2}; \sigma_{C_3}; \sigma_{n-C_7}; \sigma_{H_2S}])$ where $\sigma_U = 0.2 \text{ MJ}$, $\sigma_{C_1} = \sigma_{C_2} = \sigma_{n-C_7} = 2 \text{ mol}$, and $\sigma_{C_3} = \sigma_{H_2S} = 0.2 \text{ mol}$. x_0 is a steady-state of the system (when process noise is disregarded), and $P_0^{xx} = \sigma\sigma'$.

Fig. 1 shows a closed-loop simulation of NMPC of the flash separation process. The upper bounds on y_{H_2S} and P are satisfied throughout the entire simulation. Fig. 2 shows the estimated disturbance variable, T_F , together with the manipulated inputs, Q , F_V , and F_L . The EKF is able to track the increase in T_F , and the NMPC algorithm is able to reduce the cooling and thereby lower the energy consumption. The computation time of the NMPC algorithm is between 9 s and 74 s in each of the control intervals, and the average computation time is 29 s.

VIII. CONCLUSIONS

We present an NMPC algorithm for disturbance rejection in UV flash processes. It estimates the states and the unknown disturbance variables with the continuous-discrete

EKF and solves the involved OCPs with a single-shooting method. The gradients are computed with an adjoint method. It is natural for models of UV flash processes to be in a semi-explicit index-1 stochastic DAE form. We describe a model of a UV flash separation process in such a form. We present a numerical example of economical NMPC of the UV flash separation process. The example involves constraints on the H_2S vapor mole fraction and the pressure, and the estimation is based on temperature and pressure measurements. The NMPC algorithm is able to estimate an unknown increase in the unmeasured feed temperature while satisfying the constraints. The computation time is 74 s or less in all of the 5 min control intervals which indicates that real-time NMPC of UV flash processes is computationally feasible.

REFERENCES

- [1] T. Binder, L. Blank, H. G. Bock, R. Bulirsch, W. Dahmen, M. Diehl, T. Kronseder, W. Marquardt, J. P. Schlöder, and O. von Stryk, "Introduction to model based optimization of chemical processes on moving horizons," in *Online Optimization of Large Scale Systems*. Springer-Verlag Berlin Heidelberg, 2001, pp. 295–339.
- [2] A. S. Kumar and Z. Ahmad, "Model predictive control (MPC) and its current issues in chemical engineering," *Chemical Engineering Communications*, vol. 199, no. 4, pp. 472–511, 2012.
- [3] A. Krishnan, B. V. Patil, P. S. V. Nataraj, J. Maciejowski, and K. V. Ling, "Model predictive control of a CSTR: A comparative study among linear and nonlinear model approaches," in *Proceedings of the 2017 Indian Control Conference*, Guwahati, India, Jan. 2017, pp. 182–187.
- [4] S. Lucia, J. A. E. Andersson, H. Brandt, A. Bouaswaig, M. Diehl, and S. Engell, "Efficient robust economic nonlinear model predictive control of an industrial batch reactor," in *Proceedings of the 19th IFAC World Congress*, Cape Town, South Africa, Aug. 2014, pp. 11 093–11 098.
- [5] L. N. Petersen and J. B. Jørgensen, "Real-time economic optimization for a fermentation process using model predictive control," in *Proceedings of the 2014 European Control Conference*, Strasbourg, France, June 2014, pp. 1831–1836.
- [6] J. D. Gil, L. Roca, A. Ruiz-Aguirre, G. Zaragoza, J. L. Guzmán, and M. Berenguel, "Using a nonlinear model predictive control strategy for the efficient operation of a solar-powered membrane distillation system," in *Proceedings of the 25th Mediterranean Conference on Control and Automation*, 2017.
- [7] Z. Chen, M. A. Henson, P. Belanger, and L. Megan, "Nonlinear model predictive control of high purity distillation columns for cryogenic air separation," *IEEE Transactions on Control Systems Technology*, vol. 18, no. 4, pp. 811–821, 2010.
- [8] V. L. S. Silva, A. A. Emerick, P. Couto, and J. L. D. Alves, "History matching and production optimization under uncertainties - application of closed-loop reservoir management," *Journal of Petroleum Science and Engineering*, vol. 157, pp. 860–874, 2017.
- [9] C. Völcker, J. B. Jørgensen, P. G. Thomsen, and E. H. Stenby, "NMPC for oil reservoir production optimization," in *Proceedings of the 21st European Symposium on Computer Aided Process Engineering*, 2011, pp. 1849–1853.
- [10] M. Castier, "Dynamic simulation of fluids in vessels via entropy maximization," *Journal of Industrial and Engineering Chemistry*, vol. 16, no. 1, pp. 122–129, Jan. 2010.
- [11] A. R. J. Arendsen and G. F. Versteeg, "Dynamic thermodynamics with internal energy, volume, and amount of moles as states: Application to liquefied gas tank," *Industrial & Engineering Chemistry Research*, vol. 48, no. 6, pp. 3167–3176, Jan. 2009.
- [12] E. R. A. Lima, M. Castier, and E. C. Biscaia Jr., "Differential-algebraic approach to dynamic simulations of flash drums with rigorous evaluation of physical properties," *Oil & Gas Science and Technology*, vol. 63, no. 5, pp. 677–686, 2008.
- [13] P. Flatby, S. Skogestad, and P. Lundström, "Rigorous dynamic simulation of distillation columns based on UV-flash," in *IFAC Symposium on Advanced Control of Chemical Processes (ADCHEM '94)*, 1994, pp. 261–266.

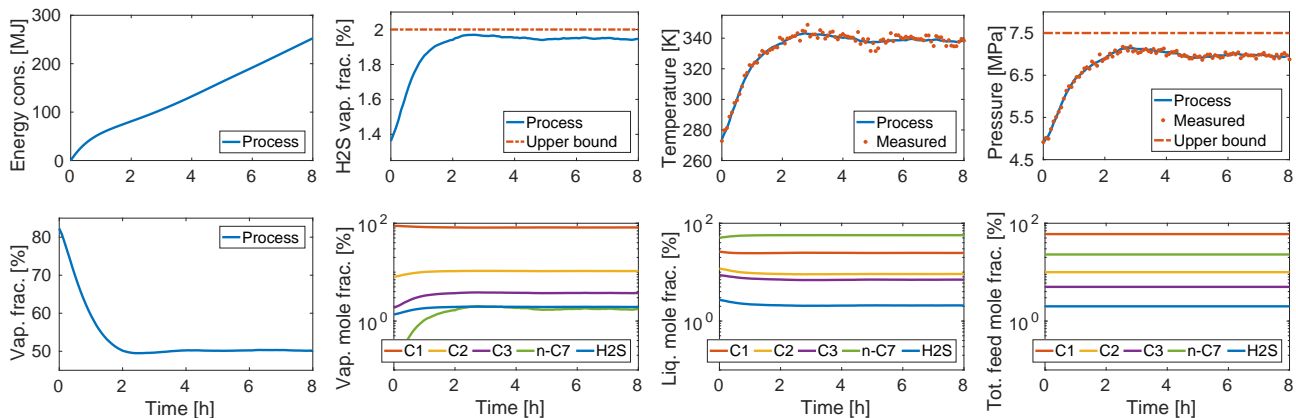


Fig. 1. Closed-loop simulation of the flash separation process. Top row: Energy consumption and H₂S vapor mole fraction as well as actual and measured temperature and pressure. Bottom row: Vapor fraction and vapor-liquid mole fractions of the mixture in the separator and total mole fractions of the feed.

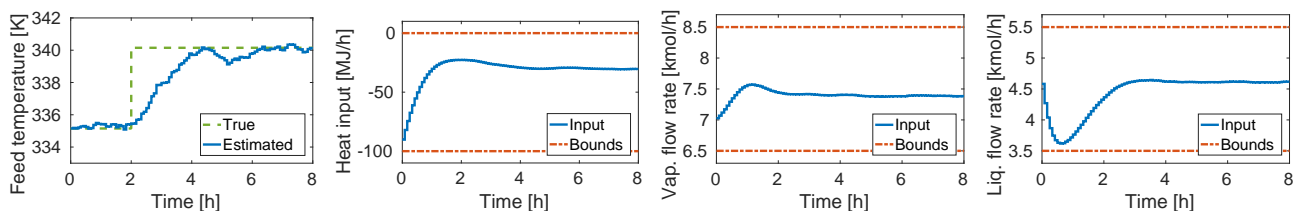


Fig. 2. The estimated disturbance variable, T_F , and the manipulated inputs, Q , F_V , and F_L .

- [14] M. Hammer and A. Morin, "A method for simulating two-phase pipe flow with real equations of state," *Computers & Fluids*, vol. 100, pp. 45–58, 2014.
- [15] L. Qiu, Y. Wang, and R. D. Reitz, "Multiphase dynamic flash simulations using entropy maximization and application to compressible flow with phase change," *AIChE Journal*, vol. 60, no. 8, pp. 3013–3024, 2014.
- [16] M. L. Michelsen, "State function based flash specifications," *Fluid Phase Equilibria*, vol. 158–160, pp. 617–626, 1999.
- [17] T. K. S. Ritschel and J. B. Jørgensen, "The extended Kalman filter for state estimation of dynamic UV flash processes," in *Proceedings of the 3rd IFAC Workshop on Automatic Control in Offshore Oil and Gas Production*, 2018, accepted.
- [18] T. K. S. Ritschel, A. Capolei, J. Gaspar, and J. B. Jørgensen, "An algorithm for gradient-based dynamic optimization of UV flash processes," *Computers and Chemical Engineering*, 2017, in Press. DOI: <https://doi.org/10.1016/j.compchemeng.2017.10.007>.
- [19] P. Mobed, S. Munusamy, D. Bhattacharyya, and R. Rengaswamy, "State and parameter estimation in distributed constrained systems. 1. extended Kalman filtering of a special class of differential-algebraic equation systems," *Industrial & Engineering Chemistry Research*, vol. 56, no. 1, pp. 206–215, 2016.
- [20] J. B. Jørgensen, M. R. Kristensen, P. G. Thomsen, and H. Madsen, "New extended Kalman filter algorithms for stochastic differential algebraic equations," in *Assessment and Future Directions of Nonlinear Model Predictive Control*, ser. Lecture Notes in Control and Information Sciences. Springer-Verlag Berlin Heidelberg, 2007, vol. 358, pp. 359–366.
- [21] D. Simon, *Optimal State Estimation: Kalman, H Infinity, and Nonlinear Approaches*. John Wiley & Sons, 2006.
- [22] G. Evensen, *Data Assimilation. The Ensemble Kalman Filter*, 2nd ed. Springer-Verlag Berlin Heidelberg, 2009.
- [23] —, "The ensemble Kalman filter for combined state and parameter estimation," *IEEE Control Systems*, vol. 29, no. 3, pp. 83–104, 2009.
- [24] A. Alessandri, M. Baglietto, G. Battistelli, and V. Zavala, "Advances in moving horizon estimation for nonlinear systems," in *Proceedings of the 49th IEEE Conference on Decision and Control*, Atlanta, Georgia, USA, Dec. 2010.
- [25] H. A. Talebi, F. Abdollahi, R. V. Patel, and K. Khorasani, *Neural Network-Based State Estimation of Nonlinear Systems. Application to Fault Detection and Isolation*, ser. Lecture Notes in Control and Information Sciences. Springer-Verlag New York, 2010, vol. 395.
- [26] J. B. Jørgensen, "Adjoint sensitivity results for predictive control, state- and parameter-estimation with nonlinear models," in *European Control Conference (ECC), 2007*. IEEE, 2007, pp. 3649–3656.
- [27] M. R. Kristensen, J. B. Jørgensen, P. G. Thomsen, and S. B. Jørgensen, "An ESDIRK method with sensitivity analysis capabilities," *Computers & Chemical Engineering*, vol. 28, no. 12, pp. 2695–2707, 2004.
- [28] B. Vatankhah and M. Farokhi, "Nonlinear model-predictive control with disturbance rejection property using adaptive neural networks," *Journal of the Franklin Institute*, vol. 354, no. 13, pp. 5201–5220, 2017.
- [29] M. Morari and U. Maeder, "Nonlinear offset-free model predictive control," *Automatica*, vol. 48, no. 9, pp. 2059–2067, 2012.
- [30] R. Huang, L. T. Biegler, and S. C. Patwardhan, "Fast offset-free nonlinear model predictive control based on moving horizon estimation," *Industrial & Engineering Chemistry Research*, vol. 49, no. 17, pp. 7882–7890, 2010.
- [31] S. Subramanian, S. Lucia, and S. Engell, "Handling structural plant-model mismatch via multi-stage nonlinear model predictive control," in *Proceedings of the 2015 European Control Conference*, Linz, Austria, July 2015, pp. 1602–1607.
- [32] T. K. S. Ritschel, J. Gaspar, and J. B. Jørgensen, "A thermodynamic library for simulation and optimization of dynamic processes," in *Proceedings of the 20th World Congress of the International Federation of Automatic Control*, 2017.
- [33] T. K. S. Ritschel, J. Gaspar, A. Capolei, and J. B. Jørgensen, "An open-source thermodynamic software library," Department of Applied Mathematics and Computer Science, Technical University of Denmark, Tech. Rep. DTU Compute Technical Report-2016-12, 2016.
- [34] S. Kolás, B. A. Foss, and T. S. Schei, "Noise modeling concepts in nonlinear state estimation," *Journal of Process Control*, vol. 19, no. 7, pp. 1111–1125, 2009.

Technical Report I

An Open-source Thermodynamic Software Library

Authors:

Tobias K. S. Ritschel, Jozsef Gaspar, Andrea Capolei, and John Bagterp Jørgensen

DTU Compute Technical Report-2016-12, 2016,
Department of Applied Mathematics and Computer Science,
Technical University of Denmark.

An open-source thermodynamic software library*

DTU Compute Technical Report-2016-12

Tobias K. S. Ritschel, Jozsef Gaspar, Andrea Capolei, John Bagterp Jørgensen

*Department of Applied Mathematics and Computer Science & Center for Energy Resources Engineering (CERE),
Technical University of Denmark, DK-2800 Kgs. Lyngby, Denmark*

Abstract

This is a technical report which accompanies the article "An open-source thermodynamic software library" which describes an efficient Matlab and C implementation for evaluation of thermodynamic properties. In this technical report we present the model equations, that are also presented in the paper, together with a full set of first and second order derivatives with respect to temperature and pressure, and in cases where applicable, also with respect to mole numbers. The library is based on parameters and correlations from the DIPPR database and the Peng-Robinson and the Soave-Redlich-Kwong equations of state.

Keywords: Thermodynamic functions, Software, Phase equilibrium

1. Introduction

The purpose of this technical report is to document the equations describing vapor-liquid enthalpy, entropy and volume of real and ideal mixtures and pure components, together with their first and second order derivatives with respect to temperature and pressure, and for mixture properties, also with respect to mole numbers. For completeness, this technical report also describes logarithmic fugacity coefficients together with first and second order derivatives. The fugacity coefficients are defined by means of residual properties that are also described in this report. However, their second order derivatives require third order derivatives of the residual properties, which are not described in this report².

Section 2 presents DIPPR correlations together with derived expressions that are necessary in the ideal gas and liquid models. Section 3 presents ideal gas properties and Section 4 presents ideal liquid properties. Section 5 presents real mixture properties based on the Peng-Robinson (PR) and the Soave-Redlich-Kwong (SRK) equations of state.

2. DIPPR correlations for pure components

This section describes the DIPPR correlations (Thomson, 1996) for ideal gas heat capacity, vapor pressure and liquid volume together with necessary integrals and temperature derivatives.

*This project is funded partly by Innovation Fund Denmark in the CITIES project (1305-00027B) and in the OPTION project (63-2013-3), and partly by the interreg project Smart Cities Accelerator (10606 SCA).

²The main motivation for including second order derivatives of fugacity coefficients is that they are necessary for formulating second order algorithms for dynamic optimization of flash processes.

Furthermore, we present all derivatives of these correlations and derived properties, that are necessary for the first and second order derivatives in the ideal gas and liquid models.

2.1. Ideal gas heat capacity

The ideal gas heat capacity of the i 'th component, $c_{P,i}^{ig} = c_{P,i}^{ig}(T)$, is

$$c_{P,i}^{ig} = A_i + B_i \left(\frac{C_i}{T} \right)^2 + D_i \left(\frac{E_i}{T} \right)^2 \quad (1)$$

The parameters (A_i, B_i, C_i, D_i, E_i) are specific to each substance and also to the ideal gas heat capacity correlation. They are provided by the DIPPR database. The unit for the molar ideal gas heat capacity is J/(kmoles K) and the temperature must be in K. The integral of the ideal gas heat capacity is used in computation of ideal gas enthalpy and is expressed using the auxiliary function $\Gamma_i = \Gamma_i(T)$

$$\int_{T_0}^{T_1} c_{P,i}^{ig} dT = \Gamma_i(T_1) - \Gamma_i(T_0) \quad (2a)$$

$$\Gamma_i(T) = A_i T + B_i C_i \coth\left(\frac{C_i}{T}\right) - D_i E_i \tanh\left(\frac{E_i}{T}\right) \quad (2b)$$

The integral of the ideal gas heat capacity divided by temperature is used for the ideal gas entropy and is expressed using the auxiliary function $\Pi_i = \Pi_i(T)$

$$\int_{T_0}^{T_1} \frac{c_{P,i}^{ig}}{T} dT = \Pi_i(T_1) - \Pi_i(T_0) \quad (3a)$$

$$\begin{aligned} \Pi_i(T) = & A_i \ln(T) + B_i \left(\frac{C_i}{T} \coth\left(\frac{C_i}{T}\right) - \ln \sinh\left(\frac{C_i}{T}\right) \right) \\ & - D_i \left(\frac{E_i}{T} \tanh\left(\frac{E_i}{T}\right) - \ln \cosh\left(\frac{E_i}{T}\right) \right) \end{aligned} \quad (3b)$$

The only necessary derivative of the ideal gas heat capacity is the first order temperature derivative

$$\frac{\partial c_{P,i}^{ig}}{\partial T} = \frac{2}{T^2} \left[(A_i - c_{P,i}^{ig}) T + \frac{B_i C_i}{\tanh\left(\frac{C_i}{T}\right)} \left(\frac{C_i}{\sinh\left(\frac{C_i}{T}\right)} \right)^2 + D_i E_i \tanh\left(\frac{E_i}{T}\right) \left(\frac{E_i}{\cosh\left(\frac{E_i}{T}\right)} \right)^2 \right] \quad (4)$$

2.2. Vapor pressure

The vapor pressure or saturation pressure of the i 'th component, $P_i^{sat} = P_i^{sat}(T)$, is

$$P_i^{sat} = \exp(\ln P_i^{sat}) \quad (5a)$$

$$\ln P_i^{sat} = A_i + \frac{B_i}{T} + C_i \ln(T) + D_i T^{E_i} \quad (5b)$$

The substance specific correlation parameters (A_i, B_i, C_i, D_i, E_i) are provided by the DIPPR database and are specific to the vapor pressure correlation. Because both the vapor pressure and its first order temperature derivative appear in the ideal liquid model, we present first, second and third

order derivatives of the above correlation. The derivatives of the vapor pressure are expressed through the derivatives of the logarithmic vapor pressure

$$\frac{\partial P_i^{sat}}{\partial T} = P_i^{sat} \frac{\partial \ln P_i^{sat}}{\partial T} \quad (6a)$$

$$\frac{\partial^2 P_i^{sat}}{\partial T^2} = P_i^{sat} \left(\left(\frac{\partial \ln P_i^{sat}}{\partial T} \right)^2 + \frac{\partial^2 \ln P_i^{sat}}{\partial T^2} \right) \quad (6b)$$

$$\frac{\partial^3 P_i^{sat}}{\partial T^3} = P_i^{sat} \left(\frac{\partial^3 \ln P_i^{sat}}{\partial T^3} + 3 \frac{\partial^2 \ln P_i^{sat}}{\partial T^2} \frac{\partial \ln P_i^{sat}}{\partial T} + \left(\frac{\partial \ln P_i^{sat}}{\partial T} \right)^3 \right) \quad (6c)$$

The derivatives of the logarithmic vapor pressure correlation are

$$\frac{\partial \ln P_i^{sat}}{\partial T} = \frac{1}{T} \left(C_i - \frac{B_i}{T} + D_i E_i T^{E_i} \right) \quad (7a)$$

$$\frac{\partial^2 \ln P_i^{sat}}{\partial T^2} = \frac{1}{T^2} \left(2 \frac{B_i}{T} - C_i + D_i E_i (E_i - 1) T^{E_i} \right) \quad (7b)$$

$$\frac{\partial^3 \ln P_i^{sat}}{\partial T^3} = \frac{1}{T^3} \left(-6 \frac{B_i}{T} + 2C_i + D_i E_i (E_i - 1)(E_i - 2) T^{E_i} \right) \quad (7c)$$

2.3. Liquid volume

The liquid volume of the i 'th component, $v_i^l = v_i^l(T)$, is

$$v_i^l = \frac{B_i^{1 + \left(1 - \frac{T}{C_i}\right)^{D_i}}}{A_i} \quad (8)$$

The substance specific correlation parameters (A_i, B_i, C_i, D_i) are provided by the DIPPR database and are specific to this correlation. Because both the liquid volume and its first order derivative appear in the ideal liquid model, we present first, second and third order temperature derivatives of the liquid volume correlation. The derivatives are

$$\frac{\partial v_i^l}{\partial T} = -\ln B_i \frac{D_i}{C_i} \left(1 - \frac{T}{C_i}\right)^{D_i-1} v_i^l \quad (9a)$$

$$\frac{\partial^2 v_i^l}{\partial T^2} = -\left(\ln B_i \frac{D_i}{C_i} \left(1 - \frac{T}{C_i}\right)^{D_i-1} + \frac{D_i - 1}{C_i \left(1 - \frac{T}{C_i}\right)} \right) \frac{\partial v_i^l}{\partial T} \quad (9b)$$

$$\frac{\partial^3 v_i^l}{\partial T^3} = -\left(\ln B_i \frac{D_i}{C_i} + \frac{D_i - 1}{C_i \left(1 - \frac{T}{C_i}\right)} \right) \frac{\partial^2 v_i^l}{\partial T^2} - \frac{D_i - 1}{C_i \left(1 - \frac{T}{C_i}\right)} \left(\ln B_i \frac{D_i}{C_i} + \frac{1}{C_i \left(1 - \frac{T}{C_i}\right)} \right) \frac{\partial v_i^l}{\partial T} \quad (9c)$$

3. Thermodynamic functions for ideal gases

This section presents enthalpy, entropy and volume of ideal gas mixtures and pure components. These equations are based on the reference enthalpy and entropy of formation provided by the DIPPR database, together with the correlation for the ideal gas heat capacity and the ideal gas law.

3.1. Pure component properties

We describe the computation of enthalpy, entropy and volume of a pure component i

3.1.1. Enthalpy

The molar ideal gas enthalpy, $h_i^{ig} = h_i^{ig}(T)$, is a function of temperature only

$$h_i^{ig} = \underbrace{h_i^{ig}(T_0, P_0) - \Gamma_i(T_0)}_{\hat{h}_{f,i}^{ig}} + \Gamma_i(T) \quad (10)$$

It is more efficient to store $\hat{h}_{f,i}^{ig} = h_i^{ig}(T_0, P_0) - \Gamma_i(T_0)$ rather than storing $h_i^{ig}(T_0, P_0)$ and recomputing $\Gamma_i(T_0)$ at every evaluation of h_i^{ig} . The derivatives of the molar enthalpy are

$$\frac{\partial h_i^{ig}}{\partial T} = c_{P,i}^{ig} \quad (11a)$$

$$\frac{\partial^2 h_i^{ig}}{\partial T^2} = \frac{\partial c_{P,i}^{ig}}{\partial T} \quad (11b)$$

where the ideal gas heat capacity and its temperature derivative are given by (1) and (4).

3.1.2. Entropy

The molar ideal gas entropy, $s_i^{ig} = s_i^{ig}(T, P)$, is a function of temperature and pressure

$$s_i^{ig} = \Pi_i(T) - R \ln(P) + \underbrace{s_i^{ig}(T_0, P_0) - \Pi_i(T_0) + R \ln(P_0)}_{\hat{s}_{f,i}^{ig}} \quad (12)$$

It is more efficient to store $\hat{s}_{f,i}^{ig} = s_i^{ig}(T_0, P_0) - \Pi_i(T_0) + R \ln(P_0)$ than recomputing $\Pi_i(T_0)$ at every evaluation of s_i^{ig} . The first order temperature and pressure derivatives of the molar ideal gas entropy are

$$\frac{\partial s_i^{ig}}{\partial T} = \frac{c_{P,i}^{ig}}{T} \quad (13a)$$

$$\frac{\partial s_i^{ig}}{\partial P} = -\frac{R}{P} \quad (13b)$$

The second order derivatives of the molar ideal gas entropy are

$$\frac{\partial^2 s_i^{ig}}{\partial T^2} = \frac{1}{T} \frac{dc_{P,i}^{ig}}{dT} - \frac{1}{T^2} c_{P,i}^{ig} \quad (14a)$$

$$\frac{\partial^2 s_i^{ig}}{\partial P^2} = \frac{R}{P^2} \quad (14b)$$

$$\frac{\partial^2 s_i^{ig}}{\partial T \partial P} = 0 \quad (14c)$$

where again, the ideal gas heat capacity and its temperature derivative are given by (1) and (4).

3.1.3. Volume

The molar ideal gas volume, $v_i^{ig} = v_i^{ig}(T, P)$, is

$$v_i^{ig} = \frac{RT}{P} \quad (15)$$

The first order derivatives of the ideal gas molar volume are

$$\frac{\partial v_i^{ig}}{\partial T} = \frac{R}{P} \quad (16a)$$

$$\frac{\partial v_i^{ig}}{\partial P} = -\frac{RT}{P^2} \quad (16b)$$

The second order derivatives are

$$\frac{\partial^2 v_i^{ig}}{\partial T^2} = 0 \quad (17a)$$

$$\frac{\partial^2 v_i^{ig}}{\partial P^2} = 2\frac{RT}{P^3} \quad (17b)$$

$$\frac{\partial^2 v_i^{ig}}{\partial T \partial P} = -\frac{R}{P^2} \quad (17c)$$

3.2. Mixture properties

We describe the computation of volume, enthalpy and entropy of an ideal gas mixture of N_C components using the molar properties of each component and the composition of the gas mixture. The mixture contains $n = \{n_i\}_{i=1}^{N_C}$ moles of each component.

3.2.1. Enthalpy

The ideal gas mixture enthalpy, $H^{ig} = H^{ig}(T, n)$, is

$$H^{ig} = \sum_{i=1}^{N_C} n_i h_i^{ig} \quad (18)$$

The first order derivatives are

$$\frac{\partial H^{ig}}{\partial T} = \sum_{i=1}^{N_C} n_i c_{P,i}^{ig} \quad (19a)$$

$$\frac{\partial H^{ig}}{\partial n_k} = h_k^{ig} \quad (19b)$$

The second order derivatives are

$$\frac{\partial^2 H^{ig}}{\partial T^2} = \sum_{i=1}^{N_C} n_i \frac{\partial^2 h_i^{ig}}{\partial T^2} \quad (20a)$$

$$\frac{\partial^2 H^{ig}}{\partial T \partial n_k} = c_{P,k}^{ig} \quad (20b)$$

$$\frac{\partial^2 H^{ig}}{\partial n_i \partial n_k} = 0 \quad (20c)$$

3.2.2. Entropy

The ideal gas mixture enthalpy, $S^{ig} = S^{ig}(T, P, n)$, is

$$S^{ig} = \sum_{i=1}^{N_C} n_i s_i^{ig} - R \sum_{i=1}^{N_C} n_i \ln(y_i) \quad (21)$$

where the total amount of moles, N , and the vapor mole fraction, y_i , are

$$y_i = \frac{n_i}{N} \quad (22a)$$

$$N = \sum_{i=1}^{N_C} n_i \quad (22b)$$

The derivatives of the mixture entropy are

$$\frac{\partial S^{ig}}{\partial T} = \frac{1}{T} \sum_{i=1}^{N_C} n_i c_{P,i}^{ig} \quad (23a)$$

$$\frac{\partial S^{ig}}{\partial P} = -\frac{NR}{P} \quad (23b)$$

$$\frac{\partial S^{ig}}{\partial n_k} = s_k^{ig} - R \ln(y_k) \quad (23c)$$

The second order derivatives are

$$\frac{\partial^2 S^{ig}}{\partial T^2} = \frac{1}{T} \sum_{i=1}^{N_C} n_i \left(\frac{\partial c_{P,i}^{ig}}{\partial T} - \frac{1}{T} c_{P,i}^{ig} \right) \quad (24a)$$

$$\frac{\partial^2 S^{ig}}{\partial P^2} = \frac{NR}{P^2} \quad (24b)$$

$$\frac{\partial^2 S^{ig}}{\partial T \partial P} = 0 \quad (24c)$$

$$\frac{\partial^2 S^{ig}}{\partial T \partial n_k} = \frac{\partial s_k^{ig}}{\partial T} = \frac{c_{P,k}^{ig}}{T} \quad (24d)$$

$$\frac{\partial^2 S^{ig}}{\partial P \partial n_k} = \frac{\partial s_k^{ig}}{\partial P} = -\frac{R}{P} \quad (24e)$$

$$\frac{\partial^2 S^{ig}}{\partial n_l \partial n_k} = -R \left(\frac{\delta_{kl}}{n_l} - \frac{1}{N} \right) \quad (24f)$$

3.2.3. Volume

The ideal gas mixture volume, $V^{ig} = V^{ig}(T, P, n)$, is

$$V^{ig} = \frac{NRT}{P} \quad (25)$$

The first order derivatives of the ideal gas mixture volume are

$$\frac{\partial V^{ig}}{\partial T} = \frac{NR}{P} \quad (26a)$$

$$\frac{\partial V^{ig}}{\partial P} = \frac{NRT}{P^2} \quad (26b)$$

$$\frac{\partial V^{ig}}{\partial n_k} = \frac{RT}{P} \quad (26c)$$

The second order derivatives are

$$\frac{\partial^2 V^{ig}}{\partial T^2} = 0 \quad (27a)$$

$$\frac{\partial^2 V^{ig}}{\partial P^2} = 2 \frac{NRT}{P^3} \quad (27b)$$

$$\frac{\partial^2 V^{ig}}{\partial T \partial P} = -\frac{NR}{P^2} \quad (27c)$$

$$\frac{\partial^2 V^{ig}}{\partial T \partial n_k} = \frac{R}{P} \quad (27d)$$

$$\frac{\partial^2 V^{ig}}{\partial P \partial n_k} = -\frac{RT}{P^2} \quad (27e)$$

$$\frac{\partial^2 V^{ig}}{\partial n_i \partial n_k} = 0 \quad (27f)$$

4. Thermodynamic functions for ideal liquids

This section presents enthalpy, entropy and volume of ideal liquid mixtures and pure components. These are based on ideal gas properties, vaporization properties and pressure correcting terms. The latter two are further based on the DIPPR correlations for vapor pressure and liquid volume, together with their first order temperature derivatives.

4.1. Pure component properties

The properties of pure component liquids are based on ideal gas properties at vaporization temperature and pressure, T and $P_i^{sat} = P_i^{sat}(T)$, as well as vaporization properties. The vaporization enthalpy, $\Delta h_i^{vap} = \Delta h_i^{vap}(T)$, entropy, $\Delta s_i^{vap} = \Delta s_i^{vap}(T)$, and volume, $\Delta v_i^{vap} = \Delta v_i^{vap}(T)$, are

$$\Delta v_i^{vap} = \frac{RT}{P_i^{sat}} - v_i^l \quad (28a)$$

$$\Delta s_i^{vap} = \frac{\partial P_i^{sat}}{\partial T} \Delta v_i^{vap} \quad (28b)$$

$$\Delta h_i^{vap} = T \Delta s_i^{vap} \quad (28c)$$

where $P_i^{sat} = P_i^{sat}(T)$ is the vapor pressure (5) and $v_i^l = v_i^l(T)$ is the liquid volume (8). The derivatives of the vaporization volume are

$$\frac{\partial \Delta v_i^{vap}}{\partial T} = \frac{R}{P_i^{sat}} \left(1 - \frac{T}{P_i^{sat}} \frac{\partial P_i^{sat}}{\partial T} \right) - \frac{\partial v_i^l}{\partial T} \quad (29a)$$

$$\frac{\partial^2 \Delta v_i^{vap}}{\partial T^2} = -\frac{R}{(P_i^{sat})^2} \left(2 \frac{\partial P_i^{sat}}{\partial T} \left(1 - \frac{T}{P_i^{sat}} \frac{\partial P_i^{sat}}{\partial T} \right) + T \frac{\partial^2 P_i^{sat}}{\partial T^2} \right) - \frac{\partial^2 v_i^l}{\partial T^2} \quad (29b)$$

The derivatives of the vaporization enthalpy are

$$\frac{\partial \Delta h_i^{vap}}{\partial T} = \Delta s_i^{vap} + T \frac{\partial \Delta s_i^{vap}}{\partial T} \quad (30a)$$

$$\frac{\partial^2 \Delta h_i^{vap}}{\partial T^2} = 2 \frac{\partial \Delta s_i^{vap}}{\partial T} + T \frac{\partial^2 \Delta s_i^{vap}}{\partial T^2} \quad (30b)$$

The derivatives of the vaporization entropy are

$$\frac{\partial \Delta s_i^{vap}}{\partial T} = \frac{\partial^2 P_i^{sat}}{\partial T^2} \Delta v_i^{vap} + \frac{\partial P_i^{sat}}{\partial T} \frac{\partial \Delta v_i^{vap}}{\partial T} \quad (31a)$$

$$\frac{\partial^2 \Delta s_i^{vap}}{\partial T^2} = \frac{\partial^3 P_i^{sat}}{\partial T^3} \Delta v_i^{vap} + 2 \frac{\partial^2 P_i^{sat}}{\partial T^2} \frac{\partial \Delta v_i^{vap}}{\partial T} + \frac{\partial P_i^{sat}}{\partial T} \frac{\partial^2 \Delta v_i^{vap}}{\partial T^2} \quad (31b)$$

The molar liquid saturation enthalpy, $h_i^{sat} = h_i^{sat}(T)$, and entropy, $s_i^{sat} = s_i^{sat}(T)$, are

$$h_i^{sat} = h_i^v(T) - \Delta h_i^{vap}(T) \quad (32a)$$

$$s_i^{sat} = s_i^v(T, P_i^{sat}) - \Delta s_i^{vap}(T) \quad (32b)$$

The derivatives of the molar liquid saturation enthalpy are

$$\frac{\partial h_i^{sat}}{\partial T} = c_{P,i}^{ig} - \frac{\partial \Delta h_i^{vap}}{\partial T} \quad (33)$$

$$\frac{\partial^2 h_i^{sat}}{\partial T^2} = \frac{\partial c_{P,i}^{ig}}{\partial T} - \frac{\partial^2 \Delta h_i^{vap}}{\partial T^2} \quad (34)$$

The derivatives of the molar liquid saturation entropy

$$\frac{\partial s_i^{sat}}{\partial T} = \frac{\partial s_i^v}{\partial T}(T, P_i^{sat}) + \frac{\partial s_i^v}{\partial P}(T, P_i^{sat}) \frac{\partial P_i^{sat}}{\partial T} - \frac{\partial \Delta s_i^{vap}}{\partial T} \quad (35)$$

$$\frac{\partial^2 s_i^{sat}}{\partial T^2} = \frac{\partial^2 s_i^v}{\partial T^2}(T, P_i^{sat}) + 2 \frac{\partial^2 s_i^v}{\partial T \partial P}(T, P_i^{sat}) \frac{\partial P_i^{sat}}{\partial T} + \frac{\partial^2 s_i^v}{\partial P^2}(T, P_i^{sat}) \left(\frac{\partial P_i^{sat}}{\partial T} \right)^2 - \frac{\partial^2 \Delta s_i^{vap}}{\partial T^2} \quad (36)$$

The molar liquid enthalpy, $h_i^{id} = h_i^{id}(T, P)$, entropy, $s_i^{id} = s_i^{id}(T, P)$, at arbitrary pressure are

$$h_i^{id} = h_i^{sat} + \left(v_i^l - T \frac{\partial v_i^l}{\partial T} \right) (P - P_i^{sat}) \quad (37a)$$

$$s_i^{id} = s_i^{sat} - \frac{\partial v_i^l}{\partial T} (P - P_i^{sat}) \quad (37b)$$

The molar ideal liquid volume is given by the DIPPR correlation (8). The first order derivatives of liquid enthalpy are

$$\frac{\partial h_i^{id}}{\partial T} = \frac{\partial h_i^{sat}}{\partial T} - T \frac{\partial^2 v_i^l}{\partial T^2} (P - P_i^{sat}) - \left(v_i^l - T \frac{\partial v_i^l}{\partial T} \right) \frac{\partial P_i^{sat}}{\partial T} \quad (38a)$$

$$\frac{\partial h_i^{id}}{\partial P} = v_i^l - T \frac{\partial v_i^l}{\partial T} \quad (38b)$$

The second order derivatives are

$$\frac{\partial^2 h_i^{id}}{\partial T^2} = \frac{\partial^2 h_i^{sat}}{\partial T^2} - \left(\frac{\partial^2 v_i^l}{\partial T^2} + T \frac{\partial^3 v_i^l}{\partial T^3} \right) (P - P_i^{sat}) + 2T \frac{\partial^2 v_i^l}{\partial T^2} \frac{\partial P_i^{sat}}{\partial T} - \left(v_i^l - T \frac{\partial v_i^l}{\partial T} \right) \frac{\partial^2 P_i^{sat}}{\partial T^2} \quad (39a)$$

$$\frac{\partial^2 h_i^{id}}{\partial P^2} = 0 \quad (39b)$$

$$\frac{\partial^2 h_i^{id}}{\partial T \partial P} = -T \frac{\partial^2 v_i^l}{\partial T^2} \quad (39c)$$

The first order derivatives of liquid entropy are

$$\frac{\partial s_i^{id}}{\partial T} = \frac{\partial s_i^{sat}}{\partial T} - \left(\frac{\partial^2 v_i^l}{\partial T^2} (P - P_i^{sat}) - \frac{\partial v_i^l}{\partial T} \frac{\partial P_i^{sat}}{\partial T} \right) \quad (40a)$$

$$\frac{\partial s_i^{id}}{\partial P} = -\frac{\partial v_i^l}{\partial T} \quad (40b)$$

The second order derivatives are

$$\frac{\partial^2 s_i^{id}}{\partial T^2} = \frac{\partial^2 s_i^{sat}}{\partial T^2} - \left(\frac{\partial^3 v_i^l}{\partial T^3} (P - P_i^{sat}) - 2 \frac{\partial^2 v_i^l}{\partial T^2} \frac{\partial P_i^{sat}}{\partial T} - \frac{\partial v_i^l}{\partial T} \frac{\partial^2 P_i^{sat}}{\partial T^2} \right) \quad (41a)$$

$$\frac{\partial^2 s_i^{id}}{\partial P^2} = 0 \quad (41b)$$

$$\frac{\partial^2 s_i^{id}}{\partial T \partial P} = -\frac{\partial^2 v_i^l}{\partial T^2} \quad (41c)$$

4.2. Mixture properties

We present volume, enthalpy and entropy of an ideal gas mixture of N_C components using the molar properties of each component and the composition of the gas mixture. The mixture contains $n = \{n_i\}_{i=1}^{N_C}$ moles of each component.

4.2.1. Enthalpy

The ideal liquid mixture enthalpy, $H^{id} = H^{id}(T, P, n)$, is

$$H^{id} = \sum_{i=1}^{N_C} n_i h_i^{id} \quad (42)$$

The first order derivatives are

$$\frac{\partial H^{id}}{\partial T} = \sum_{i=1}^{N_c} n_i \frac{\partial h_i^{id}}{\partial T} \quad (43a)$$

$$\frac{\partial H^{id}}{\partial P} = \sum_{i=1}^{N_c} n_i \left(v_i^l - T \frac{\partial v_i^l}{\partial T} \right) \quad (43b)$$

$$\frac{\partial H^{id}}{\partial n_k} = h_k^{id} \quad (43c)$$

The second order derivatives are

$$\frac{\partial^2 H^{id}}{\partial T^2} = \sum_{i=1}^{N_c} n_i \frac{\partial^2 h_i^{id}}{\partial T^2} \quad (44a)$$

$$\frac{\partial^2 H^{id}}{\partial P^2} = 0 \quad (44b)$$

$$\frac{\partial^2 H^{id}}{\partial T \partial P} = -T \sum_{i=1}^{N_c} n_i \frac{\partial^2 v_i^l}{\partial T^2} \quad (44c)$$

$$\frac{\partial^2 H^{id}}{\partial T \partial n_k} = \frac{\partial h_k^{id}}{\partial T} \quad (44d)$$

$$\frac{\partial^2 H^{id}}{\partial P \partial n_k} = v_k^l - T \frac{\partial v_k^l}{\partial T} \quad (44e)$$

$$\frac{\partial^2 H^{id}}{\partial n_l \partial n_k} = 0 \quad (44f)$$

4.2.2. Entropy

The entropy of an ideal liquid mixture is given by

$$S^{id}(T, P, n) = \sum_{i=1}^{N_c} n_i s_i^{id} - R \sum_{i=1}^{N_c} n_i \ln x_i \quad (45)$$

where the liquid mole fraction, x_i , is

$$x_i = \frac{n_i}{N} \quad (46)$$

and the total amount of moles, N , is given by (22b). The first order derivatives are

$$\frac{\partial S^{id}}{\partial T} = \sum_{i=1}^{N_c} n_i \frac{\partial s_i^{id}}{\partial T} \quad (47a)$$

$$\frac{\partial S^{id}}{\partial P} = - \sum_{i=1}^{N_c} n_i \frac{\partial v_i^l}{\partial T} \quad (47b)$$

$$\frac{\partial S^{id}}{\partial n_k} = s_k^{id} - R \ln(x_k) \quad (47c)$$

The second order derivatives are

$$\frac{\partial^2 S^{id}}{\partial T^2} = \sum_{i=1}^{N_c} n_i \frac{\partial^2 s_i^{id}}{\partial T^2} \quad (48a)$$

$$\frac{\partial^2 S^{id}}{\partial P^2} = 0 \quad (48b)$$

$$\frac{\partial^2 S^{id}}{\partial T \partial P} = - \sum_{i=1}^{N_c} n_i \frac{\partial^2 v_i^l}{\partial T^2} \quad (48c)$$

$$\frac{\partial^2 S^{id}}{\partial T \partial n_k} = \frac{\partial s_k^{id}}{\partial T} \quad (48d)$$

$$\frac{\partial^2 S^{id}}{\partial P \partial n_k} = - \frac{\partial v_k^l}{\partial T} \quad (48e)$$

$$\frac{\partial^2 S^{id}}{\partial n_l \partial n_k} = -R \left(\frac{\delta_{kl}}{n_l} - \frac{1}{N} \right) \quad (48f)$$

4.2.3. Volume

The ideal liquid mixture volume, $V^{id} = V^{id}(T, n)$, is

$$V^{id} = \sum_{i=1}^{N_c} n_i v_i^l(T) \quad (49)$$

$$(50)$$

The first order derivatives are

$$\frac{\partial V^{id}}{\partial T} = \sum_{i=1}^{N_c} n_i \frac{\partial v_i^l}{\partial T} \quad (51a)$$

$$\frac{\partial V^{id}}{\partial n_k} = v_k^l \quad (51b)$$

The second order derivatives are

$$\frac{\partial^2 V^{id}}{\partial T^2} = \sum_{i=1}^{N_c} n_i \frac{\partial^2 v_i^l}{\partial T^2} \quad (52a)$$

$$\frac{\partial^2 V^{id}}{\partial T \partial n_k} = \frac{\partial v_k^l}{\partial T} \quad (52b)$$

$$\frac{\partial^2 V^{id}}{\partial n_l \partial n_k} = 0 \quad (52c)$$

5. Thermodynamic functions for real mixtures

This section presents the enthalpy, entropy and volume of a real vapor or liquid mixture. These are based on ideal gas properties and residual properties. The latter are obtained from either of the cubic equations of state, Soave-Redlich-Kwong (SRK) or Peng-Robinson (PR)

5.1. Mixture properties

We consider a vapor or liquid phase containing N_C components with mole numbers $n = \{n_i\}_{i=1}^{N_C}$. The total amount of moles in the phase, N , and the mole fraction of the i 'th component, z_i , are given by

$$N = \sum_{i=1}^{N_C} n_i \quad (53a)$$

$$z_i = \frac{n_i}{N} \quad (53b)$$

The molar enthalpy, $h = h(T, P, n)$, and entropy, $s = s(T, P, n)$, are

$$h = h^{ig} + h^R \quad (54)$$

$$s = s^{ig} + s^R \quad (55)$$

where $h^{ig} = h^{ig}(T, P, n)$ and $s^{ig} = s^{ig}(T, P, n)$ are molar ideal gas enthalpy and entropy, and $h^R(T, P, n)$ and $s^R = s^R(T, P, n)$ are molar residual enthalpy and entropy. The molar volume, $v = v(T, P, n)$, is the solution of either the Peng-Robinson or the Soave-Redlich-Kwong equations of state, both of which are in the cubic form

$$P = \frac{RT}{v - b_m} - \frac{a_m}{(v + \epsilon b_m)(v + \sigma b_m)} \quad (56)$$

where the scalars ϵ and σ are specific to each equation of state but independent of the given substances. In practice, the equation of state (56) is solved for the compressibility factor $Z = Z(T, P, n)$ in which case the molar mixture volume is

$$v = \frac{RTZ}{P} \quad (57)$$

Appendix A presents a direct and an iterative approach for solving the cubic equations of state for the compressibility factor. The mixture parameters $a_m = a_m(T, n)$ and $b_m = b_m(n)$ are obtained with van der Waals mixing rules

$$a_m = \sum_{i=1}^{N_C} \sum_{j=1}^{N_C} z_i z_j a_{ij} \quad (58a)$$

$$b_m = \sum_{i=1}^{N_C} z_i b_i \quad (58b)$$

where the mixing parameter $a_{ij} = a_{ij}(T)$ is

$$a_{ij} = (1 - k_{ij}) \sqrt{\hat{a}_{ij}} \quad (59)$$

The parameter $\hat{a}_{ij} = \hat{a}_{ij}(T) = a_i(T)a_j(T)$ is introduced for convenience and the substance specific parameters $a_i = a_i(T)$ and b_i are determined from the critical temperature, $T_{c,i}$, and critical pressure, $P_{c,i}$, of the i 'th component

$$a_i = \alpha(T_{r,i}, \omega_i) \Psi \frac{R^2 T_{c,i}^2}{P_{c,i}} \quad (60a)$$

$$b_i = \Omega \frac{RT_{c,i}}{P_{c,i}} \quad (60b)$$

Table 1: Parameters in the Soave-Redlich-Kwong and the Peng-Robinson equations of state.

Eq.	ϵ	σ	Ω	Ψ
SRK	1	0	0.08664	0.42748
PR	$1 + \sqrt{2}$	$1 - \sqrt{2}$	0.07779	0.45724

where $\alpha(T_{r,i}, \omega_i)$ is a function of the reduced temperature, $T_{r,i} = T/T_{c,i}$, and the acentric factor, ω_i , given by

$$\alpha(T_{r,i}, \omega_i) = \left(1 + m(\omega_i)(1 - T_{r,i}^{1/2})\right)^2 \quad (61)$$

The scalars Ψ and Ω are related to the equation of state parameters, ϵ and σ , and their values are shown in Table 1. The function $m(\omega_i)$ is a second order polynomial in the acentric factor, ω_i , for the Peng-Robinson and Soave-Redlich-Kwong equations of state and is given by

$$m_{\text{SRK}}(\omega_i) = 0.480 + 1.574\omega_i - 0.176\omega_i^2 \quad (62a)$$

$$m_{\text{PR}}(\omega_i) = 0.37464 + 1.54226\omega_i - 0.26992\omega_i^2 \quad (62b)$$

The cubic equation of state (56) is solved for the compressibility factor $Z = Z(T, P, n)$ and is therefore rewritten using the third order polynomial $q = q(Z)$

$$q = Z^3 + \sum_{m=0}^2 d_m Z^m = 0 \quad (63)$$

where the polynomial coefficients $\{d_m = d_m(A, B)\}_{m=0}^2$ are

$$d_2 = B(\epsilon + \sigma - 1) - 1 \quad (64a)$$

$$d_1 = A - B(\epsilon + \sigma) + B^2(\epsilon\sigma - \epsilon - \sigma) \quad (64b)$$

$$d_0 = -(AB + (B^2 + B^3)\epsilon\sigma) \quad (64c)$$

The dimensionless quantities $A = A(T, P, n)$ and $B = B(T, P, n)$ are introduced for convenience and are given by

$$A = \frac{Pa_m}{R^2T^2} \quad (65a)$$

$$B = \frac{Pb_m}{RT} \quad (65b)$$

5.2. Residual enthalpy and entropy

The molar residual enthalpy and entropy are given in terms of the four auxiliary functions $f = f(Z, B)$, $g_h = g_h(T, n)$, $g_s = g_s(T, n)$ and $g_z = g_z(Z, B)$

$$h^R = h^R(T, P, n) = RT(Z - 1) + \frac{1}{\epsilon - \sigma} g_h f \quad (66a)$$

$$s^R = s^R(T, P, n) = Rg_z + \frac{1}{\epsilon - \sigma} g_s f \quad (66b)$$

The auxiliary functions, $f = f(Z, B)$, $g_h = g_h(T, n)$, $g_s = g_s(T, n)$ and $g_z = g_z(Z, B)$, are

$$f = \ln\left(\frac{Z + \epsilon B}{Z + \sigma B}\right) \quad (67a)$$

$$g_h = T g_s - \frac{a_m}{b_m} \quad (67b)$$

$$g_s = \frac{1}{b_m} \frac{\partial a_m}{\partial T} \quad (67c)$$

$$g_z = \ln(Z - B) \quad (67d)$$

The function $f = f(Z, B)$ depends on the equation of state parameters, ϵ and σ , whereas $g_h = g_h(T, n)$, $g_s = g_s(T, n)$ and $g_z = g_z(Z, B)$ do not. The first order derivatives of the residual enthalpy, $h^R = h^R(T, P, n)$, are

$$\frac{\partial h^R}{\partial T} = R(Z - 1) + RT \frac{\partial Z}{\partial T} + \frac{1}{\epsilon - \sigma} \left(\frac{\partial g_h}{\partial T} f + g_h \frac{\partial f}{\partial T} \right) \quad (68a)$$

$$\frac{\partial h^R}{\partial P} = RT \frac{\partial Z}{\partial P} + \frac{1}{\epsilon - \sigma} g_h \frac{\partial f}{\partial P} \quad (68b)$$

$$\frac{\partial h^R}{\partial n_k} = RT \frac{\partial Z}{\partial n_k} + \frac{1}{\epsilon - \sigma} \left(\frac{\partial g_s}{\partial n_k} f + g_h \frac{\partial f}{\partial n_k} \right) \quad (68c)$$

The second order derivatives are

$$\frac{\partial^2 h^R}{\partial T^2} = 2R \frac{\partial Z}{\partial T} + RT \frac{\partial^2 Z}{\partial T^2} + \frac{1}{\epsilon - \sigma} \left(\frac{\partial^2 g_h}{\partial T^2} f + 2 \frac{\partial g_h}{\partial T} \frac{\partial f}{\partial T} + g_h \frac{\partial^2 f}{\partial T^2} \right) \quad (69a)$$

$$\frac{\partial^2 h^R}{\partial P^2} = RT \frac{\partial^2 Z}{\partial P^2} + \frac{1}{\epsilon - \sigma} g_h \frac{\partial^2 f}{\partial P^2} \quad (69b)$$

$$\frac{\partial^2 h^R}{\partial T \partial P} = R \frac{\partial Z}{\partial P} + RT \frac{\partial^2 Z}{\partial T \partial P} + \frac{1}{\epsilon - \sigma} \left(\frac{\partial g_h}{\partial T} \frac{\partial f}{\partial P} + g_h \frac{\partial^2 f}{\partial T \partial P} \right) \quad (69c)$$

$$\frac{\partial^2 h^R}{\partial T \partial n_k} = R \frac{\partial Z}{\partial n_k} + RT \frac{\partial^2 Z}{\partial T \partial n_k} + \frac{1}{\epsilon - \sigma} \left(\frac{\partial^2 g_h}{\partial T \partial n_k} f + \frac{\partial g_h}{\partial T} \frac{\partial f}{\partial n_k} + \frac{\partial g_h}{\partial n_k} \frac{\partial f}{\partial T} + g_h \frac{\partial^2 f}{\partial T \partial n_k} \right) \quad (69d)$$

$$\frac{\partial^2 h^R}{\partial P \partial n_k} = RT \frac{\partial^2 Z}{\partial P \partial n_k} + \frac{1}{\epsilon - \sigma} \left(\frac{\partial g_h}{\partial n_k} \frac{\partial f}{\partial P} + g_h \frac{\partial^2 f}{\partial P \partial n_k} \right) \quad (69e)$$

$$\frac{\partial^2 h^R}{\partial n_i \partial n_k} = RT \frac{\partial^2 Z}{\partial n_i \partial n_k} + \frac{1}{\epsilon - \sigma} \left(\frac{\partial^2 g_h}{\partial n_i \partial n_k} f + \frac{\partial g_h}{\partial n_k} \frac{\partial f}{\partial n_i} + \frac{\partial g_h}{\partial n_i} \frac{\partial f}{\partial n_k} + g_h \frac{\partial^2 f}{\partial n_i \partial n_k} \right) \quad (69f)$$

The first order derivatives of the residual entropy, $s^R = s^R(T, P, n)$, are

$$\frac{\partial s^R}{\partial T} = R \frac{\partial g_z}{\partial T} + \frac{1}{\epsilon - \sigma} \left(\frac{\partial g_s}{\partial T} f + g_s \frac{\partial f}{\partial T} \right) \quad (70a)$$

$$\frac{\partial s^R}{\partial P} = R \frac{\partial g_z}{\partial P} + \frac{1}{\epsilon - \sigma} g_s \frac{\partial f}{\partial P} \quad (70b)$$

$$\frac{\partial s^R}{\partial n_k} = R \frac{\partial g_z}{\partial n_k} + \frac{1}{\epsilon - \sigma} \left(\frac{\partial g_s}{\partial n_k} f + g_s \frac{\partial f}{\partial n_k} \right) \quad (70c)$$

The second order derivatives are

$$\frac{\partial^2 s^R}{\partial T^2} = R \frac{\partial^2 g_z}{\partial T^2} + \frac{1}{\epsilon - \sigma} \left(\frac{\partial^2 g_s}{\partial T^2} f + 2 \frac{\partial g_s}{\partial T} \frac{\partial f}{\partial T} + g_s \frac{\partial^2 f}{\partial T^2} \right) \quad (71a)$$

$$\frac{\partial^2 s^R}{\partial P^2} = R \frac{\partial^2 g_z}{\partial T^2} + \frac{1}{\epsilon - \sigma} g_s \frac{\partial^2 f}{\partial P^2} \quad (71b)$$

$$\frac{\partial^2 s^R}{\partial T \partial P} = R \frac{\partial^2 g_z}{\partial T \partial P} + \frac{1}{\epsilon - \sigma} \left(\frac{\partial g_s}{\partial T} \frac{\partial f}{\partial P} + g_s \frac{\partial^2 f}{\partial T \partial P} \right) \quad (71c)$$

$$\frac{\partial^2 s^R}{\partial T \partial n_k} = R \frac{\partial^2 g_z}{\partial T \partial n_k} + \frac{1}{\epsilon - \sigma} \left(\frac{\partial^2 g_s}{\partial T \partial n_k} f + \frac{\partial g_s}{\partial T} \frac{\partial f}{\partial n_k} + \frac{\partial g_s}{\partial n_k} \frac{\partial f}{\partial T} + g_s \frac{\partial^2 f}{\partial T \partial n_k} \right) \quad (71d)$$

$$\frac{\partial^2 s^R}{\partial P \partial n_k} = R \frac{\partial^2 g_z}{\partial P \partial n_k} + \frac{1}{\epsilon - \sigma} \left(\frac{\partial g_s}{\partial n_k} \frac{\partial f}{\partial P} + g_s \frac{\partial^2 f}{\partial P \partial n_k} \right) \quad (71e)$$

$$\frac{\partial^2 s^R}{\partial n_l \partial n_k} = R \frac{\partial^2 g_z}{\partial n_l \partial n_k} + \frac{1}{\epsilon - \sigma} \left(\frac{\partial^2 g_s}{\partial n_l \partial n_k} f + \frac{\partial g_s}{\partial n_k} \frac{\partial f}{\partial n_l} + \frac{\partial g_s}{\partial n_l} \frac{\partial f}{\partial n_k} + g_s \frac{\partial^2 f}{\partial n_l \partial n_k} \right) \quad (71f)$$

5.3. Fugacity coefficients

In this report we also present an explicit expression for the logarithmic fugacity coefficients $\{\ln \phi_i\}_{i=1}^{N_C}$ derived from the residual properties (66). The expression for the fugacity coefficients is obtained using the auxiliary functions $f = f(Z, B)$, $g_z = g_z(Z, B)$ and $g_{\phi,i} = g_{\phi,i}(T, n)$

$$\ln \phi_i = (Z - 1) \frac{b_i}{b_m} - g_z - \frac{1}{\epsilon - \sigma} g_{\phi,i} f \quad (72)$$

where the auxiliary function $g_{\phi,i} = g_{\phi,i}(T, n)$ is

$$g_{\phi,i} = \frac{1}{RT b_m} \left(2 \sum_{j=1}^{N_C} z_j a_{ij} - a_m \frac{b_i}{b_m} \right) \quad (73)$$

The first order derivatives are

$$\frac{\partial \ln \phi_i}{\partial T} = \frac{\partial Z}{\partial T} \frac{b_i}{b_m} - \frac{\partial g_z}{\partial T} - \frac{1}{\epsilon - \sigma} \left(\frac{\partial g_{\phi,i}}{\partial T} f + g_{\phi,i} \frac{\partial f}{\partial T} \right) \quad (74a)$$

$$\frac{\partial \ln \phi_i}{\partial P} = \frac{\partial Z}{\partial P} \frac{b_i}{b_m} - \frac{\partial g_z}{\partial P} - \frac{1}{\epsilon - \sigma} g_{\phi,i} \frac{\partial f}{\partial P} \quad (74b)$$

$$\frac{\partial \ln \phi_i}{\partial n_k} = \frac{\partial Z}{\partial n_k} \frac{b_i}{b_m} - (Z - 1) \frac{b_i}{b_m^2} \frac{\partial b_m}{\partial n_k} - \frac{\partial g_z}{\partial n_k} - \frac{1}{(\epsilon - \sigma)} \left(\frac{\partial g_{\phi,i}}{\partial n_k} f + g_{\phi,i} \frac{df}{dn_k} \right) \quad (74c)$$

The second order derivatives are

$$\frac{\partial^2 \ln \phi_i}{\partial T^2} = \frac{\partial^2 Z}{\partial T^2} \frac{b_i}{b_m} - \frac{\partial^2 g_z}{\partial T^2} - \frac{1}{\epsilon - \sigma} \left(\frac{\partial^2 g_{\phi,i}}{\partial T^2} f + 2 \frac{\partial g_{\phi,i}}{\partial T} \frac{\partial f}{\partial T} + g_{\phi,i} \frac{\partial^2 f}{\partial T^2} \right) \quad (75a)$$

$$\frac{\partial^2 \ln \phi_i}{\partial P^2} = \frac{\partial^2 Z}{\partial P^2} \frac{b_i}{b_m} - \frac{\partial^2 g_z}{\partial P^2} - \frac{1}{\epsilon - \sigma} g_{\phi,i} \frac{\partial^2 f}{\partial P^2} \quad (75b)$$

$$\frac{\partial^2 \ln \phi_i}{\partial T \partial P} = \frac{\partial^2 Z}{\partial T \partial P} \frac{b_i}{b_m} - \frac{\partial^2 g_z}{\partial T \partial P} - \frac{1}{\epsilon - \sigma} \left(\frac{\partial g_{\phi,i}}{\partial T} \frac{\partial f}{\partial P} + g_{\phi,i} \frac{\partial^2 f}{\partial T \partial P} \right) \quad (75c)$$

$$\begin{aligned} \frac{\partial^2 \ln \phi_i}{\partial T \partial n_k} &= \frac{\partial^2 Z}{\partial T \partial n_k} \frac{b_i}{b_m} - \frac{\partial Z}{\partial T} \frac{b_i}{b_m^2} \frac{\partial b_m}{\partial n_k} - \frac{\partial^2 g_z}{\partial T \partial n_k} \\ &\quad - \frac{1}{(\epsilon - \sigma)} \left(\frac{\partial^2 g_{\phi,i}}{\partial T \partial n_k} f + \frac{\partial g_{\phi,i}}{\partial T} \frac{df}{dn_k} + \frac{\partial g_{\phi,i}}{\partial n_k} \frac{df}{dT} + g_{\phi,i} \frac{\partial^2 f}{\partial T \partial n_k} \right) \end{aligned} \quad (75d)$$

$$\begin{aligned} \frac{\partial^2 \ln \phi_i}{\partial P \partial n_k} &= \frac{\partial^2 Z}{\partial P \partial n_k} \frac{b_i}{b_m} - \frac{\partial Z}{\partial P} \frac{b_i}{b_m^2} \frac{\partial b_m}{\partial n_k} - \frac{\partial^2 g_z}{\partial P \partial n_k} \\ &\quad - \frac{1}{(\epsilon - \sigma)} \left(\frac{\partial g_{\phi,i}}{\partial n_k} \frac{df}{dP} + g_{\phi,i} \frac{\partial^2 f}{\partial P \partial n_k} \right) \end{aligned} \quad (75e)$$

$$\begin{aligned} \frac{\partial^2 \ln \phi_i}{\partial n_l \partial n_k} &= \frac{\partial^2 Z}{\partial n_l \partial n_k} \frac{b_i}{b_m} - \left(\frac{\partial Z}{\partial n_k} \frac{\partial b_m}{\partial n_l} + \frac{\partial Z}{\partial n_l} \frac{\partial Z}{\partial n_k} \right) \frac{b_i}{b_m^2} + (Z - 1) \frac{b_i}{b_m^2} \left(\frac{2}{b_m} \frac{\partial b_m}{\partial n_k} \frac{\partial b_m}{\partial n_l} - \frac{\partial^2 b_m}{\partial n_l \partial n_k} \right) \\ &\quad - \frac{\partial^2 g_z}{\partial n_l \partial n_k} - \frac{1}{(\epsilon - \sigma)} \left(\frac{\partial^2 g_{\phi,i}}{\partial n_l \partial n_k} f + \frac{\partial g_{\phi,i}}{\partial n_k} \frac{df}{dn_l} + \frac{\partial g_{\phi,i}}{\partial n_l} \frac{df}{dn_k} + g_{\phi,i} \frac{\partial^2 f}{\partial n_l \partial n_k} \right) \end{aligned} \quad (75f)$$

5.4. Auxiliary functions

Note on nomenclature: In order to keep the derivative information brief we introduce the auxiliary variables w_1 and w_2 . Each of these variables are a placeholder for either temperature, T , pressure, P , or a mole number, n_k . We will use these auxiliary variables in cases where the structure of the derivative equations do not depend on the type of variable.

The first order derivatives of $f = f(Z, B)$ with respect to temperature, pressure and mole numbers are expressed through the derivatives with respect to the compressibility factor $Z = Z(T, P, n)$ and $B = B(T, P, n)$

$$\frac{\partial f}{\partial w_1} = \frac{\partial f}{\partial Z} \frac{\partial Z}{\partial w_1} + \frac{\partial f}{\partial B} \frac{\partial B}{\partial w_1}, \quad w_1 \in \{T, P, n_k\} \quad (76)$$

The second order temperature, pressure and composition derivatives are

$$\begin{aligned} \frac{\partial^2 f}{\partial w_1^2} &= \frac{\partial^2 f}{\partial Z^2} \left(\frac{\partial Z}{\partial w_1} \right)^2 + \frac{\partial f}{\partial Z} \frac{\partial^2 Z}{\partial w_1^2} + 2 \frac{\partial^2 f}{\partial Z \partial B} \frac{\partial Z}{\partial w_1} \frac{\partial B}{\partial w_1} \\ &\quad + \frac{\partial^2 f}{\partial B^2} \left(\frac{\partial B}{\partial w_1} \right)^2 + \frac{\partial f}{\partial B} \frac{\partial^2 B}{\partial w_1^2} \end{aligned} \quad (77a)$$

$$\begin{aligned} \frac{\partial^2 f}{\partial w_1 \partial w_2} &= \frac{\partial^2 f}{\partial Z^2} \frac{\partial Z}{\partial w_1} \frac{\partial Z}{\partial w_2} + \frac{\partial^2 f}{\partial B^2} \frac{\partial B}{\partial T} \frac{\partial B}{\partial w_2} + \frac{\partial f}{\partial Z} \frac{\partial^2 Z}{\partial w_1 \partial w_2} \\ &\quad + \frac{\partial f}{\partial B} \frac{\partial^2 B}{\partial w_1 \partial w_2} + \frac{\partial^2 f}{\partial Z \partial B} \left(\frac{\partial Z}{\partial w_1} \frac{\partial B}{\partial w_2} + \frac{\partial Z}{\partial w_2} \frac{\partial B}{\partial w_1} \right) \end{aligned} \quad (77b)$$

where $w_1 \in \{T, P, n_k\}$ and $w_2 \in \{T, P, n_l\}$. **Note:** w_1 and w_2 can represent *different* mole numbers, n_k and n_l , respectively. The first order derivatives of $f = f(Z, B)$ with respect to the compressibility factor Z and B are

$$\frac{\partial f}{\partial Z} = \frac{1}{Z + \epsilon B} - \frac{1}{Z + \sigma B} \quad (78a)$$

$$\frac{\partial f}{\partial B} = \frac{\epsilon}{Z + \epsilon B} - \frac{\sigma}{Z + \sigma B} \quad (78b)$$

The second order derivatives are

$$\frac{\partial^2 f}{\partial Z^2} = -\frac{1}{(Z + \epsilon B)^2} + \frac{1}{(Z + \sigma B)^2} \quad (79a)$$

$$\frac{\partial^2 f}{\partial B^2} = -\left(\frac{\epsilon}{Z + \epsilon B}\right)^2 + \left(\frac{\sigma}{Z + \sigma B}\right)^2 \quad (79b)$$

$$\frac{\partial^2 f}{\partial Z \partial B} = -\frac{\epsilon}{(Z + \epsilon B)^2} + \frac{\sigma}{(Z + \sigma B)^2} \quad (79c)$$

The first order derivatives of $g_h = g_h(T, n)$ are

$$\frac{\partial g_h}{\partial T} = T \frac{\partial g_s}{\partial T} \quad (80a)$$

$$\frac{\partial g_h}{\partial n_k} = T \frac{\partial g_s}{\partial n_k} - \frac{1}{b_m} \left(\frac{\partial a_m}{\partial n_k} - \frac{a_m}{b_m} \frac{\partial b_m}{\partial n_k} \right) \quad (80b)$$

The second order derivatives are

$$\frac{\partial^2 g_h}{\partial T^2} = \frac{\partial g_s}{\partial T} + T \frac{\partial^2 g_s}{\partial T^2} \quad (81a)$$

$$\frac{\partial^2 g_h}{\partial T \partial n_k} = T \frac{\partial^2 g_s}{\partial T \partial n_k} \quad (81b)$$

$$\begin{aligned} \frac{\partial^2 g_h}{\partial n_l \partial n_k} = & T \frac{\partial^2 g_s}{\partial n_l \partial n_k} - \frac{1}{b_m} \frac{\partial^2 a_m}{\partial n_l \partial n_k} \\ & + \frac{1}{b_m^2} \left(\frac{\partial a_m}{\partial n_l} \frac{\partial b_m}{\partial n_k} + \frac{\partial a_m}{\partial n_k} \frac{\partial b_m}{\partial n_l} + a_m \left(\frac{-2}{b_m} \frac{\partial b_m}{\partial n_l} \frac{\partial b_m}{\partial n_k} + \frac{\partial^2 b_m}{\partial n_l \partial n_k} \right) \right) \end{aligned} \quad (81c)$$

The first order derivatives of $g_s = g_s(T, n)$ are

$$\frac{\partial g_s}{\partial T} = \frac{1}{b_m} \frac{\partial^2 a_m}{\partial T^2} \quad (82a)$$

$$\frac{\partial g_s}{\partial n_k} = \frac{1}{b_m} \frac{\partial^2 a_m}{\partial T \partial n_k} - \frac{1}{b_m^2} \frac{\partial b_m}{\partial n_k} \frac{\partial a_m}{\partial T} \quad (82b)$$

The second order derivatives are

$$\frac{\partial^2 g_s}{\partial T^2} = \frac{1}{b_m} \frac{\partial^3 a_m}{\partial T^3} \quad (83a)$$

$$\frac{\partial^2 g_s}{\partial T \partial n_k} = \frac{1}{b_m} \frac{\partial^3 a_m}{\partial T^2 \partial n_k} - \frac{1}{b_m^2} \frac{\partial b_m}{\partial n_k} \frac{\partial^2 a_m}{\partial T^2} \quad (83b)$$

$$\begin{aligned} \frac{\partial^2 g_s}{\partial n_l \partial n_k} = & \frac{1}{b_m^2} \left[\left(\frac{2}{b_m} \frac{\partial b_m}{\partial n_l} \frac{\partial b_m}{\partial n_k} - \frac{\partial^2 b_m}{\partial n_l \partial n_k} \right) \frac{\partial a_m}{\partial T} \right. \\ & \left. - \frac{\partial b_m}{\partial n_k} \frac{\partial^2 a_m}{\partial T \partial n_l} - \frac{\partial b_m}{\partial n_l} \frac{\partial^2 a_m}{\partial T \partial n_k} + b_m \frac{\partial^3 a_m}{\partial T \partial n_l \partial n_k} \right] \quad (83c) \end{aligned}$$

The first order derivatives of $g_z = g_z(Z, B)$ with respect to temperature, pressure and mole numbers are expressed through the derivative with respect to the compressibility factor $Z = Z(T, P, n)$ and $B = B(T, P, n)$

$$\frac{\partial g_z}{\partial w_1} = \frac{\partial g_z}{\partial Z} \frac{\partial Z}{\partial w_1} + \frac{\partial g_z}{\partial B} \frac{\partial B}{\partial w_1}, \quad w_1 \in \{T, P, n_k\} \quad (84)$$

The second order temperature, pressure and composition derivatives are

$$\begin{aligned} \frac{\partial^2 g_z}{\partial w_1^2} = & \frac{\partial^2 g_z}{\partial Z^2} \left(\frac{\partial Z}{\partial w_1} \right)^2 + \frac{\partial g_z}{\partial Z} \frac{\partial^2 Z}{\partial w_1^2} + 2 \frac{\partial^2 g_z}{\partial Z \partial B} \frac{\partial Z}{\partial w_1} \frac{\partial B}{\partial w_1} \\ & + \frac{\partial^2 g_z}{\partial B^2} \left(\frac{\partial B}{\partial w_1} \right)^2 + \frac{\partial g_z}{\partial B} \frac{\partial^2 B}{\partial w_1^2} \quad (85a) \end{aligned}$$

$$\begin{aligned} \frac{\partial^2 g_z}{\partial w_1 \partial w_2} = & \frac{\partial^2 g_z}{\partial Z^2} \frac{\partial Z}{\partial w_1} \frac{\partial Z}{\partial w_2} + \frac{\partial^2 g_z}{\partial B^2} \frac{\partial B}{\partial w_1} \frac{\partial B}{\partial w_2} + \frac{\partial g_z}{\partial Z} \frac{\partial^2 Z}{\partial w_1 \partial w_2} \\ & + \frac{\partial g_z}{\partial B} \frac{\partial^2 B}{\partial w_1 \partial w_2} + \frac{\partial^2 g_z}{\partial Z \partial B} \left(\frac{\partial Z}{\partial w_1} \frac{\partial B}{\partial w_2} + \frac{\partial Z}{\partial w_2} \frac{\partial B}{\partial w_1} \right) \quad (85b) \end{aligned}$$

where $w_1 \in \{T, P, n_k\}$ and $w_2 \in \{T, P, n_l\}$. **Note:** these derivative equations are structurally identical to the derivatives of f (76)-(77). The first order derivatives of $g_z = g_z(Z, B)$ with respect to the compressibility factor Z and B are

$$\frac{\partial g_z}{\partial Z} = \frac{1}{Z - B} \quad (86a)$$

$$\frac{\partial g_z}{\partial B} = -\frac{1}{Z - B} \quad (86b)$$

The second order derivatives are

$$\frac{\partial^2 g_z}{\partial Z^2} = -\frac{1}{(Z - B)^2} \quad (87a)$$

$$\frac{\partial^2 g_z}{\partial Z \partial B} = \frac{1}{(Z - B)^2} \quad (87b)$$

$$\frac{\partial^2 g_z}{\partial B^2} = \frac{1}{(Z - B)^2} \quad (87c)$$

The first order derivatives of the auxiliary function $g_{\phi,i} = g_{\phi,i}(T, n)$ are

$$\frac{\partial g_{\phi,i}}{\partial T} = \frac{1}{T} \left(\frac{1}{Rb_m} \left(2 \sum_{j=1}^{N_c} z_j \frac{\partial a_{ij}}{\partial T} - \frac{\partial a_m}{\partial T} \frac{b_i}{b_m} \right) - g_{\phi,i} \right) \quad (88a)$$

$$\frac{\partial g_{\phi,i}}{\partial n_k} = \frac{1}{b_m} \left[\left(\frac{2}{N} \left(a_{ik} - \sum_{j=1}^{N_c} x_j a_{ij} \right) - \frac{b_i}{b_m} \left(\frac{\partial a_m}{\partial n_k} - \frac{1}{b_m} \frac{\partial b_m}{\partial n_k} \right) \right) - \frac{\partial b_m}{\partial n_k} g_{\phi,i} + \frac{1}{RT} \right] \quad (88b)$$

The second order derivatives are

$$\frac{\partial^2 g_{\phi,i}}{\partial T^2} = \frac{1}{T} \left(\frac{1}{Rb_m} \left(2 \sum_{j=1}^{N_c} z_j \frac{\partial^2 a_{ij}}{\partial T^2} - \frac{\partial^2 a_m}{\partial T^2} \frac{b_i}{b_m} \right) - 2 \frac{\partial g_{\phi,i}}{\partial T} \right) \quad (89a)$$

$$\begin{aligned} \frac{\partial^2 g_{\phi,i}}{\partial T \partial n_k} &= -\frac{1}{T} \left(\frac{\partial g_{\phi,i}}{\partial n_k} + \frac{1}{b_m} \frac{\partial b_m}{\partial n_k} g_{\phi,i} \right) \\ &+ \frac{1}{b_m} \left[-\frac{\partial b_m}{\partial n_k} \frac{\partial g_{\phi,i}}{\partial T} + \frac{1}{RT} \left(\frac{\partial a_{ik}}{\partial T} - \sum_{j=1}^{N_c} x_j \frac{\partial a_{ij}}{\partial T} \right) - \frac{1}{b_m} \left(\frac{\partial^2 a_m}{\partial T \partial n_k} - \frac{1}{b_m} \frac{\partial b_m}{\partial n_k} \right) b_i \right], \end{aligned} \quad (89b)$$

$$\begin{aligned} \frac{\partial^2 g_{\phi,i}}{\partial n_l \partial n_k} &= -\frac{1}{b_m} \left(\frac{\partial g_{\phi,i}}{\partial n_l} \frac{\partial b_m}{\partial n_k} + \frac{\partial b_m}{\partial n_l} \frac{\partial g_{\phi,i}}{\partial n_k} + \frac{\partial^2 b_m}{\partial n_l \partial n_k} \right) + \frac{1}{RT} \frac{1}{b_m} \left[\frac{2}{N^2} \left(2 \sum_{j=1}^{N_c} x_j a_{ij} - a_{il} - a_{ik} \right) \right. \\ &\left. - \frac{1}{b_m} \left\{ \frac{\partial^2 a_m}{\partial n_l \partial n_k} - \frac{\partial a_m}{\partial n_k} \frac{1}{b_m} \frac{\partial b_m}{\partial n_l} - \frac{1}{b_m^2} \left[\left(\frac{\partial a_m}{\partial n_l} - 2a_m \frac{1}{b_m} \frac{\partial b_m}{\partial n_l} \right) \frac{\partial b_m}{\partial n_k} + a_m \frac{\partial^2 b_m}{\partial n_l \partial n_k} \right] \right\} b_i \right] \end{aligned} \quad (89c)$$

5.5. Compressibility factor

The compressibility factor is implicitly defined by the cubic equation (63) for a given temperature pressure and composition. The first order derivatives of the compressibility factor $Z = Z(T, P, n)$ are given in terms of the derivatives of the polynomial $q = q(Z)$ given in (63)-(64).

$$\frac{\partial Z}{\partial w_1} = - \left(\frac{\partial q}{\partial Z} \right)^{-1} \frac{\partial q}{\partial w_1}, \quad w_1 \in \{T, P, n_k\} \quad (90a)$$

The second order derivatives are

$$\frac{\partial^2 Z}{\partial w_1^2} = - \left(\frac{\partial q}{\partial Z} \right)^{-1} \left(\frac{\partial^2 q}{\partial w_1^2} + \frac{\partial^2 q}{\partial Z^2} \left(\frac{\partial Z}{\partial w_1} \right)^2 + 2 \frac{\partial^2 q}{\partial w_1 \partial Z} \frac{\partial Z}{\partial w_1} \right) \quad (91a)$$

$$\frac{\partial^2 Z}{\partial w_1 \partial w_2} = - \left(\frac{\partial q}{\partial Z} \right)^{-1} \left(\frac{\partial^2 q}{\partial w_1 \partial w_2} + \frac{\partial^2 q}{\partial Z^2} \frac{\partial Z}{\partial w_1} \frac{\partial Z}{\partial w_2} + \frac{\partial^2 q}{\partial w_1 \partial Z} \frac{\partial Z}{\partial w_2} + \frac{\partial^2 q}{\partial w_2 \partial Z} \frac{\partial Z}{\partial w_1} \right) \quad (91b)$$

where $w_1 \in \{T, P, n_k\}$ and $w_2 \in \{T, P, n_l\}$. The first order derivatives of the cubic polynomial $q = q(Z)$ are expressed through the derivatives of the polynomial coefficients $\{d_m = d_m(A, B)\}_{m=0}^2$

$$\frac{\partial q}{\partial Z} = 3Z^2 + \sum_{m=1}^2 m d_m Z^{m-1} \quad (92a)$$

$$\frac{\partial q}{\partial w_1} = \sum_{m=0}^2 \frac{\partial d_m}{\partial w_1} Z^m, \quad w_1 \in \{T, P, n_k\} \quad (92b)$$

The second order derivatives are

$$\frac{\partial^2 q}{\partial Z^2} = 6Z + 2d_2 \quad (93a)$$

$$\frac{\partial^2 q}{\partial Z \partial w_1} = \sum_{m=1}^2 m \frac{\partial d_m}{\partial w_1} Z^{m-1} \quad (93b)$$

$$\frac{\partial^2 q}{\partial w_1^2} = \sum_{m=0}^2 \frac{\partial^2 d_m}{\partial w_1^2} Z^m \quad (93c)$$

$$\frac{\partial^2 q}{\partial w_1 \partial w_2} = \sum_{m=0}^2 \frac{\partial^2 d_m}{\partial w_1 \partial w_2} Z^m \quad (93d)$$

where $w_1 \in \{T, P, n_k\}$ and $w_2 \in \{T, P, n_l\}$.

5.6. Polynomial coefficients

The temperature, pressure and composition derivatives of the polynomial coefficients are expressed through the derivatives with respect to $A = A(T, P, n)$ and $B = B(T, P, n)$. The first order derivatives of $\{d_m = d_m(A, B)\}_{k=0}^2$ are

$$\frac{\partial d_m}{\partial w_1} = \frac{\partial d_m}{\partial A} \frac{\partial A}{\partial w_1} + \frac{\partial d_m}{\partial B} \frac{\partial B}{\partial w_1}, \quad w_1 \in \{T, P, n_k\} \quad (94)$$

The second order derivatives are

$$\begin{aligned} \frac{\partial^2 d_m}{\partial w_1^2} &= \frac{\partial^2 d_m}{\partial A^2} \left(\frac{\partial A}{\partial w_1} \right)^2 + \frac{\partial d_m}{\partial A} \frac{\partial^2 A}{\partial w_1^2} + 2 \frac{\partial^2 d_m}{\partial A \partial B} \frac{\partial A}{\partial w_1} \frac{\partial B}{\partial w_1} \\ &\quad + \frac{\partial^2 d_m}{\partial B^2} \left(\frac{\partial B}{\partial w_1} \right)^2 + \frac{\partial d_m}{\partial B} \frac{\partial^2 B}{\partial w_1^2} \end{aligned} \quad (95a)$$

$$\begin{aligned} \frac{\partial^2 d_m}{\partial w_1 \partial w_2} &= \frac{\partial^2 d_m}{\partial A^2} \frac{\partial A}{\partial w_1} \frac{\partial A}{\partial w_2} + \frac{\partial^2 d_m}{\partial B^2} \frac{\partial B}{\partial w_1} \frac{\partial B}{\partial w_2} + \frac{\partial d_m}{\partial A} \frac{\partial^2 A}{\partial w_1 \partial w_2} \\ &\quad + \frac{\partial d_m}{\partial B} \frac{\partial^2 B}{\partial w_1 \partial w_2} + \frac{\partial^2 d_m}{\partial A \partial B} \left(\frac{\partial A}{\partial w_1} \frac{\partial B}{\partial w_2} + \frac{\partial A}{\partial w_2} \frac{\partial B}{\partial w_1} \right) \end{aligned} \quad (95b)$$

where $w_1 \in \{T, P, n_k\}$ and $w_2 \in \{T, P, n_l\}$. **Note:** w_1 and w_2 can represent *different* mole numbers, n_k and n_l , respectively. The first order derivatives of the polynomial coefficient $d_2 = d_2(A, B)$ with respect to A and B are

$$\frac{\partial d_2}{\partial A} = 0 \quad (96a)$$

$$\frac{\partial d_2}{\partial B} = (\epsilon + \sigma - 1) \quad (96b)$$

The second order derivatives are

$$\frac{\partial^2 d_2}{\partial A^2} = 0 \quad (97a)$$

$$\frac{\partial^2 d_2}{\partial B^2} = 0 \quad (97b)$$

$$\frac{\partial^2 d_2}{\partial A \partial B} = 0 \quad (97c)$$

The first order derivatives of $d_1 = d_1(A, B)$ are

$$\frac{\partial d_1}{\partial A} = 1 \quad (98a)$$

$$\frac{\partial d_1}{\partial B} = -(\epsilon + \sigma) + 2(\epsilon\sigma - \epsilon - \sigma)B \quad (98b)$$

The second order derivatives are

$$\frac{\partial^2 d_1}{\partial A^2} = 0 \quad (99a)$$

$$\frac{\partial^2 d_1}{\partial B^2} = 2(\epsilon\sigma - \epsilon - \sigma) \quad (99b)$$

$$\frac{\partial^2 d_1}{\partial A \partial B} = 0 \quad (99c)$$

The first order derivatives of $d_0 = d_0(A, B)$ are

$$\frac{\partial d_0}{\partial A} = -B \quad (100a)$$

$$\frac{\partial d_0}{\partial B} = -(A + \epsilon\sigma(2B + 3B^2)) \quad (100b)$$

The second order derivatives are

$$\frac{\partial^2 d_0}{\partial A^2} = 0 \quad (101a)$$

$$\frac{\partial^2 d_0}{\partial B^2} = -\epsilon\sigma(2 + 6B) \quad (101b)$$

$$\frac{\partial^2 d_0}{\partial A \partial B} = -1 \quad (101c)$$

5.7. The quantities A and B

The first order derivatives of $A = A(T, P, n)$ are given in terms of the derivatives of the mixing parameter $a_m = a_m(T, n)$

$$\frac{\partial A}{\partial T} = \frac{\partial a_m}{\partial T} \frac{P}{R^2 T^2} - \frac{2}{T} A \quad (102a)$$

$$\frac{\partial A}{\partial P} = \frac{a_m}{R^2 T^2} \quad (102b)$$

$$\frac{\partial A}{\partial n_k} = \frac{\partial a_m}{\partial n_k} \frac{P}{R^2 T^2} \quad (102c)$$

The second order derivatives are

$$\frac{\partial^2 A}{\partial T^2} = \frac{P}{R^2 T^2} \left(\frac{\partial^2 a_m}{\partial T^2} - \frac{1}{T} \frac{\partial a_m}{\partial T} \right) - \frac{3}{T} \frac{\partial A}{\partial T} \quad (103a)$$

$$\frac{\partial^2 A}{\partial P^2} = 0 \quad (103b)$$

$$\frac{\partial^2 A}{\partial T \partial P} = \frac{\partial a_m}{\partial T} \frac{1}{R^2 T^2} - \frac{2}{T} \frac{\partial A}{\partial P} \quad (103c)$$

$$\frac{\partial^2 A}{\partial T \partial n_k} = \frac{\partial^2 a_m}{\partial T \partial n_k} \frac{P}{R^2 T^2} - \frac{2}{T} \frac{\partial A}{\partial n_k} \quad (103d)$$

$$\frac{\partial^2 A}{\partial P \partial n_k} = \frac{\partial a_m}{\partial n_k} \frac{1}{R^2 T^2} \quad (103e)$$

$$\frac{\partial^2 A}{\partial n_l \partial n_k} = \frac{\partial^2 a_m}{\partial n_l \partial n_k} \frac{P}{R^2 T^2} \quad (103f)$$

The first order derivatives of $B = B(T, P, n)$ are given in terms of the derivatives of the mixing parameter $b_m = b_m(n)$

$$\frac{\partial B}{\partial T} = -\frac{b_m P}{RT^2} \quad (104a)$$

$$\frac{\partial B}{\partial P} = \frac{b_m}{RT} \quad (104b)$$

$$\frac{\partial B}{\partial n_k} = \frac{\partial b_m}{\partial n_k} \frac{P}{RT} \quad (104c)$$

The second order derivatives are

$$\frac{\partial^2 B}{\partial T^2} = 2 \frac{b_m P}{RT^3} \quad (105a)$$

$$\frac{\partial^2 B}{\partial P^2} = 0 \quad (105b)$$

$$\frac{\partial^2 B}{\partial T \partial P} = -\frac{b_m}{RT^2} \quad (105c)$$

$$\frac{\partial^2 B}{\partial T \partial n_k} = -\frac{\partial b_m}{\partial n_k} \frac{P}{RT^2} \quad (105d)$$

$$\frac{\partial^2 B}{\partial P \partial n_k} = \frac{\partial b_m}{\partial n_k} \frac{1}{RT} \quad (105e)$$

$$\frac{\partial^2 B}{\partial n_l \partial n_k} = \frac{\partial^2 b_m}{\partial n_l \partial n_k} \frac{P}{RT} \quad (105f)$$

5.8. Mixing parameters

The derivatives of the mixing parameters $a_m = a_m(T, n)$ exploit the symmetry of $a_{ij} = a_{ji}(T) = a_{ji}(T)$. The first order derivatives are

$$\frac{\partial a_m}{\partial T} = \sum_{i=1}^{N_C} \sum_{j=1}^{N_C} z_i z_j \frac{\partial a_{ij}}{\partial T} \quad (106a)$$

$$\frac{\partial a_m}{\partial n_k} = \frac{2}{N} \left(\sum_{i=1}^{N_c} z_i a_{ik} - a_m \right) \quad (106b)$$

The second order derivatives are

$$\frac{\partial^2 a_m}{\partial T^2} = \sum_{i=1}^{N_c} \sum_{j=1}^{N_c} z_i z_j \frac{\partial^2 a_{ij}}{\partial T^2} \quad (107a)$$

$$\frac{\partial^2 a_m}{\partial T \partial n_k} = \frac{2}{N} \left(\sum_{i=1}^{N_c} z_i \frac{\partial a_{ik}}{\partial T} - \frac{\partial a_m}{\partial T} \right) \quad (107b)$$

$$\frac{\partial^2 a_m}{\partial n_l \partial n_k} = \frac{1}{N} \left(a_{kl} + a_m - \frac{\partial a_m}{\partial n_k} - \frac{\partial a_m}{\partial n_l} \right) \quad (107c)$$

The relevant third order derivatives are

$$\frac{\partial^3 a_m}{\partial T^3} = \sum_{i=1}^{N_c} \sum_{j=1}^{N_c} z_i z_j \frac{\partial^3 a_{ij}}{\partial T^3} \quad (108a)$$

$$\frac{\partial^3 a_m}{\partial T^2 \partial n_k} = \frac{2}{N} \left(\sum_{i=1}^{N_c} z_i \frac{\partial^2 a_{ik}}{\partial T^2} - \frac{\partial^2 a_m}{\partial T^2} \right) \quad (108b)$$

$$\frac{\partial^3 a_m}{\partial T \partial n_l \partial n_k} = \frac{1}{N} \left(\frac{\partial a_{kl}}{\partial T} + \frac{\partial a_m}{\partial T} - \frac{\partial^2 a_m}{\partial T \partial n_k} - \frac{\partial^2 a_m}{\partial T \partial n_l} \right) \quad (108c)$$

The derivatives of the mixing parameter $b_m = b_m(n)$ are

$$\frac{\partial b_m}{\partial n_k} = \frac{b_k - b_m}{N} \quad (109a)$$

$$\frac{\partial^2 b_m}{\partial n_l \partial n_k} = \frac{2b_m - b_l - b_k}{N^2} \quad (109b)$$

The derivatives of $a_{ij} = a_{ij}(T)$ are given in terms of the derivatives of the auxiliary function $\hat{a}_{ij} = \hat{a}_{ij}(T)$

$$\frac{\partial a_{ij}}{\partial T} = \frac{1 - k_{ij}}{2\sqrt{\hat{a}_{ij}}} \frac{\partial \hat{a}_{ij}}{\partial T} \quad (110a)$$

$$\frac{\partial^2 a_{ij}}{\partial T^2} = \frac{1 - k_{ij}}{2\sqrt{\hat{a}_{ij}}} \left(\frac{\partial^2 \hat{a}_{ij}}{\partial T^2} - \frac{1}{2\hat{a}_{ij}} \left(\frac{\partial \hat{a}_{ij}}{\partial T} \right)^2 \right) \quad (110b)$$

$$\frac{\partial^3 a_{ij}}{\partial T^3} = -\frac{1 - k_{ij}}{4\sqrt{\hat{a}_{ij}}} \left(\frac{1}{\hat{a}_{ij}} \frac{\partial^2 \hat{a}_{ij}}{\partial T^2} + 2 \frac{\partial^3 \hat{a}_{ij}}{\partial T^3} \right) + \frac{1}{2\hat{a}_{ij}} \left(\frac{1}{\hat{a}_{ij}} \frac{\partial \hat{a}_{ij}}{\partial T} \frac{\partial a_{ij}}{\partial T} - \frac{\partial^2 a_{ij}}{\partial T^2} \right) \quad (110c)$$

The derivatives of the auxiliary function $\hat{a}_{ij} = \hat{a}_{ij}(T)$ are given in terms of derivatives of the pure component properties $a_i = a_i(T)$

$$\frac{\partial \hat{a}_{ij}}{\partial T} = \frac{\partial a_i}{\partial T} a_j + a_i \frac{\partial a_j}{\partial T} \quad (111a)$$

$$\frac{\partial^2 \hat{a}_{ij}}{\partial T^2} = \frac{\partial^2 a_i}{\partial T^2} a_j + 2 \frac{\partial a_i}{\partial T} \frac{\partial a_j}{\partial T} + a_i \frac{\partial^2 a_j}{\partial T^2} \quad (111b)$$

$$\frac{\partial^3 \hat{a}_{ij}}{\partial T^3} = \frac{\partial^3 a_i}{\partial T^3} a_j + 3 \frac{\partial^2 a_i}{\partial T^2} \frac{\partial a_j}{\partial T} + 3 \frac{\partial a_i}{\partial T} \frac{\partial^2 a_j}{\partial T^2} + a_i \frac{\partial^3 a_j}{\partial T^3} \quad (111c)$$

5.9. Pure component properties

The pure component parameters $a_i = a_i(T)$ are directly proportional to $\alpha = \alpha(T_{r,i}, \omega_i)$ and as such their derivatives are

$$\frac{\partial a_i}{\partial T} = \frac{\partial \alpha}{\partial T} \Psi \frac{R^2 T_{c,i}^2}{P_{c,i}} \quad (112a)$$

$$\frac{\partial^2 a_i}{\partial T^2} = \frac{\partial^2 \alpha}{\partial T^2} \Psi \frac{R^2 T_{c,i}^2}{P_{c,i}} \quad (112b)$$

$$\frac{\partial^3 a_i}{\partial T^3} = \frac{\partial^3 \alpha}{\partial T^3} \Psi \frac{R^2 T_{c,i}^2}{P_{c,i}} \quad (112c)$$

The derivatives of $\alpha = \alpha(T_{r,i}, \omega_i)$ are

$$\frac{\partial \alpha}{\partial T} = -\alpha \frac{m(\omega_i)}{\sqrt{\alpha T T_{c,i}}} \quad (113a)$$

$$\frac{\partial^2 \alpha}{\partial T^2} = -\frac{1}{2} \frac{\partial \alpha}{\partial T} \left(\frac{1}{T} - \frac{1}{\alpha} \frac{\partial \alpha}{\partial T} \right) \quad (113b)$$

$$\frac{\partial^3 \alpha}{\partial T^3} = -\frac{1}{2} \left[\frac{\partial^2 \alpha}{\partial T^2} \left(\frac{1}{T} - \frac{1}{\alpha} \frac{\partial \alpha}{\partial T} \right) + \frac{\partial \alpha}{\partial T} \left(\left(\frac{1}{\alpha} \frac{\partial \alpha}{\partial T} \right)^2 - \frac{1}{\alpha} \frac{\partial^2 \alpha}{\partial T^2} - \frac{1}{T^2} \right) \right] \quad (113c)$$

Appendix A. Solution of cubic equations

There exists a number of approaches for solving the cubic equation of state (56) for the roots when pressure and temperature are given. These approaches are either direct approaches that use explicit formula for computing the roots, iterative approaches that approximate the roots of interest or a combination of both where the direct solution is refined by an iterative approach in order to remove imprecision arising from rounding errors. In this work we use an iterative approach as described by Smith et al. (2005) and compare to Cardano's approach which is briefly described by Monroy-Loperena (2012). The equation of state (56) is rewritten in terms of the compressibility factor $Z = PV/(RT)$

$$\begin{aligned} Z^3 - Z^2(1 - B(\epsilon + \sigma - 1)) \\ - Z(\epsilon + \sigma - B(\epsilon\sigma - \epsilon - \sigma) - A/B)B \\ - (A + B(1 + B)\epsilon\sigma)B = 0, \end{aligned} \quad (A.1)$$

where A and B are given by

$$A = \frac{a_m(T, n)P}{R^2 T^2} \quad (\text{A.2})$$

$$B = \frac{Pb_m(n)}{RT} \quad (\text{A.3})$$

The equation of state (A.1) is written compactly

$$q(Z) = Z^3 + d_1 Z^2 + d_2 Z + d_3 = 0 \quad (\text{A.4})$$

Cardano's direct approach

The number of real roots are determined by the two quantities Q and R

$$Q = (d_1^2 - 3d_2)/9 \quad (\text{A.5})$$

$$R = (2d_1^3 - 9d_1 d_2 + 27d_3)/54 \quad (\text{A.6})$$

There are three real roots if $R^2 \leq Q^3$. In that case, the roots are found by the formula

$$Z_1 = -2\sqrt{Q}\cos(\theta/3) - d_1/3 \quad (\text{A.7a})$$

$$Z_2 = -2\sqrt{Q}\cos((\theta + 2\pi)/3) - d_1/3 \quad (\text{A.7b})$$

$$Z_3 = -2\sqrt{Q}\cos((\theta - 2\pi)/3) - d_1/3 \quad (\text{A.7c})$$

where θ is computed by

$$\theta = \arccos(R/\sqrt{Q^3}) \quad (\text{A.8})$$

If $R^2 > Q^3$, there is one real root and two complex conjugate roots that are given by

$$Z_1 = (S + T) - d_1/3 \quad (\text{A.9a})$$

$$Z_2 = -1/2(S + T) - d_1/3 + i\sqrt{3}/2(S - T) \quad (\text{A.9b})$$

$$Z_3 = -1/2(S + T) - d_1/3 - i\sqrt{3}/2(S - T) \quad (\text{A.9c})$$

where

$$S = -\text{sgn}(R)\left(|R| + \sqrt{R^2 - Q^3}\right)^{1/3} \quad (\text{A.10})$$

$$T = \begin{cases} Q/S & (S \neq 0) \\ 0 & (S = 0) \end{cases} \quad (\text{A.11})$$

In the case of multiple roots, the smallest represents the liquid phase compressibility factor, $Z^l = \min\{Z_1, Z_2, Z_3\}$, and the largest is vapor phase compressibility factor, $Z^v = \max\{Z_1, Z_2, Z_3\}$.

An iterative Newton approach

The approach described here uses Newton iterations to solve the cubic equation (A.1). It is possible to use higher-order methods as discussed by Olivera-Fuentes (1993), due to the cubic nature of the equation. In the Newton approach, an initial guess, Z_0 , is iteratively improved by

$$Z_{k+1} = Z_k - q(Z_k)/q'(Z_k) \quad (\text{A.12})$$

$$q'(Z_k) = 3Z_k^2 + 2d_1 Z_k + d_2 \quad (\text{A.13})$$

The iterative sequence is terminated when both of the following criteria are satisfied

$$|Z_{k+1} - Z_k| < \epsilon \quad (\text{A.14})$$

$$|Z_{k+1}^3 + d_1 Z_{k+1}^2 + d_2 Z_{k+1} + d_3| < \epsilon \quad (\text{A.15})$$

Once the sequence is terminated, a single root has been found. The following initial estimates are used, depending on whether the compressibility factor of the vapor phase, Z^v , or of the liquid phase, Z^l , is sought

$$Z_0^v = 1 \quad (\text{A.16})$$

$$Z_0^l = B \quad (\text{A.17})$$

References

- Monroy-Loperena, R., 2012. A note on the analytical solution of cubic equations of state in process simulation. *Industrial & Engineering Chemistry Research* 51, 6972–6976.
- Olivera-Fuentes, C., 1993. The optimal solution of cubic equations of state. *Latin American Applied Research* 23, 243–256.
- Smith, J.M., Van Ness, H.C., Abbott, M.M., 2005. *Introduction to Chemical Engineering Thermodynamics*. 7th ed., McGraw-Hill, New York, NY.
- Thomson, G., 1996. The DIPPR® databases. *International Journal of Thermophysics* 17, 223–232.

Technical Report II

Computation of Phase Equilibrium and Phase Envelopes

Authors:

Tobias K. S. Ritschel and John Bagterp Jørgensen

DTU Compute Technical Report-2017-11, 2017,
Department of Applied Mathematics and Computer Science,
Technical University of Denmark.

Computation of Phase Equilibrium and Phase Envelopes

Abstract

In this technical report, we describe the computation of phase equilibrium and phase envelopes based on expressions for the fugacity coefficients. We derive those expressions from the residual Gibbs energy. We consider 1) ideal gases and liquids modeled with correlations from the DIPPR database and 2) nonideal gases and liquids modeled with cubic equations of state. Next, we derive the equilibrium conditions for an isothermal-isobaric (constant temperature, constant pressure) vapor-liquid equilibrium process (PT flash), and we present a method for the computation of phase envelopes. We formulate the involved equations in terms of the fugacity coefficients. We present expressions for the first-order derivatives. Such derivatives are necessary in computationally efficient gradient-based methods for solving the vapor-liquid equilibrium equations and for computing phase envelopes. Finally, we describe a Matlab program that computes the phase envelope of a mixture. We present the source code and discuss practical details of the implementation.



Tobias K. S. Ritschel
John Bagterp Jørgensen

DTU Compute Technical Report-2017-11

ISSN: 1601-2321

Technical University of Denmark

Department of Applied Mathematics and Computer Science &

Center for Energy Resources Engineering

Kgs. Lyngby

December 15, 2017

Contents

1	Introduction	4
1.1	A note on nomenclature	5
2	Residual Gibbs energy of ideal vapor-liquid mixtures	5
2.1	Gibbs energy of ideal gas mixtures	6
2.2	Gibbs energy of ideal liquid mixtures	7
2.3	Residual Gibbs energy of ideal vapor and liquid mixtures	9
2.4	Summary	9
3	Residual Gibbs energy of nonideal mixtures	9
3.1	Residual Gibbs energy with pressure-explicit equations of state	10
3.2	Cubic equations of state	11
3.2.1	Solution of cubic equations of state	12
3.3	Residual Gibbs energy with cubic equations of state	13
3.4	Summary	13
4	Fugacity and fugacity coefficients	13
4.1	Definition of chemical potential	14
4.2	Chemical potential of a pure component ideal gas	14
4.3	Definition of fugacity	14
4.4	Fugacity coefficients	15
4.4.1	Ideal gas mixture	16
4.4.2	Ideal liquid mixture	16
4.4.3	Nonideal mixture	17
4.5	Summary	17
5	Vapor-liquid equilibrium	17
5.1	Equilibrium conditions	17
5.2	Solution of the equilibrium conditions	19
5.3	Summary	19
6	Equilibrium constants	19
6.1	Equilibrium constants	19
6.1.1	Ideal vapor-liquid mixture	20
6.2	Summary	20
7	Vapor-liquid equilibrium of ideal mixtures	20
7.1	Solution of the Rachford-Rice equation	22
7.2	Summary	22

8	Computation of phase envelopes	23
8.1	Solution of the isocurve equations	24
8.2	Computation of initial guess	24
8.3	Summary	25
9	Derivatives	26
9.1	Logarithmic fugacity coefficients	26
9.1.1	Ideal liquid mixture	26
9.1.2	Nonideal mixture	27
9.2	Vapor-liquid equilibrium equations	30
9.3	Ideal vapor-liquid equilibrium equations	31
9.4	Phase envelope equations	31
10	An algorithm for computing phase envelopes and an example program	32
10.1	Wilson's approximation of equilibrium constants	32
10.2	Cubic interpolation	32
10.3	Automatic selection of specified variable	34
10.4	Automatic step size selection	34
10.5	Algorithm for computing isocurves	34
10.6	Example program	34
10.7	Discussion	44
A	Derivative of mixing term	45
B	Derivation of the residual Gibbs energy for cubic equations of state	46
C	Derivation of the logarithmic fugacity coefficients for cubic equations of state	47

1 Introduction

This technical report is structured in the following way: the purpose of Sections 2-4 is to derive expressions for the fugacity coefficients which are relevant to the vapor-liquid equilibrium equations that we present in Section 5 and 7. The vapor-liquid equilibrium equations (and therefore also the fugacity coefficients) are central in the phase envelope computations that we describe in Section 8. In the remainder of this section, we give a brief overview of the content of the sections.

We derive expressions for the residual Gibbs energy for ideal gas and liquid mixtures in Section 2 and for nonideal gas and liquid mixtures in Section 3. The residual Gibbs energy of ideal gases is zero by definition. We obtain the expression for the residual Gibbs energy of ideal liquids directly from the Gibbs energy of the ideal gas and liquid. We construct expressions for the residual Gibbs energy of nonideal mixtures from the residual Helmholtz energy. That is because the partial derivative of Helmholtz energy with respect to volume is negative pressure and because pressure appears explicitly in cubic equations of state. We do not use the expressions for the residual Gibbs energy in the actual computations. We only use them to derive expressions for the fugacity coefficients in Section 4.

In Section 4, we first introduce the chemical potential which is the partial derivative of Gibbs energy with respect to a given mole number. Next, we define the fugacity based on the chemical potential of the mixture of interest and the chemical potential of a pure component ideal gas. We then define the fugacity coefficient and derive an expression for the fugacity coefficient based on the residual Gibbs energy. The fugacity and the chemical potentials are not involved in the actual computations. They only serve to introduce the relation between the fugacity coefficients and the residual Gibbs energy.

In Section 5, we derive vapor-liquid equilibrium conditions for an isothermal-isobaric (constant temperature, constant pressure) vapor-liquid mixture. The second law of thermodynamics states that the entropy of a closed system is maximal at equilibrium. An isothermal-isobaric system is not closed, but the combination of the isothermal-isobaric system and its surroundings is. The condition of maximal entropy of the isothermal-isobaric system and its surroundings is equivalent to a condition of minimal Gibbs energy for the system alone (Callen, 1985). We therefore formulate the vapor-liquid equilibrium problem as an optimization problem where the objective function is the Gibbs energy of the system. The equilibrium conditions are then the first-order optimality conditions of this optimization problem. We rewrite these first-order optimality conditions in terms of the fugacity coefficients.

In Section 6, we introduce the equilibrium constants (sometimes called equilibrium ratios) and use the vapor-liquid equilibrium conditions to derive an expression for the equilibrium constants based on the fugacity coefficients. In Section 7, we use the fact that the equilibrium constants for an ideal vapor-liquid mixture are independent of the composition to solve the vapor-liquid equilibrium problem efficiently.

In section 8, we describe the equations that define what is called the phase envelope. The phase envelope is a collection of vapor-liquid equilibrium states that have the same vapor fraction, i.e. the same total moles of vapor as compared to the total moles of both vapor and liquid. The phase envelope equations are therefore based on the vapor-liquid equilibrium conditions. The method for phase envelope computations that we present is described by Michelsen and Mollerup (2007,

Chap. 12). The method involves a choice of equilibrium constants, temperature, and pressure as independent variables, i.e. the variables that are solved for. The method constructs the phase envelope in a sequential manner for a given specified vapor fraction. It solves the equations for a specified equilibrium constant, temperature, or pressure. Next, it uses the sensitivities of the solution to create an initial estimate for solving the equations with a slightly different specified equilibrium constant, temperature or pressure.

We collect the derivatives of the expressions for the fugacity coefficients, the vapor-liquid equilibrium equations, and the phase envelope equations in Section 9.

In Section 10, we describe, in more detail, the algorithm by Michelsen and Mollerup (2007) for computing the phase envelope. We also describe an example of a Matlab program that implements the algorithm and computes the phase envelope for a hydrocarbon mixture. We present and discuss the programs Matlab code.

1.1 A note on nomenclature

In Sections 2-4, we describe thermodynamic properties of either a vapor or a liquid mixture. When we describe properties of vapor mixtures, e.g. the Gibbs energy of an ideal gas, we denote the composition (in moles) by a vector, n^v . In that case, we denote the mole fraction of component i by y_i . For liquid mixtures, we denote the composition vector n^l and the mole fraction x_i . When we describe the properties of nonideal mixtures, and when we define fugacity and fugacity coefficients, the expressions are the same for both vapor mixtures and liquid mixtures. We therefore denote the composition by n and the mole fraction by z_i .

Later, in Section 5 to 8, we consider mixtures that exist in both a vapor phase and a liquid phase. Again, the vapor phase has composition n^v and mole fraction y_i , and the liquid phase has composition n^l and mole fraction x_i . But in that case, n will denote the composition of the entire mixture, i.e. $n_i = n_i^v + n_i^l$, and z_i will be the corresponding mole fraction.

2 Residual Gibbs energy of ideal vapor-liquid mixtures

We derive the Gibbs energy of ideal gas and liquid mixtures from the enthalpy and entropy of those mixtures. We use the pure component molar enthalpy and entropy of each component in the mixture to derive expressions for the enthalpy and the entropy of the mixtures. The residual Gibbs energy of a mixture is, by definition, the Gibbs energy of that mixture minus the Gibbs energy of an ideal gas at the same state, e.g. temperature, pressure, and composition (in moles). The residual Gibbs energy of an ideal gas mixture is therefore zero by definition. We use the Gibbs energy of an ideal gas mixture and an ideal liquid mixture to derive an expression for the residual Gibbs energy of a liquid mixture.

2.1 Gibbs energy of ideal gas mixtures

We consider an ideal gas mixture of N_C components. The molar enthalpy and entropy of the i 'th component at temperature T and pressure P are

$$h_i^{ig}(T) = h_i^{ig}(T_0) + \int_{T_0}^T c_{P,i}^{ig}(T) dT, \quad (1a)$$

$$s_i^{ig}(T, P) = s_i^{ig}(T_0, P_0) + \int_{T_0}^T \frac{c_{P,i}^{ig}(T)}{T} dT - R \ln \frac{P}{P_0}. \quad (1b)$$

The ideal gas enthalpy is independent of pressure. $h_i^{ig}(T_0)$ and $s_i^{ig}(T_0, P_0)$ are the ideal gas enthalpy and entropy of formation at reference temperature T_0 and P_0 . $c_{P,i}^{ig}(T)$ is the ideal gas heat capacity at constant pressure. Values of $h_i^{ig}(T_0)$ and $s_i^{ig}(T_0, P_0)$ and correlations for $c_{P,i}^{ig}(T)$ are available in databases such as the DIPPR database¹ (Thomson, 1996). However, terms that involve $h_i^{ig}(T_0)$, $s_i^{ig}(T_0, P_0)$, and $c_{P,i}^{ig}(T)$ will cancel out in the computations that we describe in this technical report. The molar gibbs energy of component i is

$$\begin{aligned} g_i^{ig}(T, P) &= h_i^{ig}(T) - T s_i^{ig}(T, P) \\ &= h_i^{ig}(T_0) - T s_i^{ig}(T_0, P_0) - RT \ln P_0 + \int_{T_0}^T c_{P,i}^{ig}(T) dT - T \int_{T_0}^T \frac{c_{P,i}^{ig}(T)}{T} dT + RT \ln P \\ &= \Gamma_i(T) + RT \ln P, \end{aligned} \quad (2)$$

where we have introduced the auxiliary variable

$$\Gamma_i(T) = h_i^{ig}(T_0) - T s_i^{ig}(T_0, P_0) - RT \ln P_0 + \int_{T_0}^T c_{P,i}^{ig}(T) dT - T \int_{T_0}^T \frac{c_{P,i}^{ig}(T)}{T} dT. \quad (3)$$

$h_i^{ig}(T_0)$, $s_i^{ig}(T_0, P_0)$, and $c_{P,i}^{ig}(T)$ appear only in the auxiliary variable $\Gamma_i(T)$ and it is this variable that will cancel out in the computations in later sections. The enthalpy and entropy of the ideal gas mixture are

$$H^{ig}(T, n^v) = \sum_i n_i^v h_i^{ig}(T), \quad (4a)$$

$$S^{ig}(T, P, n^v) = \sum_i n_i^v s_i^{ig}(T, P) - R \sum_i n_i^v \ln y_i. \quad (4b)$$

n^v is a vector of compositions (in moles). The i 'th component, n_i^v , denotes the number of moles of component i . y_i is the vapor mole fraction of component i : $y_i = n_i^v / (\sum_j n_j^v)$. The sums are over all components. The second term in the expression for the entropy is a mixing term. The Gibbs

¹www.aiche.org/dippr/events-products/801-database

energy of the ideal gas mixture is

$$\begin{aligned}
 G^{ig}(T, P, n^v) &= H^{ig}(T, n^v) - TS^{ig}(T, P, n^v) \\
 &= \sum_i n_i^v h_i^{ig}(T) - T \sum_i n_i^v s_i^{ig}(T, P) + RT \sum_i n_i^v \ln y_i \\
 &= \sum_i n_i^v g_i^{ig}(T, P) + RT \sum_i n_i^v \ln y_i \\
 &= \sum_i n_i^v (\Gamma_i(T) + RT \ln P) + RT \sum_i n_i^v \ln y_i.
 \end{aligned} \tag{5}$$

2.2 Gibbs energy of ideal liquid mixtures

The vaporization change in molar enthalpy of component i , $\Delta h_i^{vap}(T)$, is related to the vaporization change in molar entropy, $\Delta s_i^{vap}(T)$, by

$$\Delta h_i^{vap}(T) = T \Delta s_i^{vap}(T). \tag{6}$$

We do not present an expression for $\Delta s_i^{vap}(T)$ because it will cancel out in later expressions. The molar ideal liquid enthalpy at saturation is

$$\begin{aligned}
 h_i^{sat}(T) &= h_i^{ig}(T) - \Delta h_i^{vap}(T) \\
 &= h_i^{ig}(T) - T \Delta s_i^{vap}(T).
 \end{aligned} \tag{7}$$

It is necessary to know the saturation pressure in order to provide expressions for the molar ideal liquid entropy at saturation. We use the following correlation from the DIPPR database

$$P_i^{sat}(T) = \exp(\ln P_i^{sat}(T)), \tag{8a}$$

$$\ln P_i^{sat}(T) = A_i + \frac{B_i}{T} + C_i \ln(T) + D_i T^{E_i}. \tag{8b}$$

A_i , B_i , C_i , D_i , and E_i are component-specific parameters. They are also specific to this particular correlation. The ideal liquid entropy at saturation is

$$s_i^{sat}(T) = s_i^{ig}(T, P_i^{sat}(T)) - \Delta s_i^{vap}(T). \tag{9}$$

The molar ideal liquid Gibbs energy at saturation is

$$\begin{aligned}
 g_i^{sat}(T) &= h_i^{sat}(T) - T s_i^{sat}(T) \\
 &= h_i^{ig}(T) - T \Delta s_i^{vap}(T) - T s_i^{ig}(T, P_i^{sat}(T)) + T \Delta s_i^{vap}(T) \\
 &= h_i^{ig}(T) - T s_i^{ig}(T, P_i^{sat}(T)) \\
 &= \Gamma_i(T) + RT \ln P_i^{sat}(T).
 \end{aligned} \tag{10}$$

In order to provide expressions for the molar ideal liquid enthalpy and entropy at pressures, P , that are different from the saturation pressure, $P_i^{sat}(T)$, it is necessary to know the liquid volume as a function of temperature, as well as its first order temperature derivative. We use the following

2. Residual Gibbs energy of ideal vapor-liquid mixtures

correlation from the DIPPR database

$$v_i^l(T) = \frac{B_i^{1+(1-\frac{T}{C_i})^{D_i}}}{A_i}. \quad (11)$$

Again, A_i , B_i , C_i , and D_i are component-specific parameters. They are also specific for this correlation, i.e. they are different from the parameters in the correlation for the saturation pressure. The molar ideal liquid enthalpy and entropy are

$$h_i^{id}(T, P) = h_i^{sat}(T) + \left(v_i^l(T) - T \frac{\partial v_i^l}{\partial T} \right) (P - P_i^{sat}(T)), \quad (12a)$$

$$s_i^{id}(T, P) = s_i^{sat}(T) - \frac{\partial v_i^l}{\partial T} (P - P_i^{sat}(T)). \quad (12b)$$

Terms that involve the temperature derivative will cancel out in later expressions. The molar ideal liquid Gibbs energy is

$$\begin{aligned} g_i^{id}(T, P) &= h_i^{id}(T, P) - T s_i^{id}(T, P) \\ &= h_i^{sat}(T) + \left(v_i^l(T) - T \frac{\partial v_i^l}{\partial T} \right) (P - P_i^{sat}(T)) - T s_i^{sat}(T) + T \frac{\partial v_i^l}{\partial T} (P - P_i^{sat}(T)) \\ &= h_i^{sat}(T) - T s_i^{sat}(T) + v_i^l(T) (P - P_i^{sat}(T)) \\ &= g_i^{sat}(T) + v_i^l(T) (P - P_i^{sat}(T)) \\ &= \Gamma_i(T) + RT \ln P_i^{sat}(T) + v_i^l(T) (P - P_i^{sat}(T)). \end{aligned} \quad (13)$$

The ideal liquid mixture enthalpy and entropy are

$$H^{id}(T, P, n^l) = \sum_i n_i^l h_i^{id}(T, P), \quad (14a)$$

$$S^{id}(T, P, n^l) = \sum_i n_i^l s_i^{id}(T, P) - R \sum_i n_i^l \ln x_i. \quad (14b)$$

n^l is a vector of compositions in moles, i.e. n_i^l is the moles of component i . $x_i = n_i^l / (\sum_j n_j^l)$ is the liquid mole fraction of component i . The second term in the expression for entropy is a mixing term. The ideal liquid mixture Gibbs energy is

$$\begin{aligned} G^{id}(T, P, n^l) &= H^{id}(T, P, n^l) - T S^{id}(T, P, n^l) \\ &= \sum_i n_i^l h_i^{id}(T, P) - T \sum_i n_i^l s_i^{id}(T, P) + RT \sum_i n_i^l \ln x_i \\ &= \sum_i n_i^l g_i^{id}(T, P) + RT \sum_i n_i^l \ln x_i \\ &= \sum_i n_i^l (\Gamma_i(T) + RT \ln P_i^{sat}(T) + v_i^l(T) (P - P_i^{sat}(T))) + RT \sum_i n_i^l \ln x_i. \end{aligned} \quad (15)$$

2.3 Residual Gibbs energy of ideal vapor and liquid mixtures

The residual Gibbs energy is the difference between the Gibbs energy of the mixture and the Gibbs energy of the mixture if it was considered to be an ideal gas mixture. The residual Gibbs energy of an ideal gas mixture is therefore zero:

$$G^{R,ig}(T, P, n^v) = 0 \quad (16)$$

We use the Gibbs energy of ideal gas mixtures to derive the expression for the residual Gibbs energy of an ideal liquid mixture. It is the difference between the ideal liquid Gibbs energy and the ideal gas Gibbs energy evaluated at the same temperature, T , pressure, P , and composition, n^l :

$$\begin{aligned} G^{R,id}(T, P, n^l) &= G^{id}(T, P, n^l) - G^{ig}(T, P, n^l) \\ &= \left(\sum_i n_i^l (\Gamma_i(T) + RT \ln P_i^{sat}(T) + v_i^l(T)(P - P_i^{sat}(T))) + RT \sum_i n_i^l \ln x_i \right) \\ &\quad - \left(\sum_i n_i^l (\Gamma_i(T) + RT \ln P) + RT \sum_i n_i^l \ln x_i \right) \\ &= \sum_i n_i^l \left(RT \ln \frac{P_i^{sat}(T)}{P} + v_i^l(T)(P - P_i^{sat}(T)) \right) \end{aligned} \quad (17)$$

It is important to note that the ideal gas Gibbs energy is evaluated at the liquid composition, n^l , and not n^v in the above expression. That is also the reason that x_i appears in place of y_i which means that certain terms cancel out. Unlike the ideal gas heat capacity, $c_{P,i}^{ig}(T)$, the saturation pressure, $P_i^{sat}(T)$, and the liquid volume, $v_i^l(T)$, are necessary in the computations that we present in this report.

2.4 Summary

The residual Gibbs energy of ideal gas and liquid mixtures are

$$G^{R,ig}(T, P, n^v) = 0, \quad (18a)$$

$$G^{R,id}(T, P, n^l) = \sum_i n_i^l \left(RT \ln \frac{P_i^{sat}(T)}{P} + v_i^l(T)(P - P_i^{sat}(T)) \right). \quad (18b)$$

They will be used in expressions for the fugacity coefficients in Section 4.

3 Residual Gibbs energy of nonideal mixtures

In this section, we first present an expression for the residual Gibbs energy of a nonideal mixture without assuming a specific equation of state. Then we describe cubic equations of state with van der Waals' mixing rules. Finally, we present the expression for the residual Gibbs energy of a nonideal mixture based on a cubic equation of state. The derivation of the expression based on the cubic equation of state is presented in Appendix B. We will not use the expression for the Gibbs energy

of an ideal gas mixture that was derived in Section 2. Instead, we use the ideal gas law for the ideal gas mixture.

3.1 Residual Gibbs energy with pressure-explicit equations of state

In this section, we consider a nonideal vapor or liquid mixture. We treat both phases in the same way. We use the residual Helmholtz energy in the derivation. We will consider the Helmholtz energy a function of temperature, T , volume, V , and composition (in moles), n . This is only for the sake of the derivation. Eventually, we describe the residual Gibbs energy as function of temperature, T , pressure, P , and mole numbers, n . We omit the superscript v or l on the composition vector because we treat both phases in the same way. The residual Helmholtz energy is

$$A^R(T, V, n) = A(T, V, n) - A^{ig}(T, V, n). \quad (19)$$

$A(T, V, n)$ is the Helmholtz energy of the mixture and $A^{ig}(T, V, n)$ is the Helmholtz energy of the mixture if it was an ideal gas. We use the fact that the volume derivative of Helmholtz energy is pressure with a negative sign:

$$\begin{aligned} \frac{\partial A^R}{\partial V}(T, V, n) &= \frac{\partial A}{\partial V}(T, V, n) - \frac{\partial A^{ig}}{\partial V}(T, V, n) \\ &= - (P(T, V, n) - P^{ig}(T, V, n)). \end{aligned} \quad (20)$$

$P^{ig}(T, V, n)$ is obtained from the ideal gas law, i.e. $P^{ig}(T, V, n) = NRT/V$, where $N = \sum_j n_j$ is the total number of moles in the mixture and R is the gas constant. $P(T, V, n)$ is obtained from an equation of state. The derivation that we present in this section is therefore best suited to equations of state that are explicit in pressure. That is the case for the cubic equations of state that we describe later. We obtain an expression for the residual Helmholtz energy of the nonideal mixture:

$$A^R(T, V, n) = - \int_{V=\infty}^V \left(P(T, V, n) - \frac{NRT}{V} \right) dV. \quad (21)$$

The choice of integrating from infinite volume is a matter of convenience. We introduce the compressibility factor, $Z = \frac{PV}{NRT}$. We use relations between residual properties to derive an expression for the residual Gibbs energy evaluated at T , P , and n (Michelsen and Mollerup, 2007, Table 6):

$$\begin{aligned} G^R(T, P, n) &= G^R(T, V, n) - NRT \ln Z \\ &= A^R(T, V, n) + PV - NRT - NRT \ln Z \\ &= - \int_{V=\infty}^V \left(P(T, V, n) - \frac{NRT}{V} \right) dV + NRTZ - NRT - NRT \ln Z \\ &= - \int_{V=\infty}^V \left(P(T, V, n) - \frac{NRT}{V} \right) dV + NRT(Z - 1) - NRT \ln Z. \end{aligned} \quad (22)$$

The above expression only becomes useful when we substitute a specific equation of state for $P(T, V, n)$.

3.2 Cubic equations of state

In this section, we describe cubic equation equations of state and van der Waals' mixing rules. The cubic equations of state are formulated in terms of the molar volume, $v = V/N$:

$$P = P(T, V, n) = \frac{RT}{v - b_m} - \frac{a_m}{(v + \epsilon b_m)(v + \sigma b_m)} \\ = \frac{RT}{V/N - b_m} - \frac{a_m}{(V/N + \epsilon b_m)(V/N + \sigma b_m)}. \quad (23)$$

The parameters ϵ and σ are specific for each cubic equation of state. Two of the most popular cubic equations of state were developed by Soave (1972) and by Peng and Robinson (1976). The parameters that are specific to each equation of state are shown in Table 1. a_m and b_m are mixing parameters given by van der Waals' mixing rules:

$$a_m = a_m(T, n) = \sum_i \sum_j z_i z_j a_{ij} \quad (24a)$$

$$b_m = b_m(n) = \sum_i z_i b_i \quad (24b)$$

$z_i = n_i/N$ is the mole fraction of component i . We define the pure component parameter b_i shortly. The binary parameters a_{ij} are

$$a_{ij} = a_{ij}(T) = (1 - k_{ij})\sqrt{\hat{a}_{ij}}. \quad (25)$$

We assume that the binary interaction parameters, k_{ij} , are given constants, i.e. we do not provide an expression for them. They can either be measured experimentally or predicted with a model. The parameter \hat{a}_{ij} is the product of the pure component parameters a_i and a_j :

$$\hat{a}_{ij} = \hat{a}_{ij}(T) = a_i a_j. \quad (26)$$

The pure component parameters a_i and b_i are

$$a_i = a_i(T) = \alpha(T_{r,i}, \omega_i) \Psi \frac{R^2 T_{c,i}^2}{P_{c,i}}, \quad (27a)$$

$$b_i = \Omega \frac{RT_{c,i}}{P_{c,i}}. \quad (27b)$$

$T_{r,i} = T/T_{c,i}$ is the reduced temperature. $T_{c,i}$ and $P_{c,i}$ are the critical temperature and pressure of component i . ω_i is the acentricity factor. We use values from the DIPPR database for the critical temperature, pressure, and for the acentricity factor. Ψ and Ω are parameters that are specific to the equation of state. Their values are shown in Table 1. The function, α , is

$$\alpha(T_{r,i}, \omega_i) = \left(1 + m(\omega_i)(1 - T_{r,i}^{1/2})\right)^2. \quad (28)$$

$m(\omega_i)$ is a polynomial that is specific for each equation of state. Its expression is shown in Table 1. This concludes the description of the cubic equations of state. Next, we describe how to solve the

3. Residual Gibbs energy of nonideal mixtures

Table 1: Parameters in the Soave-Redlich-Kwong (SRK) and the Peng-Robinson (PR) cubic equations of state.

	ϵ	σ	Ω	Ψ	$m(\omega)$
PR	$1 + \sqrt{2}$	$1 - \sqrt{2}$	0.07779	0.45724	$m(\omega) = 0.37464 + 1.54226\omega - 0.26992\omega^2$
SRK	1	0	0.08664	0.42748	$m(\omega) = 0.480 + 1.574\omega - 0.176\omega^2$

cubic equation of state of state for the compressibility factor.

3.2.1 Solution of cubic equations of state

We reformulate the cubic equation of state as a polynomial in the compressibility factor Z :

$$q(Z) = Z^3 + \sum_{m=0}^2 d_m Z^m = 0. \quad (29)$$

The polynomial coefficients are

$$d_2 = d_2(A, B) = B(\epsilon + \sigma - 1) - 1, \quad (30a)$$

$$d_1 = d_1(A, B) = A - B(\epsilon + \sigma) + B^2(\epsilon\sigma - \epsilon - \sigma), \quad (30b)$$

$$d_0 = d_0(A, B) = -(AB + (B^2 + B^3)\epsilon\sigma), \quad (30c)$$

where A and B are functions of temperature, T , pressure, P , and composition, n :

$$A = A(T, P, n) = \frac{Pa_m}{R^2T^2}, \quad (31a)$$

$$B = B(T, P, n) = \frac{Pb_m}{RT}. \quad (31b)$$

The compressibility factors are therefore also functions of temperature, T , pressure, P , and composition, n , i.e. $Z = Z(T, P, n)$. The polynomial $q(Z) = q(Z; T, P, n)$ will have either one or three roots depending on the given T , P , and n . When it has three roots, the smallest root is the liquid phase compressibility and the largest root is the vapor phase compressibility. This is essentially what distinguishes the residual Gibbs energy of the vapor phase from the residual Gibbs energy of the liquid phase. For given T , P , and n , we use Newton's method to solve for the compressibility factor:

$$Z_{k+1} = Z_k - q(Z_k)/q'(Z_k), \quad (32a)$$

$$q'(Z_k) = 3Z_k^2 + 2d_2Z_k + d_1. \quad (32b)$$

We use the stopping criterium

$$|q(Z_{k+1})| < \epsilon, \quad (33)$$

4. Fugacity and fugacity coefficients

for a given tolerance, ϵ . The initial guess for the Newton iterations depends on whether we are searching for a vapor root or a liquid root:

$$Z_0 = 1, \quad (\text{Vapor}) \quad (34a)$$

$$Z_0 = B. \quad (\text{Liquid}) \quad (34b)$$

It is important to note that in vapor-liquid equilibrium computations, the vapor and the liquid phases will have different compositions, n^v and n^l . It is therefore necessary to solve the polynomial $q(Z; T, P, n^v)$ for the vapor compressibility and $q(Z; T, P, n^l)$ for the liquid compressibility. The compressibilities are not roots of the same polynomial.

3.3 Residual Gibbs energy with cubic equations of state

We insert the cubic equation of state into the expression for the residual Gibbs energy. The evaluation of the integral is described in Appendix B. The final expression becomes

$$\begin{aligned} G^R(T, P, n) &= - \int_{V=\infty}^V \left(\frac{RT}{V/N - b_m} - \frac{a_m}{(V/N + \epsilon b_m)(V/N + \sigma b_m)} - \frac{NRT}{V} \right) dV \\ &\quad + NRT(Z - 1) - NRT \ln Z \\ &= -NRT \ln \left(\frac{Z - B}{Z} \right) - \frac{N}{\epsilon - \sigma} \frac{a_m}{b_m} \ln \left(\frac{Z + \epsilon B}{Z + \sigma B} \right) + NRT(Z - 1) - NRT \ln Z \\ &= NRT(Z - 1) - NRT \ln(Z - B) - \frac{N}{\epsilon - \sigma} \frac{a_m}{b_m} \ln \left(\frac{Z + \epsilon B}{Z + \sigma B} \right). \end{aligned} \quad (35)$$

We reiterate that the above expression describes the residual Gibbs energy of both the vapor phase, $G^{R,v}(T, P, n^v)$, and of the liquid phase, $G^{R,l}(T, P, n^l)$. The cubic equation of state is a third order polynomial, $q(Z; T, P, n)$, in the compressibility factor. The vapor phase compressibility, $Z^v = Z^v(T, P, n^v)$, is the largest root of $q(Z^v; T, P, n^v)$ and the liquid phase compressibility factor, $Z^l = Z^l(T, P, n^l)$, is the smallest root of $q(Z^l; T, P, n^l)$.

3.4 Summary

The residual Gibbs energy of a nonideal mixture (either a vapor or a liquid mixture) based on a cubic equation of state is

$$G^R(T, P, n) = NRT(Z - 1) - NRT \ln(Z - B) - \frac{N}{\epsilon - \sigma} \frac{a_m}{b_m} \ln \left(\frac{Z + \epsilon B}{Z + \sigma B} \right). \quad (36)$$

4 Fugacity and fugacity coefficients

In this section, we introduce the chemical potential in order to define the fugacity. Next, we introduce the fugacity coefficients which are the quantities that we will actually use in the vapor-liquid equilibrium computations. That is, the chemical potentials and the fugacities only serve the purpose of introducing expressions for the fugacity coefficients.

4.1 Definition of chemical potential

The chemical potential of component i in a mixture is by definition the partial derivative of the Gibbs energy of that mixture with respect to mole number i :

$$\mu_i(T, P, n) = \frac{\partial G}{\partial n_i}(T, P, n). \quad (37)$$

The same definition holds for any mixture, i.e. also for ideal gas and liquid mixtures:

$$\mu_i^{ig}(T, P, n) = \frac{\partial G^{ig}}{\partial n_i^v}(T, P, n^v), \quad (38a)$$

$$\mu_i^{ld}(T, P, n) = \frac{\partial G^{ld}}{\partial n_i^l}(T, P, n^l). \quad (38b)$$

4.2 Chemical potential of a pure component ideal gas

The definition of the fugacity of component i depends on the chemical potential of a pure component (pc) ideal gas consisting entirely of component i . The Gibbs energy of such a pure component ideal gas is

$$\begin{aligned} G_i^{pc,ig}(T, P, n_i^v) &= n_i^v g_i^{ig}(T, P) \\ &= n_i^v (\Gamma_i(T) + RT \ln P). \end{aligned} \quad (39)$$

The chemical potential of that pure component ideal gas is therefore

$$\begin{aligned} \mu_i^{pc,ig}(T, P, n_i^v) &= \frac{\partial G_i^{pc,ig}}{\partial n_i^v}(T, P, n_i^v) \\ &= g_i^{ig}(T, P) \\ &= \Gamma_i(T) + RT \ln P. \end{aligned} \quad (40)$$

We note that the pure component ideal gas chemical potential is independent of the mole number, i.e. $\mu_i^{pc,ig}(T, P, n_i^v) = \mu_i^{pc,ig}(T, P)$.

4.3 Definition of fugacity

We define the fugacity of a mixture (vapor or liquid) of composition n . First, we recall that the Gibbs energy of an ideal gas mixture of composition n^v (in moles) is

$$G^{ig}(T, P, n^v) = \sum_i n_i^v (\Gamma_i(T) + RT \ln P) + RT \sum_i n_i^v \ln y_i. \quad (41)$$

Next, we express the ideal gas mixture chemical potential of component i in terms of the chemical potential of a pure component ideal gas that consists of component i . Because we define fugacity in the same way for both vapor and liquid mixtures, we use n for the composition instead of n^v . Similarly, we use $z_i = n_i/N$, where $N = \sum_j n_j$, for the mole fraction of component i . The i 'th

chemical potential of an ideal gas mixture is

$$\begin{aligned}
 \mu_i^{ig}(T, P, n) &= \frac{\partial G^{ig}}{\partial n_i^v}(T, P, n) \\
 &= \Gamma_i(T) + RT \ln P + RT \ln z_i \\
 &= \Gamma_i(T) + RT \ln P_0 + RT \ln \frac{z_i P}{P_0} \\
 &= \mu_i^{pc,ig}(T, P_0) + RT \ln \frac{z_i P}{P_0}.
 \end{aligned} \tag{42}$$

The differentiation of the first sum in the expression (41) for the Gibbs energy of an ideal gas mixture is straightforward. We describe the differentiation of the second sum in Appendix A. The fugacity of component i , $f_i(T, P, n)$, is defined such that the above expression is valid for the chemical potential of mixtures that are not ideal gas mixtures. It is therefore implicitly defined by replacing $z_i P$ with $f_i(T, P, n)$ in the above:

$$\mu_i(T, P, n) = \mu_i^{pc,ig}(T, P_0) + RT \ln \frac{f_i(T, P, n)}{P_0}. \tag{43}$$

Note that it is still the chemical potential of a pure component ideal gas that appears on the right-hand side of the above expression. Next, we introduce the fugacity coefficients and use the above expression to relate the fugacity coefficients to the residual Gibbs energy.

4.4 Fugacity coefficients

In this section, we introduce the fugacity coefficients and relate them to the residual Gibbs energy. First, we isolate the chemical potential of the pure component ideal gas in (42):

$$\mu_i^{pc,ig}(T, P_0) = \mu_i^{ig}(T, P, n) - RT \ln \frac{z_i P}{P_0}. \tag{44}$$

Then we substitute the chemical potential of the pure component ideal gas into (43):

$$\begin{aligned}
 \mu_i(T, P, n) &= \mu_i^{pc,ig}(T, P_0) + RT \ln \frac{f_i(T, P, n)}{P_0} \\
 &= \mu_i^{ig}(T, P, n) - RT \ln \frac{z_i P}{P_0} + RT \ln \frac{f_i(T, P, n)}{P_0} \\
 &= \mu_i^{ig}(T, P, n) + RT \ln \frac{f_i(T, P, n)}{z_i P}.
 \end{aligned} \tag{45}$$

We define the fugacity coefficient of component i as

$$\phi_i(T, P, n) = \frac{f_i(T, P, n)}{z_i P}. \tag{46}$$

4. Fugacity and fugacity coefficients

We substitute the fugacity coefficient into (45), isolate the term that contains the fugacity coefficient, and use the definition of the chemical potentials:

$$\begin{aligned}
 RT \ln \phi_i(T, P, n) &= \mu_i(T, P, n) - \mu_i^{ig}(T, P, n) \\
 &= \frac{\partial G}{\partial n_i}(T, P, n) - \frac{\partial G^{ig}}{\partial n_i}(T, P, n) \\
 &= \frac{\partial}{\partial n_i} (G(T, P, n) - G^{ig}(T, P, n)) \\
 &= \frac{\partial G^R}{\partial n_i}(T, P, n)
 \end{aligned} \tag{47}$$

We thus obtain an expression for the logarithm of the fugacity coefficients:

$$\ln \phi_i(T, P, n) = \frac{1}{RT} \frac{\partial G^R}{\partial n_i}(T, P, n). \tag{48}$$

It is only the logarithms of the fugacity coefficients that we use in the vapor-liquid equilibrium computations. It is therefore not necessary to isolate the fugacity coefficients in the above. Next, we derive expressions for the logarithmic fugacity coefficients of ideal and nonideal gas and liquid mixtures.

4.4.1 Ideal gas mixture

The residual Gibbs energy of ideal gas mixtures is zero. The logarithmic fugacity coefficients are therefore also zero:

$$\begin{aligned}
 \ln \phi_i^{ig}(T, P, n^v) &= \frac{1}{RT} \frac{\partial G^{R,ig}}{\partial n_i}(T, P, n) \\
 &= 0.
 \end{aligned} \tag{49}$$

4.4.2 Ideal liquid mixture

The residual Gibbs energy of ideal liquid mixtures is linear in the mole numbers. The differentiation is therefore straightforward. The logarithmic fugacity coefficients are

$$\begin{aligned}
 \ln \phi_i^{id}(T, P, n^l) &= \frac{1}{RT} \frac{\partial G^{R,id}}{\partial n_i^l} \\
 &= \frac{1}{RT} \left(RT \ln \frac{P_i^{sat}(T)}{P} + v_i^l(T)(P - P_i^{sat}(T)) \right) \\
 &= \ln \frac{P_i^{sat}(T)}{P} + \frac{v_i^l(T)(P - P_i^{sat}(T))}{RT}.
 \end{aligned} \tag{50}$$

The fugacity coefficients of ideal liquids are therefore independent of the composition, i.e. $\phi_i^{id}(T, P, n^l) = \phi_i^{id}(T, P)$.

4.4.3 Nonideal mixture

We present the derivation of the fugacity coefficients for nonideal mixtures in Appendix C. The final expression is

$$\ln \phi_i(T, P, n) = (Z - 1) \frac{b_i}{b_m} - \ln(Z - B) - \frac{1}{\epsilon - \sigma} \frac{1}{RTb_m} \left(2 \sum_j z_j a_{ij} - a_m \frac{b_i}{b_m} \right) \ln \left(\frac{Z + \epsilon B}{Z + \sigma B} \right). \quad (51)$$

The above expression applies to both vapor and liquid mixtures. The logarithmic fugacity coefficients of vapor mixtures, $\ln \phi_i^v(T, P, n^v)$, use the vapor phase compressibility, $Z^v = Z^v(T, P, n^v)$, which is the largest root of the cubic equation of state, $q(Z^v; T, P, n^v)$. The logarithmic fugacity coefficients of liquid mixtures use the liquid phase compressibility, $Z^l(T, P, n^l)$, which is the smallest root of $q(Z^l; T, P, n^l)$. Furthermore, several of the quantities in the above expression depend on the composition. It is therefore different values of $a_m(T, n)$, $b_m(n)$, $B(T, P, n)$, and z_i that appear in the expressions for the vapor and liquid logarithmic fugacity coefficients.

4.5 Summary

The expressions for the logarithmic fugacity coefficients are

$$\ln \phi_i^{ig}(T, P, n^v) = 0, \quad (52a)$$

$$\ln \phi_i^{id}(T, P, n^l) = \ln \frac{P_i^{sat}(T)}{P} + \frac{v_i^l(T)(P - P_i^{sat}(T))}{RT}, \quad (52b)$$

$$\ln \phi_i(T, P, n) = (Z - 1) \frac{b_i}{b_m} - \ln(Z - B) - \frac{1}{\epsilon - \sigma} \frac{1}{RTb_m} \left(2 \sum_j z_j a_{ij} - a_m \frac{b_i}{b_m} \right) \ln \left(\frac{Z + \epsilon B}{Z + \sigma B} \right). \quad (52c)$$

The latter expression applies to both nonideal vapor and liquid mixtures.

5 Vapor-liquid equilibrium

In this section, we formulate the vapor-liquid equilibrium problem as an optimization problem. We then derive the vapor-liquid equilibrium conditions as the first-order optimality conditions of this optimization problem. Next, we reformulate the equilibrium conditions in terms of the fugacity coefficients that we introduced in Section 4. Finally, we use Newton's method to solve the equilibrium conditions.

5.1 Equilibrium conditions

We consider a mixture that exists in both a vapor phase (v) and a liquid phase (l). The mixture contains N_C components. The vapor phase has composition n^v , and the liquid phase has composition n^l . Both n^v and n^l are vectors of mole numbers. Both phases have the same temperature, T , and the same pressure, P . The mixture is isothermal and isobaric, i.e. the temperature and the pressure are

5. Vapor-liquid equilibrium

specified constants. The problem is to find the vapor-liquid composition (in moles) at equilibrium. The condition of equilibrium is that Gibbs energy of the mixture is minimal. We formulate this condition as the optimization problem

$$\min_{n^v, n^l} G^v(T, P, n^v) + G^l(T, P, n^l), \quad (53a)$$

$$s.t. \quad n_i^v + n_i^l = n_i, \quad i = 1, \dots, N_C. \quad (53b)$$

The total moles of each component, n_i , is specified. The constraint ensures that mass is conserved, i.e. that mass is distributed among the vapor phase and the liquid phase. Because the constraint is linear, we use it to eliminate the liquid moles, i.e. $n^l = n - n^v$. By doing so, we obtain an unconstrained optimization problem:

$$\min_{n^v} G^v(T, P, n^v) + G^l(T, P, n - n^v). \quad (54)$$

The first-order optimality conditions require that the derivatives of the objective function with respect to the vapor mole numbers are zero:

$$\frac{\partial G^v}{\partial n_i^v}(T, P, n^v) - \frac{\partial G^l}{\partial n_i^l}(T, P, n - n^v) = 0. \quad (55)$$

We have used that $\frac{\partial}{\partial n_i^v}(G^l(T, P, n - n^v)) = -\frac{\partial G^l}{\partial n_i^l}(T, P, n - n^v)$. For brevity, we write n^l instead of $n - n^v$ in most of the following equations in this section, but it is implicitly understood that n^l has been eliminated. Next, we rewrite the equilibrium conditions in terms of the chemical potentials:

$$\begin{aligned} \frac{\partial G^v}{\partial n_i^v}(T, P, n^v) - \frac{\partial G^l}{\partial n_i^l}(T, P, n^l) &= \mu_i^v(T, P, n^v) - \mu_i^l(T, P, n^l) \\ &= \left(\mu_i^v(T, P, n^v) - \mu_i^{pc,ig}(T, P_0) \right) - \left(\mu_i^l(T, P, n^l) - \mu_i^{pc,ig}(T, P_0) \right). \end{aligned} \quad (56)$$

We can obtain an expression for the difference in each parentheses from the definition of the fugacities:

$$\begin{aligned} \left(\mu_i^v(T, P, n^v) - \mu_i^{pc,ig}(T, P_0) \right) - \left(\mu_i^l(T, P, n^l) - \mu_i^{pc,ig}(T, P_0) \right) \\ = RT \ln \frac{f_i^v(T, P, n^v)}{P_0} - RT \ln \frac{f_i^l(T, P, n^l)}{P_0}. \end{aligned} \quad (57)$$

From the definition of the fugacity coefficients, we know that $f_i^v(T, P, n^v) = \phi_i^v(T, P, n^v)y_iP$ and $f_i^l(T, P, n^l) = \phi_i^l(T, P, n^l)x_iP$. We use those expressions to introduce the fugacity coefficients:

$$\begin{aligned} RT \ln \frac{f_i^v(T, P, n^v)}{P_0} - RT \ln \frac{f_i^l(T, P, n^l)}{P_0} &= RT (\ln f_i^v(T, P, n^v) - \ln f_i^l(T, P, n^l)) \\ &= RT (\ln (\phi_i^v(T, P, n^v)y_iP) - \ln (\phi_i^l(T, P, n^l)x_iP)) \\ &= RT (\ln \phi_i^v(T, P, n^v) + \ln y_i - \ln \phi_i^l(T, P, n^l) - \ln x_i). \end{aligned} \quad (58)$$

6. Equilibrium constants

We recall that we have eliminated the liquid mole numbers, i.e. $n^l = n - n^v$. We solve the vapor-liquid equilibrium problem by solving the following nonlinear equations for the vapor mole numbers, n^v :

$$g_i(T, P, n^v) = \ln \phi_i^v(T, P, n^v) + \ln y_i - \ln \phi_i^l(T, P, n - n^v) - \ln x_i = 0. \quad (59)$$

The liquid mole numbers are $x_i = n_i^l / \left(\sum_j n_j^l \right) = (n_i - n_i^v) / \left(\sum_j n_j - n^v \right)$.

5.2 Solution of the equilibrium conditions

We solve the vapor-liquid equilibrium conditions with Newton's method:

$$n^{v,k+1} = n^{v,k} - \left(\frac{\partial g}{\partial n^v} \right)^{-1} g(T, P, n^{v,k}). \quad (60)$$

5.3 Summary

The vapor-liquid equilibrium conditions for a vapor-liquid mixture at temperature T , pressure P , and total composition n are

$$g_i(T, P, n^v) = \ln \phi_i^v(T, P, n^v) + \ln y_i - \ln \phi_i^l(T, P, n - n^v) - \ln x_i = 0. \quad (61)$$

We solve the above equilibrium conditions for the vapor composition, n^v . The liquid mole numbers are $x_i = (n_i - n_i^v) / \left(\sum_j n_j - n^v \right)$. We solve the vapor-liquid equilibrium equations with Newton's method:

$$n^{v,k+1} = n^{v,k} - \left(\frac{\partial g}{\partial n^v} \right)^{-1} g(T, P, n^{v,k}). \quad (62)$$

After solution, the liquid mole numbers are computed by $n^l = n - n^v$.

6 Equilibrium constants

In this section, we introduce the equilibrium constants and provide an expression for them in terms of the logarithmic fugacity coefficients. The equilibrium constants are useful when solving vapor-liquid equilibrium problems for ideal gas and liquid mixtures. We also use them as independent variables in the phase envelope computations in Section 8.

6.1 Equilibrium constants

The i 'th equilibrium constant (sometimes called the equilibrium ratio) is the ratio between the vapor mole fraction and the liquid mole fraction of component i :

$$K_i = \frac{y_i}{x_i}. \quad (63)$$

7. Vapor-liquid equilibrium of ideal mixtures

We can therefore express the equilibrium conditions,

$$\ln \phi_i^v(T, P, n^v) + \ln y_i - \ln \phi_i^l(T, P, n - n^v) - \ln x_i = 0, \quad (64)$$

in terms of the equilibrium constants:

$$\ln K_i + \ln \phi_i^v(T, P, n^v) - \ln \phi_i^l(T, P, n^l) = 0. \quad (65)$$

At equilibrium, we can therefore obtain the following expression for the logarithmic equilibrium constants:

$$\ln K_i(T, P, n^v, n^l) = \ln \phi_i^l(T, P, n^l) - \ln \phi_i^v(T, P, n^v). \quad (66)$$

6.1.1 Ideal vapor-liquid mixture

For an ideal vapor-liquid mixture, the logarithmic equilibrium constants are equal to the logarithmic fugacity coefficients of the ideal liquid phase:

$$\begin{aligned} \ln K_i^{id}(T, P, n^v, n^l) &= \ln \phi_i^{id}(T, P, n^l) - \ln \phi_i^{ig}(T, P, n^v) \\ &= \ln \phi_i^{id}(T, P, n^l) \\ &= \ln \frac{P_i^{sat}(T)}{P} + \frac{v_i^l(T)(P - P_i^{sat}(T))}{RT} \end{aligned} \quad (67)$$

The equilibrium constants are therefore independent of the composition vectors, i.e. $K_i^{id}(T, P, n^v, n^l) = K_i^{id}(T, P)$. That can be exploited in the ideal vapor-liquid equilibrium computations.

6.2 Summary

The equilibrium constants for ideal and nonideal vapor-liquid mixtures are

$$\ln K_i^{id}(T, P) = \ln \frac{P_i^{sat}(T)}{P} + \frac{v_i^l(T)(P - P_i^{sat}(T))}{RT}, \quad (68a)$$

$$\ln K_i(T, P, n^v, n^l) = \ln \phi_i^l(T, P, n^l) - \ln \phi_i^v(T, P, n^v). \quad (68b)$$

7 Vapor-liquid equilibrium of ideal mixtures

In this section, we exploit the fact that the equilibrium constants of ideal vapor-liquid mixtures, $K_i^{id}(T, P)$, are independent of composition. We can therefore solve the vapor-liquid equilibrium by 1) evaluating the equilibrium constants at the specified temperature, T , and pressure, P , 2) solve for the vapor fraction, and 3) compute the vapor-liquid composition (in moles) from the equilibrium constants and the vapor fraction. The ideal vapor-liquid equilibrium constants are

$$K_i^{id}(T, P) = \exp(\ln K_i^{id}(T, P)), \quad (69a)$$

$$\ln K_i^{id}(T, P) = \ln \frac{P_i^{sat}(T)}{P} + \frac{v_i^l(T)(P - P_i^{sat}(T))}{RT}. \quad (69b)$$

7. Vapor-liquid equilibrium of ideal mixtures

We express the vapor mole fractions, y_i , in terms of the equilibrium constants, $K_i^{id}(T, P)$, and the liquid mole fractions, x_i :

$$y_i = K_i^{id}(T, P)x_i. \quad (70)$$

The total moles in the vapor and liquid phases are $N^v = \sum_i n_i^v$ and $N^l = \sum_i n_i^l$. The total moles in the mixture is $N = N^v + N^l$. The vapor fraction is $\beta = N^v/N$ and $1 - \beta = 1 - N^v/N = (N - N^v)/N = N^l/N$. The total mole fractions are $z_i = n_i/N$. We derive the expressions for x_i from the mass balance:

$$n_i^v + n_i^l = n_i, \quad (71a)$$

$$\frac{n_i^v}{N} + \frac{n_i^l}{N} = \frac{n_i}{N}, \quad (71b)$$

$$\frac{N^v}{N} \frac{n_i^v}{N^v} + \frac{N^l}{N} \frac{n_i^l}{N^l} = \frac{n_i}{N}, \quad (71c)$$

$$\beta y_i + (1 - \beta)x_i = z_i, \quad (71d)$$

$$\beta K_i^{id}(T, P)x_i + (1 - \beta)x_i = z_i, \quad (71e)$$

$$(1 - \beta + \beta K_i^{id}(T, P))x_i = z_i. \quad (71f)$$

The liquid mole fraction, x_i , is therefore

$$x_i = \frac{z_i}{1 + \beta(K_i^{id}(T, P) - 1)}. \quad (72)$$

Because we solve for β , y and x will not sum to one during the solution procedure. We therefore require that both y and x sum to one. When $0 < \beta \leq 1$, the condition that

$$\sum_i x_i = 1, \quad (73)$$

together with $\sum_i z_i = 1$ implies that

$$\sum_i y_i = 1. \quad (74)$$

We omit the derivation. It is therefore sufficient to only require that $\sum_i x_i = 1$ for $0 < \beta \leq 1$. $\beta = 0$ implies that $x_i = z_i$ such that $\sum_i x_i = 1$ is satisfied independent of the equilibrium ratios, $K_i^{id}(T, P)$. In that case, we therefore require that

$$\sum_i y_i = 1. \quad (75)$$

The following condition is equivalent to $\sum_i x_i = 1$ for $0 < \beta \leq 1$ and to $\sum_i y_i = 1$ for $\beta = 0$:

$$\sum_i (y_i - x_i) = 0. \quad (76)$$

That is, $\sum_i y_i = 1$ and $\sum_i x_i = 1$ clearly imply that $\sum_i (y_i - x_i) = 0$. Because of the way that we compute y_i and x_i , $\sum_i (y_i - x_i) = 0$ also implies that $\sum_i y_i = 1$ and $\sum_i x_i = 1$. We insert the expressions

for y_i and x_i to obtain an equation in β :

$$\begin{aligned}\sum_i (y_i - x_i) &= \sum_i (K_i^{id}(T, P) - 1) x_i \\ &= \sum_i \frac{K_i^{id}(T, P) - 1}{1 + \beta(K_i^{id}(T, P) - 1)} z_i = 0,\end{aligned}\quad (77)$$

We therefore solve the ideal vapor-liquid equilibrium problem by solving the scalar nonlinear equation,

$$f(\beta) = \sum_i \frac{K_i^{id}(T, P) - 1}{1 + \beta(K_i^{id}(T, P) - 1)} z_i = 0, \quad (78a)$$

$$f'(\beta) = - \sum_i \left(\frac{K_i^{id}(T, P) - 1}{1 + \beta(K_i^{id}(T, P) - 1)} \right)^2 z_i. \quad (78b)$$

$f(\beta) = 0$ is called the Rachford-Rice equation. Once we have solved the Rachford-Rice equation for β , we compute the vapor-liquid composition (in moles) by

$$\begin{aligned}n_i^v &= \frac{n_i^v}{N^v} \frac{N^v}{N} N \\ &= y_i \beta N \\ &= \frac{\beta K_i^{id}(T, P)}{1 + \beta(K_i^{id}(T, P) - 1)} n_i.\end{aligned}\quad (79)$$

Next, we compute the liquid mole numbers by $n^l = n - n^v$.

7.1 Solution of the Rachford-Rice equation

We solve the Rachford-Rice equation with Newton's method:

$$\beta_{k+1} = \beta_k - \frac{f(\beta_k)}{f'(\beta_k)}. \quad (80)$$

7.2 Summary

We compute the ideal vapor-liquid equilibrium constants with

$$K_i^{id}(T, P) = \exp(\ln K_i^{id}(T, P)), \quad (81a)$$

$$\ln K_i^{id}(T, P) = \ln \frac{P_i^{sat}(T)}{P} + \frac{v_i^l(T)(P - P_i^{sat}(T))}{RT}. \quad (81b)$$

Next, we solve the Rachford-Rice equation,

$$f(\beta) = \sum_i \frac{K_i^{id}(T, P) - 1}{1 + \beta(K_i^{id}(T, P) - 1)} z_i = 0, \quad (82)$$

for β using Newton's method,

$$\beta_{k+1} = \beta_k - \frac{f(\beta_k)}{f'(\beta_k)}, \quad (83)$$

where the derivative is

$$f'(\beta) = - \sum_i \left(\frac{K_i^{id}(T, P) - 1}{1 + \beta(K_i^{id}(T, P) - 1)} \right)^2 z_i. \quad (84)$$

We compute the vapor-liquid composition (in moles) by

$$n_i^v = \frac{\beta K_i^{id}(T, P)}{1 + \beta(K_i^{id}(T, P) - 1)} n_i, \quad (85a)$$

$$n_i^l = n_i - n_i^v. \quad (85b)$$

8 Computation of phase envelopes

In this section, we describe the equations that we solve in order to construct the phase envelope, and we discuss how to solve them. The phase envelope consists of two curves; the bubble-point curve where $\beta = 0$ and the dew-point curve where $\beta = 1$. We will consider the more general case of computing isocurves where $\beta \in [0, 1]$ is constant. In the method that we present in this section, the logarithmic equilibrium constants, $\ln K_i$, the logarithmic temperature, $\ln T$, and the logarithmic pressure, $\ln P$, are the independent variables, i.e. the variables that we solve for. The vapor mole fractions, y_i , and the liquid mole fractions, x_i , are dependent variables that are functions of the vapor fraction, β , the equilibrium constants, K_i , and the total mole fractions, z_i . The vapor fraction, β , and the vector of total mole fractions, z , are parameters in the problem. We introduce $N_C + 2$ equations that define one point on the isocurve. N_C is the number of components in the mixture. We solve these equations repeatedly in a sequential manner in order to construct the isocurve. The first N_C equations are the vapor-liquid equilibrium conditions formulated in terms of the equilibrium constants, i.e. (65). We repeat the equilibrium conditions here:

$$\ln K_i + \ln \phi_i^v(T, P, y) - \ln \phi_i^l(T, P, x) = 0. \quad (86)$$

The mole numbers do not appear explicitly in the expression for the fugacity coefficients. They only appear implicitly through the mole fractions. We can therefore write $\ln \phi_i^v(T, P, n^v) = \ln \phi_i^v(T, P, y(n^v)) = \ln \phi_i^v(T, P, y)$ where $y_i = n_i^v / (\sum_j n_j^v)$ and similarly for the liquid fugacity coefficients. That is why y and x appear in place of n^v and n^l in the above equilibrium conditions. However, because y and x are dependent variables, they will not necessarily sum to one during the solution of the equations. We therefore treat them as mole numbers when we evaluate the logarithmic fugacity coefficients. That is, we evaluate the vapor phase logarithmic fugacity coefficients, $\ln \phi_i^v(T, P, y)$, as $\ln \phi_i^v(T, P, n^v)$ where $n^v = y$, and similarly for the liquid fugacity coefficients. Also, because y_i and x_i are functions of K_i , we cannot explicitly isolate K_i in the vapor-liquid equilibrium equations above. We compute y_i and x_i in the same way that we did for the ideal vapor-liquid equilibrium problem:

$$x_i = \frac{z_i}{1 + \beta(K_i - 1)}, \quad (87a)$$

$$y_i = K_i x_i. \quad (87b)$$

However, here we do not use the ideal vapor-liquid equilibrium constants, and the equilibrium constants are independent variables. The next equation is the Rachford-Rice equation (78a). However, we implement it in the form,

$$\sum_i (y_i - x_i) = 0, \quad (88)$$

where we do not substitute the expressions for the vapor-liquid mole fractions. That is because we need to compute y_i and x_i anyway, in order to evaluate the logarithmic fugacity coefficients. The Rachford-Rice equation ensures that $\sum_i y_i = 1$ and $\sum_i x_i = 1$ are both satisfied². In order to define the last equation, we collect the independent variables in a vector,

$$X = [\ln K; \ln T; \ln P] \in \mathbb{R}^{N_C+2}, \quad (89)$$

where $\ln K = [\ln K_1; \dots; \ln K_{N_C}]$. The last isocurve equation states that one of the independent variables should be specified, i.e. the s 'th component of X should have the value S :

$$X_s - S = 0. \quad (90)$$

That means that we specify either temperature, pressure, or one of the equilibrium constants in order to compute a point on the isocurve. Michelsen and Mollerup (2007, Chap. 12) discuss different strategies for selecting the specified variable. The isocurve is constructed by solving the isocurve equations for a sufficient number of values of S . We write the isocurve equations compactly as

$$F(X; S) = 0, \quad (91)$$

where

$$F_i = \ln K_i + \ln \phi_i^v(T, P, y) - \ln \phi_i^l(T, P, x), \quad i = 1, \dots, N_C, \quad (92a)$$

$$F_{N_C+1} = \sum_i (y_i - x_i), \quad (92b)$$

$$F_{N_C+2} = X_s - S. \quad (92c)$$

8.1 Solution of the isocurve equations

As mentioned, we construct the isocurve by solving the isocurve equations for a number of specified values of the s 'th variable. For the m 'th value of the specified variable, S^m , we solve the isocurve equations, $F(X^m; S^m) = 0$, with Newton's method:

$$X^{m,k+1} = X^{m,k} - \left(\frac{\partial F}{\partial X} \right)^{-1} F(X^{m,k}; S^m). \quad (93)$$

8.2 Computation of initial guess

When we have solved the isocurve equations for one value of the specified variable, we want to compute the sensitivities of the solution in order to compute an initial guess for the subsequent

²It is not immediately obvious that this is true. It is because of the way that we compute y_i and x_i .

Newton iterations. We differentiate the equation $F(X; S) = F(X(S); S) = 0$ with respect to S :

$$\frac{\partial F}{\partial X} \frac{\partial X}{\partial S} + \frac{\partial F}{\partial S} = 0. \quad (94)$$

We isolate the sensitivities of X with respect to S :

$$\frac{\partial X}{\partial S} = - \left(\frac{\partial F}{\partial X} \right)^{-1} \frac{\partial F}{\partial S}. \quad (95)$$

Once we have solved $F(X^m; S^m) = 0$ for X^m , we can compute an initial guess, $X^{m+1,0}$, for the solution of the isocurve equations for the next value of the specified variable, $F(X^{m+1}; S^{m+1}) = 0$:

$$X^{m+1,0} = X^m + \frac{\partial X^m}{\partial S^m} (S^{m+1} - S^m). \quad (96)$$

8.3 Summary

The isocurve equations (or phase envelope equations if $\beta = 0$ or $\beta = 1$) are

$$F(X; S) = 0, \quad (97)$$

where the independent variables are $X = [\ln K; \ln T; \ln P]$. The equations are

$$F_i = \ln K_i + \ln \phi_i^v(T, P, y) - \ln \phi_i^l(T, P, x), \quad i = 1, \dots, N_C, \quad (98a)$$

$$F_{N_C+1} = \sum_i (y_i - x_i), \quad (98b)$$

$$F_{N_C+2} = X_s - S, \quad (98c)$$

where the vapor-liquid mole fractions are dependent variables:

$$x_i = \frac{z_i}{1 + \beta(K_i - 1)}, \quad (99a)$$

$$y_i = K_i x_i. \quad (99b)$$

We solve the equations with Newton's method:

$$X^{m,k+1} = X^{m,k} - \left(\frac{\partial F}{\partial X} \right)^{-1} F(X^{m,k}; S^m). \quad (100)$$

We compute initial guesses for the Newton iterations with

$$X^{m+1,0} = X^m + \frac{\partial X^m}{\partial S^m} (S^{m+1} - S^m), \quad (101)$$

where the sensitivities are

$$\frac{\partial X}{\partial S} = - \left(\frac{\partial F}{\partial X} \right)^{-1} \frac{\partial F}{\partial S}. \quad (102)$$

9 Derivatives

In this section, we provide the derivatives that are necessary to solve the vapor-liquid equilibrium problem (for both ideal and nonideal mixtures) and to compute the isocurves (i.e. curves where β is constant). In order to do so, we also provide the derivatives of the logarithmic fugacity coefficients.

9.1 Logarithmic fugacity coefficients

We provide the derivatives of the logarithmic fugacity coefficients for ideal liquids and nonideal mixtures. The logarithmic fugacity coefficients of ideal gases are zero, and the derivatives are therefore also zero.

9.1.1 Ideal liquid mixture

The logarithmic fugacity coefficients of an ideal liquid mixture are

$$\ln \phi_i^{id}(T, P) = \ln \frac{P_i^{sat}(T)}{P} + \frac{v_i^l(T)(P - P_i^{sat}(T))}{RT}. \quad (103)$$

The liquid volume is given by the DIPPR correlation,

$$v_i^l = \frac{B_i^{1+(1-\frac{T}{C_i})^{D_i}}}{A_i}. \quad (104)$$

The saturation pressure is given by the DIPPR correlation,

$$P_i^{sat} = \exp(\ln P_i^{sat}), \quad (105a)$$

$$\ln P_i^{sat} = A_i + \frac{B_i}{T} + C_i \ln(T) + D_i T^{E_i}. \quad (105b)$$

Note that the parameters, A_i , B_i , C_i , and D_i in the DIPPR correlation for the saturation pressure are not the same as those in the DIPPR correlation for the liquid volume. The derivatives of the logarithmic fugacity coefficients are

$$\frac{\partial \ln \phi_i^{id}}{\partial T} = \frac{\partial \ln P_i^{sat}}{\partial T} + \frac{1}{RT} \left(\left(\frac{\partial v_i^l}{\partial T} - \frac{v_i^l(T)}{T} \right) (P - P_i^{sat}(T)) - v_i^l(T) \frac{\partial P_i^{sat}}{\partial T} \right), \quad (106a)$$

$$\frac{\partial \ln \phi_i^{id}}{\partial P} = -\frac{1}{P} + \frac{v_i^l(T)}{RT}. \quad (106b)$$

The derivative of the liquid volume is

$$\frac{\partial v_i^l}{\partial T} = -\ln B_i \frac{D_i}{C_i} \left(1 - \frac{T}{C_i} \right)^{D_i-1} v_i^l. \quad (107)$$

The derivative of the saturation pressure is

$$\frac{\partial \ln P_i^{sat}}{\partial T} = \frac{1}{T} \left(C_i - \frac{B_i}{T} + D_i E_i T^{E_i} \right), \quad (108a)$$

$$\frac{\partial P_i^{sat}}{\partial T} = P_i^{sat}(T) \frac{\partial \ln P_i^{sat}}{\partial T}. \quad (108b)$$

9.1.2 Nonideal mixture

The logarithmic fugacity coefficients of nonideal mixtures have a complicated expression. We therefore introduce auxiliary functions such that we can write it as

$$\ln \phi_i = (Z - 1) \frac{b_i}{b_m} - g_z - \frac{1}{\epsilon - \sigma} g_{\phi,i} f. \quad (109)$$

The auxiliary functions are

$$f = \ln \left(\frac{Z + \epsilon B}{Z + \sigma B} \right), \quad (110a)$$

$$g_z = \ln(Z - B), \quad (110b)$$

$$g_{\phi,i} = \frac{1}{RT b_m} \left(2 \sum_{j=1}^{N_C} z_j a_{ij} - a_m \frac{b_i}{b_m} \right). \quad (110c)$$

The derivatives of the logarithmic fugacity coefficients are

$$\frac{\partial \ln \phi_i}{\partial T} = \frac{\partial Z}{\partial T} \frac{b_i}{b_m} - \frac{\partial g_z}{\partial T} - \frac{1}{\epsilon - \sigma} \left(\frac{\partial g_{\phi,i}}{\partial T} f + g_{\phi,i} \frac{\partial f}{\partial T} \right), \quad (111a)$$

$$\frac{\partial \ln \phi_i}{\partial P} = \frac{\partial Z}{\partial P} \frac{b_i}{b_m} - \frac{\partial g_z}{\partial P} - \frac{1}{\epsilon - \sigma} g_{\phi,i} \frac{\partial f}{\partial P}, \quad (111b)$$

$$\frac{\partial \ln \phi_i}{\partial n_k} = \frac{\partial Z}{\partial n_k} \frac{b_i}{b_m} - (Z - 1) \frac{b_i}{b_m^2} \frac{\partial b_m}{\partial n_k} - \frac{\partial g_z}{\partial n_k} - \frac{1}{(\epsilon - \sigma)} \left(\frac{\partial g_{\phi,i}}{\partial n_k} f + g_{\phi,i} \frac{\partial f}{\partial n_k} \right). \quad (111c)$$

The derivatives of the auxiliary function f are

$$\frac{\partial f}{\partial T} = \frac{\partial f}{\partial Z} \frac{\partial Z}{\partial T} + \frac{\partial f}{\partial B} \frac{\partial B}{\partial T}, \quad (112a)$$

$$\frac{\partial f}{\partial P} = \frac{\partial f}{\partial Z} \frac{\partial Z}{\partial P} + \frac{\partial f}{\partial B} \frac{\partial B}{\partial P}, \quad (112b)$$

$$\frac{\partial f}{\partial n_k} = \frac{\partial f}{\partial Z} \frac{\partial Z}{\partial n_k} + \frac{\partial f}{\partial B} \frac{\partial B}{\partial n_k}, \quad (112c)$$

where

$$\frac{\partial f}{\partial Z} = \frac{1}{Z + \epsilon B} - \frac{1}{Z + \sigma B}, \quad (113a)$$

$$\frac{\partial f}{\partial B} = \frac{\epsilon}{Z + \epsilon B} - \frac{\sigma}{Z + \sigma B}. \quad (113b)$$

9. Derivatives

The derivatives of the auxiliary function g_z are

$$\frac{\partial g_z}{\partial T} = \frac{\partial g_z}{\partial Z} \frac{\partial Z}{\partial T} + \frac{\partial g_z}{\partial B} \frac{\partial B}{\partial T}, \quad (114a)$$

$$\frac{\partial g_z}{\partial P} = \frac{\partial g_z}{\partial Z} \frac{\partial Z}{\partial P} + \frac{\partial g_z}{\partial B} \frac{\partial B}{\partial P}, \quad (114b)$$

$$\frac{\partial g_z}{\partial n_k} = \frac{\partial g_z}{\partial Z} \frac{\partial Z}{\partial n_k} + \frac{\partial g_z}{\partial B} \frac{\partial B}{\partial n_k}, \quad (114c)$$

where

$$\frac{\partial g_z}{\partial Z} = \frac{1}{Z - B}, \quad (115a)$$

$$\frac{\partial g_z}{\partial B} = -\frac{1}{Z - B}. \quad (115b)$$

The derivatives of the auxiliary function $g_{\phi,i}$ are

$$\frac{\partial g_{\phi,i}}{\partial T} = \frac{1}{T} \left(\frac{1}{Rb_m} \left(2 \sum_{j=1}^{N_C} z_j \frac{\partial a_{ij}}{\partial T} - \frac{\partial a_m}{\partial T} \frac{b_i}{b_m} \right) - g_{\phi,i} \right), \quad (116a)$$

$$\frac{\partial g_{\phi,i}}{\partial n_k} = \frac{1}{b_m} \left[\left(\frac{2}{N} \left(a_{ik} - \sum_{j=1}^{N_C} x_j a_{ij} \right) - \frac{b_i}{b_m} \left(\frac{\partial a_m}{\partial n_k} - \frac{1}{b_m} \frac{\partial b_m}{\partial n_k} \right) \right) - \frac{\partial b_m}{\partial n_k} g_{\phi,i} + \frac{1}{RT} \right]. \quad (116b)$$

The compressibility factor, Z , satisfies the cubic polynomial $q(Z; T, P, n) = 0$. We obtain the derivatives of the compressibility factor with the inverse function theorem:

$$\frac{\partial Z}{\partial T} = - \left(\frac{\partial q}{\partial Z} \right)^{-1} \frac{\partial q}{\partial T}, \quad (117a)$$

$$\frac{\partial Z}{\partial P} = - \left(\frac{\partial q}{\partial Z} \right)^{-1} \frac{\partial q}{\partial P}, \quad (117b)$$

$$\frac{\partial Z}{\partial n_k} = - \left(\frac{\partial q}{\partial Z} \right)^{-1} \frac{\partial q}{\partial n_k}. \quad (117c)$$

The derivatives of the polynomial are

$$\frac{\partial q}{\partial Z} = 3Z^2 + \sum_{m=1}^2 m d_m Z^{m-1} \quad (118a)$$

$$\frac{\partial q}{\partial T} = \sum_{m=0}^2 \frac{\partial d_m}{\partial T} Z^m, \quad (118b)$$

$$\frac{\partial q}{\partial P} = \sum_{m=0}^2 \frac{\partial d_m}{\partial P} Z^m, \quad (118c)$$

$$\frac{\partial q}{\partial n_k} = \sum_{m=0}^2 \frac{\partial d_m}{\partial n_k} Z^m. \quad (118d)$$

9. Derivatives

The derivatives of the polynomial coefficients are

$$\frac{\partial d_m}{\partial T} = \frac{\partial d_m}{\partial A} \frac{\partial A}{\partial T} + \frac{\partial d_m}{\partial B} \frac{\partial B}{\partial T}, \quad (119a)$$

$$\frac{\partial d_m}{\partial P} = \frac{\partial d_m}{\partial A} \frac{\partial A}{\partial P} + \frac{\partial d_m}{\partial B} \frac{\partial B}{\partial P}, \quad (119b)$$

$$\frac{\partial d_m}{\partial n_k} = \frac{\partial d_m}{\partial A} \frac{\partial A}{\partial n_k} + \frac{\partial d_m}{\partial B} \frac{\partial B}{\partial n_k}, \quad (119c)$$

where

$$\frac{\partial d_2}{\partial A} = 0, \quad (120a)$$

$$\frac{\partial d_2}{\partial B} = (\epsilon + \sigma - 1), \quad (120b)$$

$$\frac{\partial d_1}{\partial A} = 1, \quad (120c)$$

$$\frac{\partial d_1}{\partial B} = -(\epsilon + \sigma) + 2(\epsilon\sigma - \epsilon - \sigma)B, \quad (120d)$$

$$\frac{\partial d_0}{\partial A} = -B, \quad (120e)$$

$$\frac{\partial d_0}{\partial B} = -(A + \epsilon\sigma(2B + 3B^2)). \quad (120f)$$

The derivatives of A are

$$\frac{\partial A}{\partial T} = \frac{\partial a_m}{\partial T} \frac{P}{R^2 T^2} - \frac{2}{T} A, \quad (121a)$$

$$\frac{\partial A}{\partial P} = \frac{a_m}{R^2 T^2}, \quad (121b)$$

$$\frac{\partial A}{\partial n_k} = \frac{\partial a_m}{\partial n_k} \frac{P}{R^2 T^2}. \quad (121c)$$

The derivatives of B are

$$\frac{\partial B}{\partial T} = -\frac{b_m P}{RT^2}, \quad (122a)$$

$$\frac{\partial B}{\partial P} = \frac{b_m}{RT}, \quad (122b)$$

$$\frac{\partial B}{\partial n_k} = \frac{\partial b_m}{\partial n_k} \frac{P}{RT}. \quad (122c)$$

The derivatives of the van der Waals' mixing parameter a_m are

$$\frac{\partial a_m}{\partial T} = \sum_{i=1}^{N_C} \sum_{j=1}^{N_C} z_i z_j \frac{\partial a_{ij}}{\partial T}, \quad (123a)$$

$$\frac{\partial a_m}{\partial n_k} = \frac{2}{N} \left(\sum_{i=1}^{N_C} z_i a_{ik} - a_m \right). \quad (123b)$$

The derivative of the van der Waals' mixing parameter b_m is

$$\frac{\partial b_m}{\partial n_k} = \frac{b_k - b_m}{N}. \quad (124)$$

The derivative of a_{ij} is

$$\frac{\partial a_{ij}}{\partial T} = \frac{1 - k_{ij}}{2\sqrt{\hat{a}_{ij}}} \frac{\partial \hat{a}_{ij}}{\partial T}, \quad (125)$$

where the derivative of \hat{a}_{ij} are

$$\frac{\partial \hat{a}_{ij}}{\partial T} = \frac{\partial a_i}{\partial T} a_j + a_i \frac{\partial a_j}{\partial T}. \quad (126)$$

The derivative of the pure component parameter a_i is

$$\frac{\partial a_i}{\partial T} = \frac{\partial \alpha}{\partial T} \Psi \frac{R^2 T_{c,i}^2}{P_{c,i}}. \quad (127)$$

The derivative of the function α is

$$\frac{\partial \alpha}{\partial T} = -\alpha \frac{m(\omega_i)}{\sqrt{\alpha T T_{c,i}}}. \quad (128)$$

9.2 Vapor-liquid equilibrium equations

The vapor-liquid equilibrium conditions are

$$g_i(T, P, n^v) = \ln \phi_i^v(T, P, n^v) + \ln y_i - \ln \phi_i^l(T, P, n^l) + \ln x_i, \quad (129)$$

where the liquid mole numbers are functions of the vapor mole numbers, i.e. $n^l = n - n^v$. The derivatives of $g_i(T, P, n^v)$ are

$$\frac{\partial g_i}{\partial n_k^v} = \frac{\partial \ln \phi_i^v}{\partial n_k^v} + \frac{\partial \ln y_i}{\partial n_k^v} + \frac{\partial \ln \phi_i^l}{\partial n_k^l} + \frac{\partial \ln x_i}{\partial n_k^l}. \quad (130)$$

The derivatives of the logarithmic mole fractions are

$$\frac{\partial \ln y_i}{\partial n_k^v} = \frac{\delta_{ik}}{n_k^v} - \frac{1}{N^v}, \quad (131a)$$

$$\frac{\partial \ln x_i}{\partial n_k^l} = \frac{\delta_{ik}}{n_k^l} - \frac{1}{N^l}. \quad (131b)$$

δ_{ik} is one if $i = k$ and zero if $i \neq k$.

9.3 Ideal vapor-liquid equilibrium equations

For ideal vapor-liquid mixtures, we solve the Rachford-Rice equation:

$$f(\beta) = \sum_i \frac{K_i^{id}(T, P) - 1}{1 + \beta(K_i^{id}(T, P) - 1)} z_i = 0. \quad (132)$$

The derivative of the $f(\beta)$ with respect to the vapor fraction is

$$f'(\beta) = - \sum_i \left(\frac{K_i^{id}(T, P) - 1}{1 + \beta(K_i^{id}(T, P) - 1)} \right)^2 z_i. \quad (133)$$

9.4 Phase envelope equations

The independent variables in the phase envelope computations are

$$X = [\ln K; \ln T; \ln P]. \quad (134)$$

The isocurve equations (or phase envelope equations if $\beta = 0$ or $\beta = 1$) are

$$F_i = \ln K_i + \ln \phi_i^v(T, P, y) - \ln \phi_i^l(T, P, x), \quad i = 1, \dots, N_C, \quad (135a)$$

$$F_{N_C+1} = \sum_i (y_i - x_i), \quad (135b)$$

$$F_{N_C+2} = X_s - S. \quad (135c)$$

The vapor-liquid mole fractions are dependent variables given by

$$x_i = \frac{z_i}{1 - \beta + \beta K_i}, \quad (136a)$$

$$y_i = K_i x_i. \quad (136b)$$

The derivatives of the vapor-liquid mole fractions with respect to the equilibrium constants are

$$\frac{\partial x_i}{\partial K_k} = \begin{cases} \frac{-\beta x_i^2}{z_i} & i = k, \\ 0 & i \neq k, \end{cases} \quad (137a)$$

$$\frac{\partial y_i}{\partial K_k} = \begin{cases} x_i + K_i \frac{\partial x_i}{\partial K_k} & i = k, \\ 0 & i \neq k. \end{cases} \quad (137b)$$

The derivatives of the vapor-liquid equilibrium conditions are therefore

$$\frac{\partial F_i}{\partial \ln K_k} = \delta_{ik} + K_k \left(\frac{\partial \ln \phi_i^v}{\partial n_k^v}(T, P, y) \frac{\partial y_k}{\partial K_k} - \frac{\partial \ln \phi_i^l}{\partial n_k^l}(T, P, x) \frac{\partial x_k}{\partial K_k} \right), \quad i = 1, \dots, N_C, \quad (138a)$$

$$\frac{\partial F_i}{\partial \ln T} = T \left(\frac{\partial \ln \phi_i^v}{\partial T}(T, P, y) - \frac{\partial \ln \phi_i^l}{\partial T}(T, P, x) \right), \quad i = 1, \dots, N_C, \quad (138b)$$

$$\frac{\partial F_i}{\partial \ln P} = P \left(\frac{\partial \ln \phi_i^v}{\partial P}(T, P, y) - \frac{\partial \ln \phi_i^l}{\partial P}(T, P, x) \right), \quad i = 1, \dots, N_C. \quad (138c)$$

The derivative of the equation that corresponds to the Rachford-Rice equation are

$$\frac{\partial F_{N_C+1}}{\partial K_k} = K_k \left(\frac{\partial y_k}{\partial K_k} - \frac{\partial x_k}{\partial K_k} \right). \quad (139)$$

The derivatives of the last equation are

$$\frac{\partial F_{N_C+2}}{\partial X} = e_s, \quad (140)$$

where element s of e_s is 1 and all other elements are zero. The derivative of the last equation with respect to the value of the specified variable is

$$\frac{\partial F_{N_C+2}}{\partial S} = -1. \quad (141)$$

10 An algorithm for computing phase envelopes and an example program

In this section, we describe an algorithm for computing isocurves, i.e. curves where the vapor fraction, β , is constant. The algorithm is described by [Michelsen and Mollerup \(2007, Chap. 12\)](#) and also by [Michelsen \(1980\)](#). It features automatic selection of the specified variable and the step size, and it uses cubic interpolation to enhance initial guesses for the Newton iterations. The algorithm sequentially constructs the isocurve. It begins at a point where the equilibrium constants can be approximated and sequentially computes points on the isocurve until it reaches the critical point.

10.1 Wilson's approximation of equilibrium constants

We approximate the equilibrium constants at the first point of the isocurve with the approximation by [Wilson \(1969\)](#):

$$K_i(T, P) = \frac{P_{c,i}}{P} \exp \left(5.373(1 + \omega_i) \left(1 - \frac{T_{c,i}}{T} \right) \right). \quad (142)$$

$T_{c,i}$, $P_{c,i}$, and ω_i are the critical temperature, critical pressure, and the acentric factor. We use values of those parameters from the DIPPR database. We could potentially also use the ideal vapor-liquid equilibrium constants that we described in [Section 6](#).

10.2 Cubic interpolation

It is essential to have good initial guesses for the Newton iterations. The algorithm therefore uses cubic interpolation to improve the estimates we described in [Section 8](#). When we have computed two points on the isocurve and their sensitivities, we can determine a cubic polynomial for each of the variables that intersects these points. The polynomial is

$$p_i(S) = \sum_j a_{ji} S^j. \quad (143)$$

The superscript j on S is a power, and a_{ji} are the polynomial coefficients. Once we have solved for the m 'th point on the isocurve, we interpolate the solution curves. We require that the polynomial and its derivatives intersect X^m and X^{m-1} and their derivatives with respect to S^m and S^{m-1} :

$$p_i(S^m) = X_i^m, \quad (144a)$$

$$p'_i(S^m) = \frac{\partial X_i^m}{\partial S^m}, \quad (144b)$$

$$p_i(S^{m-1}) = X_i^{m-1}, \quad (144c)$$

$$p'_i(S^{m-1}) = \frac{\partial X_i^{m-1}}{\partial S^{m-1}}. \quad (144d)$$

We write the above equations as a linear system of equations

$$Ma = b. \quad (145)$$

The system matrix, M , is a 4×4 matrix that contains powers of S^m and S^{m-1} :

$$M = \begin{bmatrix} 1 & S^m & (S^m)^2 & (S^m)^3 \\ 0 & 1 & 2S^m & 3(S^m)^2 \\ 1 & S^{m-1} & (S^{m-1})^2 & (S^{m-1})^3 \\ 0 & 1 & 2S^{m-1} & 3(S^{m-1})^2 \end{bmatrix} \quad (146)$$

The matrix a is $4 \times N_C + 2$ and simply contains the polynomial coefficients a_{ji} (row j , column i). $N_C + 2$ is the number of elements in the vector X . The right-hand side, b , is also a $4 \times N_C + 2$ matrix and each column contains the right-hand sides of (144):

$$b_{1i} = X_i^m, \quad (147a)$$

$$b_{2i} = \frac{\partial X_i^m}{\partial S^m}, \quad (147b)$$

$$b_{3i} = X_i^{m-1}, \quad (147c)$$

$$b_{4i} = \frac{\partial X_i^{m-1}}{\partial S^{m-1}}. \quad (147d)$$

Once we have solved the linear system (145) for the polynomial coefficients, we compute the initial estimate, $X^{m+1,0}$, for the subsequent Newton iterations:

$$X_i^{m+1,0} = p_i(S^{m+1}). \quad (148)$$

The matrix M can become ill-conditioned if the difference between S^m and S^{m-1} is small, i.e. if the step is small³. In that case, it can be useful to revert to the linear approximation:

$$X^{m+1,0} = X^m + \frac{\partial X^m}{\partial S^m}(S^{m+1} - S^m). \quad (149)$$

³It is known that the monomial basis can lead to ill-conditioned system matrices in interpolation. It might be possible to remedy this by choosing another basis, e.g. the Lagrange polynomials.

We use the above linear approximation when the condition number of M is larger than 10^6 . We also use the linear approximation when we do not have two points on the isocurve, e.g. when $m = 1$.

10.3 Automatic selection of specified variable

In each iteration, we choose the specified variable in subsequent iterations to be the variable whose sensitivity with respect to the value of the specified variable has the largest absolute value, i.e. we choose

$$\bar{s} = \arg \min_s \frac{\partial X_s^m}{\partial S^m} \quad (150)$$

In order to avoid rapid switching between variables, we only switch variables (i.e. set $s = \bar{s}$) if $\partial X_{\bar{s}}^m / \partial S^m > 1.1 \partial X_s^m / \partial S^m$. That is, we only switch the specified variable if the sensitivity is 1.1 times larger than the sensitivity of the current specified variable.

10.4 Automatic step size selection

We use a number of heuristics to choose the step size. First, we specify a target number of Newton iterations, usually 3 or 4. If the Newton iterations require more iterations to converge, we half the step size. If they require less, we double the step size. We also half the step size if the specified variable has changed. In order to ensure that we move in the right direction when we switch variables, we multiply the step size with the sign of the difference between the specified variable in the previous two iterations. That is, if the specified variable is X_s , then we use the sign of $X_s^m - X_s^{m-1}$ to decide whether S should increase or decrease.

10.5 Algorithm for computing isocurves

Algorithm 1 provides an overview of the algorithm that we implement in Section 10.6.

10.6 Example program

We now describe a program that implements Algorithm 1. The program consists of a script and a number of functions. We use Matlab routines from ThermoLib to compute the logarithmic fugacity coefficients. ThermoLib is an open-source thermodynamic library created by Ritschel et al. (2017, 2016) and Gaspar et al. (2017). It is available at www.psetools.org. It implements the expressions for the logarithmic fugacity coefficients that we described in Section 4 and 9.1.2. The names of the example script and the functions are

- CreatePhaseEnvelope.m
Example script
- ComputeWilsonKFactors.m
Evaluates approximate equilibrium constants
- ComputeIsocurve.m
Constructs an isocurve

Algorithm 1: An algorithm for computing isocurves.

Input: T, P, β, z, s
Output: $N, \{X^m\}_{m=1}^N$

```

1 Set  $m = 1$ ;
2 Set  $\Delta S^m = \Delta S^{\max}$ ;
3 while the critical point has not been reached do
4   Solve the isocurve equations  $F(X^m; S^m) = 0$  for  $X^m$ ;
5   Check whether  $X^m$  is close to the critical point, i.e. if  $|\ln K_i| < \epsilon$  for all  $i$ ;
6   Compute the sensitivities  $\partial X^m / \partial S^m$ ;
7   Set  $\bar{s} = \arg \min_s \partial X_s^m / \partial S^m$ ;
8   if  $\partial X_{\bar{s}}^m / \partial S^m > 1.1 \partial X_s^m / \partial S^m$  and  $s \neq \bar{s}$  then
9     | Set  $\Delta S^m = 0.5 \Delta S^m$ ;
10    | Set  $s = \bar{s}$ ;
11  end
12  if the number of Newton iterations exceeds the target number of iterations then
13    | Set  $\Delta S^m = 0.5 \Delta S^m$ ;
14  else if the number of Newton iterations is below the target number of iterations then
15    | Set  $\Delta S^m = 2 \Delta S^m$ ;
16  end
17  Set  $\Delta S^m = \min(\Delta S^m, \Delta S^{\max})$ ;
18  Compute the new specified variable,  $S^{m+1} = X_s^m + \text{sgn}(X_s^m - X_s^{m-1}) \Delta S^m$ ;
19  if  $m \geq 2$  and  $s$  has not changed in this iteration and  $\text{cond}(M) < 10^6$  then
20    | Compute  $X^{m+1,0}$  with cubic interpolation;
21  else
22    | Compute  $X^{m+1,0}$  with the linear approximation;
23  end
24  Increment  $m$  by 1;
25 end

```


10. An algorithm for computing phase envelopes and an example program

- IsocurveEquations.m
Evaluates the isocurve equations (and derivatives) described in Section 8
- CreateFlashOpts.m
Default options for the computations

Listing 1 shows the first part of the example script, CreatePhaseEnvelope.m. First, the script runs another script, LoadLibrary.m, from ThermoLib. LoadLibrary adds all ThermoLib routines to the path such that we can run them. Next, we select the Peng-Robinson equation of state, and we manually specify the binary interaction parameters, k_{ij} . We use LoadParams from ThermoLib to construct a vector of parameters that we need to pass on to other ThermoLib routines. The first argument to LoadParams is a vector of indices that specifies the components in the mixture. 1, 2, 3, and 7 refer to methane, ethane, propane, and n-heptane, respectively. 363 is carbon-dioxide (CO₂). We will use the variable NC (number of components) later in the script. Finally, we specify the composition which is 60% methane, 8% ethane, 5% propane, 25% n-heptane, and 2% CO₂.

Listing 1: CreatePhaseEnvelope.m – Initialization

```
1 %% Create phase diagram
2 clc; clear all; close all;
3
4 % Add ThermoLib
5 run('C:\Users\Tobia\Dropbox\3 PhD Nonlinear Model Predictive Control for Oil
      ↪ Reservoirs\Projects\ThermodynamicLibrary\ComputeThermoLib\matlab\
      ↪ LoadLibrary');
6
7 %% Load parameters
8 % Equation of state
9 EoS = 'PR'; % Peng-Robinson
10
11 % Binary interaction parameters
12 kij = [
13     0         0         0         0         0.1200
14     0         0         0         0         0.1500
15     0         0         0         0         0.1500
16     0         0         0         0         0.1500
17     0.1200    0.1500    0.1500    0.1500    0    ];
18
19 % Load DIPPR parameters
20 params = LoadParams([1:3, 7, 363], EoS, kij);
21
22 % Number of components
23 NC = params(5);
24
25 % Composition
26 z = [60; 8; 5; 25; 2]./100;
```

Listing 2 shows the remaining part of CreatePhaseEnvelope.m. This remaining part computes the bubble-point curve ($\beta = 0$) and the dew-point curve ($\beta = 1$). First, we use CreateFlashOpts.m to create a struct called opts that contains default settings for the computations. We discuss CreateFlashOpts.m later. Next, we select that the pressure of the first point on both isocurves is

specified. Pressure is variable number $N_C + 2$. We set the pressure of the first point on both the bubble-point and the dew-point curves to be 0.005 MPa. We use `ComputeIsocurve` to compute both the bubble-point curve and the dew-point curve. We use an initial guess of 88 K for the temperature of the first point on the bubble-point curve and 250 K for the dew-point curve. In the remaining part, we extract the temperature and pressure of the isocurves and plot them. Figure 1 shows the plot of the phase envelope. The phase envelope separates the two-phase region (inside) from the single-phase regions (outside). We see that the two-phase region extends up to around 18 MPa and to around 450 K. We also note that at low pressures, the phase envelope extends to very low temperatures, i.e. around 90 K. That means that it is necessary to bring the mixture to below 90 K in order to completely liquefy it. That is because the mixture contains a large amount of light gases. In particular, it contains 60% methane. The bubble-point and the dew-point curves meet at the critical point. It is marked with a black dot.

Listing 2: `CreatePhaseEnvelope.m` – Computation of the phase envelope

```

28 %% Compute phase envelope
29 % Create options structure
30 opts = CreateFlashOpts();
31
32 % The first specified variable is pressure
33 Idx = NC+2;
34
35 % Initial pressure
36 P0 = 0.005; % MPa
37
38 % Compute the bubble-point curve
39 T0 = 88;
40 b = 0;
41 BubblePointCurve = ComputeIsocurve(T0, P0, b, z, Idx, params, opts);
42
43 % Compute the dew-point curve
44 T0 = 250;
45 b = 1;
46 DewPointCurve = ComputeIsocurve(T0, P0, b, z, Idx, params, opts);
47
48 % Temperature and pressure
49 T = [exp(BubblePointCurve.X(NC+1, :)), exp(DewPointCurve.X(NC+1, end:-1:1))];
50 P = [exp(BubblePointCurve.X(NC+2, :)), exp(DewPointCurve.X(NC+2, end:-1:1))];
51
52 % Critical temperature and pressure (approximate)
53 Tc = exp(BubblePointCurve.X(NC+1, end));
54 Pc = exp(BubblePointCurve.X(NC+2, end));
55
56 % Create figure
57 figure(1);
58 plot(T, P, '-k', 'linewidth', 2); hold on;
59 plot(Tc, Pc, '.k', 'markersize', 20); hold off;
60 set(gca, 'fontsize', 14);
61 xlabel('Temperature [K]');
62 ylabel('Pressure [MPa]');

```

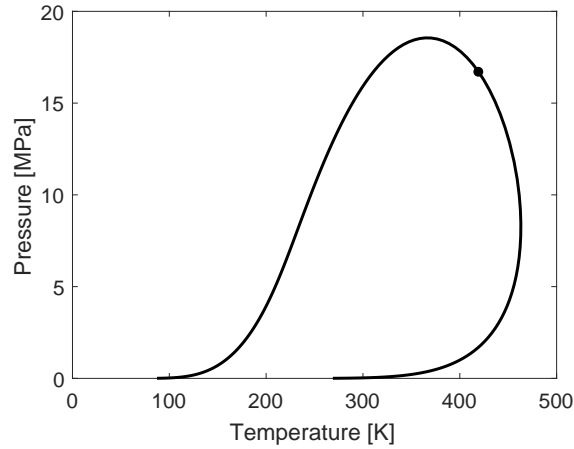


Figure 1: Phase envelope created with the example script CreatePhaseEnvelope.m.

Before we describe the ComputeIsocurve function, we show the ComputeWilsonKFactors function in Listing 3. It simply loads the relevant parameters and evaluates the equilibrium constants from (142). Note that the critical pressures are converted from Pa to MPa. That is required for the expression to be correct.

Listing 3: ComputeWilsonKFactors.m

```

1 function K = ComputeWilsonKFactors(T, P, params)
2
3 % Conversion factor
4 Pa2MPa = 1e-6; % MPa to Pa
5
6 % Extract relevant parameters
7 NoSinglePar = params(1);
8 NC          = params(5);
9 omega      = params(NoSinglePar+1 + 24*NC + (1:NC) + 1);
10 Tc         = params(NoSinglePar+1 + 25*NC + (1:NC) + 1);
11 Pc         = params(NoSinglePar+1 + 26*NC + (1:NC) + 1)*Pa2MPa;
12
13 % Compute Wilson K factors
14 K = Pc./P.*exp(5.373*(1 + omega).*(1 - Tc./T));

```

Next, we describe the ComputeIsocurve function. Listing 4 shows the interface of ComputeIsocurve, the loading of parameters from the struct created with CreateFlashOpts, and the computation of initial guesses for the equilibrium constants. The function requires specification of temperature, T, pressure, P, the specified vapor fraction, b, the total composition, z, the index of the specified variable at the first point, Idx, the ThermoLib parameters, params, and finally, the options, opts.

Listing 4: ComputeIsocurve.m – Computation of an isocurve with β specified

```

1 function Env = ComputeIsocurve(T, P, b, z, Idx, params, opts)
2 %% Settings
3 % Target number of iterations
4 TargetIter = opts.TargetIter;
5

```

10. An algorithm for computing phase envelopes and an example program

```

6 % Maximum number of Newton iterations
7 kmax = opts.MaxNewtonIter;
8
9 % Tolerance for the Newton iterations
10 newttol = opts.NewtonTol;
11
12 % Tolerance for the detection of the critical point
13 crittol = opts.CritPointTol;
14
15 % Maximum change in specification
16 maxdS = opts.MaxDeltaSpec;
17
18 % Whether to use polynomial regression or not
19 UseCubicInterpolation = opts.UseCubicInterpolation;
20
21 %% Compute initial guess
22 % Compute approximate equilibrium constants
23 K = ComputeWilsonKFactors(T, P, params);
24
25 % Initial guess for the first point
26 X = log([K; T; P]);

```

Listing 5 shows the first part of the computations that we carry out for each point on the isocurve. We solve the isocurve equations that we presented in Section 8 using Newton's method and we check whether we are closed to the critical point. The vapor phase and the liquid phase have the same composition at the critical point. That means that $y_i = x_i$ such that $K_i = y_i/x_i = 1$ and therefore $\ln K_i = 0$. That is how we detect whether we are close to the critical point. We present the function `IsocurveEquations` later. It essentially evaluates, $F(X; S)$, $\frac{\partial F}{\partial X}$, and $\frac{\partial F}{\partial S}$.

Listing 5: ComputeIsocurve.m – Solution of isocurve equations

```

28 %% Compute isocurve
29 % Initial specification
30 S(1) = X(Idx);
31
32 % Initial step
33 dS = maxdS;
34
35 i = 1;
36 CriticalPointReached = false;
37 while(~CriticalPointReached)
38     % Evaluate function and Jacobian
39     [F, dFdX, dFdS] = IsocurveEquations(X(:, i), b, z, S(i), Idx, params, opts);
40
41     k = 0;
42     Converged = (norm(F) < newttol);
43     Diverged = false;
44     while(~Converged && ~Diverged)
45         % Increment iteration counter
46         k = k+1;
47
48         % Newton update

```

10. An algorithm for computing phase envelopes and an example program

```

49     X(:, i) = X(:, i) - dFdX\F;
50
51     % Evaluate function and Jacobian
52     [F, dFdX, dFdS] = IsocurveEquations(X(:, i), b, z, S(i), Idx, params,
    ↪ opts);
53
54     % Check for convergence
55     Converged = (norm(F) < newttol);
56     Diverged  = (k >= kmax);
57 end
58
59 % Check if the critical point has been reached
60 CriticalPointReached = (norm(X(1:end-2, i)) < crittol);

```

Listing 6 shows the remaining part of ComputeIsocurve.m. It computes the sensitivities, $\frac{\partial X}{\partial S}$, and uses them to select the specified variable. If the specified variable is changed, the step size is decreased. If the number of Newton iterations are above or below the target iterations, we double or half the step size, respectively. Next, we ensure that the step size is below the maximum step size, and we use X_s^m and X_s^{m-1} to determine whether S should increase or decrease. We use either cubic interpolation or a linear approximation to compute the initial guess for the subsequent Newton iterations.

Listing 6: ComputeIsocurve.m – Selection of specified variable and step size

```

62 % Compute sensitivities
63 dXdS(:, i) = -dFdX\dFdS;
64
65 % Update selection variable
66 [~, IdxNew] = max(abs(dXdS(:, i)));
67 SpecifiedVariableUpdated = false;
68 if(abs(dXdS(IdxNew, i)) > 1.1*abs(dXdS(Idx, i)))
69     SpecifiedVariableUpdated = true;
70     dS = 0.5*dS;
71     Idx = IdxNew;
72 end
73
74 % Update the step
75 if(k > TargetIter)
76     dS = dS*0.5;
77 elseif(k < TargetIter)
78     dS = dS*2;
79 end
80
81 % Apply upper bound to step size
82 dS = min(dS, maxdS);
83
84 % New specification
85 if(i > 1)
86     fac = sign(X(Idx, i) - X(Idx, i-1));
87 else
88     fac = 1;
89 end
90 S(i+1) = X(Idx, i) + fac*dS;
91

```

10. An algorithm for computing phase envelopes and an example program

```

92     if(i > 1 && ~SpecifiedVariableUpdated && UseCubicInterpolation)
93         X(:, i+1) = PolynomialInitialEstimate ...
94             (S(i+1), X(:, i), S(i), dXdS(:, i), X(:, i-1), S(i-1), dXdS(:, i-1),
95             ↪ opts);
96     else
97         % New initial estimate
98         X(:, i+1) = LinearInitialEstimate ...
99             (X(:, i), dS, dXdS(:, i));
100     end
101     % Display message
102     if(opts.Verbose > 0)
103         fprintf('%3d, %d, %2d), (' , i, Idx, k);
104         fprintf('%7.2f, ', exp(X(:, i)));
105         fprintf('\n');
106     end
107
108     % Store data
109     Env.X(:, i) = X(:, i);
110     Env.k(i) = k;
111
112     % Increment counter
113     i = i+1;
114 end
115 end
116
117 function Xip1 = PolynomialInitialEstimate(Sip1, Xi, Si, dXidS, Xim1, Sim1,
118     ↪ dXim1dS, opts)
119 % Maximum condition number of system matrix in polynomial regression
120 maxcond = opts.MaxCondInterpol;
121
122 % System matrix
123 M = [
124     1, Si, Si^2, Si^3;
125     0, 1, 2*Si, 3*Si^2;
126     1, Sim1, Sim1^2, Sim1^3;
127     0, 1, 2*Sim1, 3*Sim1^2];
128
129 if(cond(M) < maxcond)
130     % Right-hand side
131     rhs = [Xi'; dXidS'; Xim1'; dXim1dS'];
132
133     % Polynomial coefficients
134     a = M\rhs;
135
136     % Extrapolate new initial estimate
137     Xip1 = [1, Sip1, Sip1^2, Sip1^3]*a;
138 else
139     % Step
140     dSip1 = Sip1 - Si;
141
142     % New initial estimate
143     Xip1 = LinearInitialEstimate(Xi, dSip1, dXidS);

```

10. An algorithm for computing phase envelopes and an example program

```
143 end
144 end
145
146 function Xip1 = LinearInitialEstimate(Xi, dSip1, dXidS)
147 % New initial estimate
148 Xip1 = Xi + dXidS*dSip1;
149 end
```

Listing 7 shows the `IsocurveEquations` function. It essentially evaluates $F(X;S)$, $\frac{\partial F}{\partial X}$, and $\frac{\partial F}{\partial S}$. We mostly comment on the usage of the ThermoLib routines. In the phase envelope computations, we consider pressure in MPa. However, ThermoLib requires pressure to be in Pa. The ThermoLib routine `MixFug` computes the logarithmic fugacity coefficients as functions of temperature, pressure, and composition (in moles). It also computes both first and second order derivatives depending on the number of requested outputs. In the phase envelope computations, we only require the first order derivatives so we only request two outputs. Consequently, the second order derivatives are not computed. As we discussed in Section 4, we need to solve the cubic equation of state in order to evaluate the logarithmic fugacity coefficients. ThermoLib solves the cubic equation of state with Newton's method. We therefore specify both the tolerance (`tol`) and the maximum number of Newton iterations (`maxit`) when we call `MixFug`. The remainder of `IsocurveEquations` is a straightforward implementation of the isocurve equations and their derivatives. It only computes the Jacobian and the derivatives of F with respect to S if those outputs are requested.

Listing 7: `IsocurveEquations.m` – Evaluation of the isocurve equations

```
1 function [F, dFdX, dFdS] = IsocurveEquations(X, b, z, S, Idx, params, opts)
2 % Maximum Newton iterations for the equation of state
3 maxit = opts.MaxEosIter;
4
5 % Tolerance for Newton iterations for the equation of state
6 tol = opts.EosTol;
7
8 % Number of components
9 NC = params(5);
10
11 % Extract variables
12 K = exp(X(1:NC));
13 T = exp(X(NC+1));
14 P = exp(X(NC+2));
15
16 % Mole fractions
17 x = z./(1 - b + b*K);
18 y = K.*x;
19
20 % Conversion factor
21 MPa2Pa = 1e6;
22
23 % Compute fugacities with ThermoLib
24 [lnphiv, dlnphiv] = MixFug(T, P*MPa2Pa, y, 0, params, tol, maxit);
25 [lnphil, dlnphil] = MixFug(T, P*MPa2Pa, x, 1, params, tol, maxit);
26
27 % Temperature derivatives
```

10. An algorithm for computing phase envelopes and an example program

```

28 dlnphivT = dlnphiv(:, 1);
29 dlnphilT = dlnphil(:, 1);
30
31 % Scale pressure derivatives
32 dlnphivP = dlnphiv(:, 2)*MPa2Pa;
33 dlnphilP = dlnphil(:, 2)*MPa2Pa;
34
35 % Scale composition derivatives
36 dlnphivnv = dlnphiv(:, 3:end);
37 dlnphilnl = dlnphil(:, 3:end);
38
39 %% Evaluate isocurve equations
40 F = zeros(NC+2, 1);
41 F(1:NC) = log(K) + lnphiv - lnphil;
42 F(NC+1) = sum(y - x);
43 F(NC+2) = X(Idx) - S;
44
45 if(nargout > 1)
46     %% Evaluate Jacobian
47     % Derivatives of K values wrt. ln K values
48     dlnKdlnK = eye(NC,NC);
49
50     % Derivatives of mole fractions wrt. ln K values
51     dxdK = -b*x.^2./z;
52     dydK = x + K.*dxdK;
53
54     % Derivatives of fugacities wrt. ln K values
55     dlnphivdlnK = repmat(K', NC, 1).*dlnphivnv*diag(dydK);
56     dlnphildlnK = repmat(K', NC, 1).*dlnphilnl*diag(dxdK);
57
58     % Jacobian wrt. ln K values
59     dFdX = zeros(NC+2, NC+2);
60     dFdX(1:NC, 1:NC) = dlnKdlnK + dlnphivdlnK - dlnphildlnK;
61
62     % Jacobian wrt. ln T
63     dFdX(1:NC, NC+1) = T*(dlnphivT - dlnphilT);
64
65     % Jacobian wrt. ln P
66     dFdX(1:NC, NC+2) = P*(dlnphivP - dlnphilP);
67
68     % Jacobian wrt. ln K
69     dFdX(NC+1, 1:NC) = K.*(dydK - dxdK);
70
71     % Jacobian wrt. specified variable (ln K, ln T or ln P)
72     dFdX(NC+2, Idx) = 1;
73
74     if(nargout > 2)
75         %% Derivatives wrt. S
76         dFdS = zeros(NC+2, 1);
77         dFdS(NC+2) = -1;
78     end
79 end

```


For completeness, we show CreateFlashOpts in Listing 8. In our experience, the computations can be fairly sensitive to some of the parameters. In particular, large values of MaxDeltaSpec can cause failure to convergence in the Newton iterations. Furthermore, small values of CritPointTol can cause the algorithm to come too close to or overstep the critical point, again leading to failure to converge. Verbose can be set to 1 in order to print out information about each iteration. Large values of EosTol can also cause problems.

Listing 8: CreateFlashOpts.m – Default settings for the computations

```

1 function opts = CreateFlashOpts()
2
3 % Settings for Newton iterations
4 opts.MaxNewtonIter = 2e1;
5 opts.NewtonTol = 1e-6;
6
7 % Settings for the solution of the equation of state
8 opts.MaxEosIter = 2e1;
9 opts.EosTol = 1e-10;
10
11 % Target number of iterations (for selecting step size)
12 opts.TargetIter = 4;
13
14 % Maximum change in specification
15 opts.MaxDeltaSpec = 1e-1;
16
17 % Tolerance for the detection of the critical point
18 opts.CritPointTol = 1e-1;
19
20 % Whether to use polynomial regression to generate initial estimates or not
21 opts.UseCubicInterpolation = true;
22
23 % Maximum condition number of system matrix in polynomial regression
24 opts.MaxCondInterpol = 1e6;
25
26 % Set verbosity
27 opts.Verbose = 0;

```

10.7 Discussion

In this section, we have described a program that computes isocurves, i.e. curves where the vapor fraction, β , is constant. In order to create the phase envelope, we therefore compute the bubble-point curve ($\beta = 0$) and the dew-point curve ($\beta = 1$) separately. Often, we both want to compute the isocurve of β and $1 - \beta$. An example is the phase envelope. Michelsen and Mollerup (2007) suggest to combine the computations, e.g. such that we 1) compute the bubble-point curve, 2) "step over" the critical point, and 3) continue by computing the dew-point curve until a sufficiently low pressure is reached.

A Derivative of mixing term

The expression for the ideal gas mixture entropy (4b) involves a mixing term that contains $\sum_i n_i^v \ln y_i$. We use the derivative of the mixing term when we derive an expression for the chemical potential of an ideal gas mixture in (42). The derivative of the mixing term can appear to be incorrect at first sight. Because the derivative of this mixing term is important to the definition of fugacity, we present the differentiation of $\sum_i n_i^v \ln y_i$ here such that it is clear that it is correct. The derivative is

$$\begin{aligned} \frac{\partial}{\partial n_k^v} \left(\sum_i n_i^v \ln y_i \right) &= \sum_i \left(\frac{\partial n_i^v}{\partial n_k^v} \ln y_i + \frac{n_i^v}{y_i} \frac{\partial y_i}{\partial n_k^v} \right) \\ &= \sum_i \left(\delta_{ik} \ln y_i + \left(\sum_j n_j^v \right) \frac{\partial y_i}{\partial n_k^v} \right) \\ &= \ln y_k + \sum_i \left(\left(\sum_j n_j^v \right) \frac{\partial y_i}{\partial n_k^v} \right). \end{aligned} \quad (151)$$

The derivative of the mole fraction, y_i , is

$$\begin{aligned} \frac{\partial y_i}{\partial n_k^v} &= \frac{1}{\sum_j n_j^v} \frac{\partial n_i^v}{\partial n_k^v} - \frac{n_i^v}{\left(\sum_j n_j^v \right)^2} \\ &= \frac{1}{\sum_j n_j^v} (\delta_{ik} - y_i). \end{aligned} \quad (152)$$

We insert the above into the double sum in (151):

$$\begin{aligned} \sum_i \left(\left(\sum_j n_j^v \right) \frac{\partial y_i}{\partial n_k^v} \right) &= \sum_i (\delta_{ik} - y_i) \\ &= 1 - \sum_i y_i \\ &= 0. \end{aligned} \quad (153)$$

The derivative is therefore

$$\frac{\partial}{\partial n_k^v} \left(\sum_i n_i^v \ln y_i \right) = \ln y_k, \quad (154)$$

where it appears as if one has considered $\ln y_i$ to be independent of all the mole numbers, which is not the case.

B Derivation of the residual Gibbs energy for cubic equations of state

In this section, we evaluate the integral that appears in the expression for the residual Gibbs energy in (35). First, we split up the integral:

$$\begin{aligned} & - \int_{V=\infty}^V \left(\frac{RT}{V/N - b_m} - \frac{a_m}{(V/N + \epsilon b_m)(V/N + \sigma b_m)} - \frac{NRT}{V} \right) dV \\ & = -RT \int_{V=\infty}^V \left(\frac{1}{V/N - b_m} - \frac{NRT}{V} \right) dV + a_m \int_{V=\infty}^V \frac{1}{(V/N + \epsilon b_m)(V/N + \sigma b_m)} dV. \end{aligned} \quad (155)$$

We compute the antiderivative (or primitive function) of the integrand in the left term:

$$\int \left(\frac{1}{V/N - b_m} - \frac{NRT}{V} \right) dV = N \ln \left(\frac{1}{N} - \frac{b_m}{V} \right). \quad (156)$$

Next, we evaluate the integral in the left term using the antiderivative:

$$\begin{aligned} \int_{V=\infty}^V \left(\frac{1}{V/N - b_m} - \frac{NRT}{V} \right) dV & = N \left(\ln \left(\frac{1}{N} - \frac{b_m}{V} \right) - \ln \frac{1}{N} \right) \\ & = N \ln \left(1 - \frac{Nb_m}{V} \right). \end{aligned} \quad (157)$$

The antiderivative of the integrand in the right term is

$$\int \frac{1}{(V/N + \epsilon b_m)(V/N + \sigma b_m)} dV = -\frac{N}{\epsilon - \sigma} \frac{1}{b_m} \ln \left(\frac{V + \epsilon Nb_m}{V + \sigma Nb_m} \right). \quad (158)$$

The integral in the right term is therefore

$$\begin{aligned} \int_{V=\infty}^V \frac{1}{(V/N + \epsilon b_m)(V/N + \sigma b_m)} dV & = -\frac{N}{\epsilon - \sigma} \frac{1}{b_m} \left(\ln \left(\frac{V + \epsilon Nb_m}{V + \sigma Nb_m} \right) - \ln 1 \right) \\ & = -\frac{N}{\epsilon - \sigma} \frac{1}{b_m} \ln \left(\frac{V + \epsilon Nb_m}{V + \sigma Nb_m} \right). \end{aligned} \quad (159)$$

We now rewrite the two expressions that contain volume such that they contain the compressibility factor instead. We use that $PV = NRTZ$ such that $V = NRTZ/P$. We also recall that $B = Pb_m/(RT)$. The one expression is

$$\begin{aligned} 1 - \frac{Nb_m}{V} & = 1 - \frac{Nb_m}{NRTZ/P} \\ & = 1 - \frac{Pb_m}{RTZ} \\ & = 1 - \frac{B}{Z} \\ & = \frac{Z - B}{Z}. \end{aligned} \quad (160)$$

The other expression is

$$\begin{aligned}
 \frac{V + \epsilon N b_m}{V + \sigma N b_m} &= \frac{NRTZ/P + \epsilon N b_m}{NRTZ/P + \sigma N b_m} \\
 &= \frac{NRT/P \frac{Z + \epsilon P b_m / (RT)}{Z + \sigma P b_m / (RT)}}{NRT/P \frac{Z + \epsilon B}{Z + \sigma B}} \\
 &= \frac{Z + \epsilon B}{Z + \sigma B}.
 \end{aligned} \tag{161}$$

The original integral is therefore

$$\begin{aligned}
 & - \int_{V=\infty}^V \left(\frac{RT}{V/N - b_m} - \frac{a_m}{(V/N + \epsilon b_m)(V/N + \sigma b_m)} - \frac{NRT}{V} \right) dV \\
 &= -NRT \ln \left(\frac{Z - B}{Z} \right) - \frac{N}{\epsilon - \sigma} \frac{a_m}{b_m} \ln \left(\frac{Z + \epsilon B}{Z + \sigma B} \right).
 \end{aligned} \tag{162}$$

We insert the expression for the integral into the expression for the residual Gibbs energy and simplify:

$$\begin{aligned}
 G^R(T, P, n) &= - \int_{V=\infty}^V \left(\frac{RT}{V/N - b_m} - \frac{a_m}{(V/N + \epsilon b_m)(V/N + \sigma b_m)} - \frac{NRT}{V} \right) dV \\
 &+ NRT(Z - 1) - NRT \ln Z \\
 &= -NRT \ln \left(\frac{Z - B}{Z} \right) - \frac{N}{\epsilon - \sigma} \frac{a_m}{b_m} \ln \left(\frac{Z + \epsilon B}{Z + \sigma B} \right) + NRT(Z - 1) - NRT \ln Z \\
 &= NRT(Z - 1) - NRT \ln(Z - B) - \frac{N}{\epsilon - \sigma} \frac{a_m}{b_m} \ln \left(\frac{Z + \epsilon B}{Z + \sigma B} \right).
 \end{aligned} \tag{163}$$

C Derivation of the logarithmic fugacity coefficients for cubic equations of state

The expression for the i 'th logarithmic fugacity coefficient depends on the partial derivative of the residual Gibbs energy with respect to mole number i :

$$\ln \phi_i(T, P, n) = \frac{1}{RT} \frac{\partial G^R}{\partial n_i}. \tag{164}$$

The residual Gibbs energy (from a cubic equation of state) is

$$G^R(T, P, n) = NRT(Z - 1) - NRT \ln(Z - B) - \frac{N}{\epsilon - \sigma} \frac{a_m}{b_m} \ln \left(\frac{Z + \epsilon B}{Z + \sigma B} \right). \tag{165}$$

We consider the molar residual Gibbs energy:

$$\begin{aligned}
 g^R(T, P, n) &= G^R(T, P, n)/N \\
 &= RT(Z - 1) - RT \ln(Z - B) - \frac{1}{\epsilon - \sigma} \frac{a_m}{b_m} \ln \left(\frac{Z + \epsilon B}{Z + \sigma B} \right).
 \end{aligned} \tag{166}$$

3. Derivation of the logarithmic fugacity coefficients for cubic equations of state

Conversely, the residual Gibbs energy is $G^R(T, P, n) = Ng^R(T, P, n)$, and the expression for the logarithmic fugacity coefficients is therefore

$$\begin{aligned}\ln \phi_i(T, P, n) &= \frac{1}{RT} \frac{\partial G^R}{\partial n_i} \\ &= \frac{1}{RT} \frac{\partial}{\partial n_i} (Ng^R(T, P, n)) \\ &= \frac{1}{RT} \left(g^R(T, P, n) + N \frac{\partial g^R}{\partial n_i} \right).\end{aligned}\quad (167)$$

The remainder of this section will be concerned with finding an expression for $\partial g^R/\partial n_i$ and use it to derive the expression for the logarithmic fugacity coefficients. First, we use straightforward differentiation:

$$\begin{aligned}\frac{\partial g^R}{\partial n_i} &= RT \frac{\partial Z}{\partial n_i} - RT \frac{1}{Z - B} \left(\frac{\partial Z}{\partial n_i} - \frac{\partial B}{\partial n_i} \right) \\ &\quad - \frac{1}{\epsilon - \sigma} \left[\left(\frac{\partial a_m}{\partial n_i} \frac{1}{b_m} - \frac{a_m}{b_m^2} \frac{\partial b_m}{\partial n_i} \right) \ln \left(\frac{Z + \epsilon B}{Z + \sigma B} \right) + \frac{a_m}{b_m} \frac{\partial}{\partial n_i} \left(\ln \left(\frac{Z + \epsilon B}{Z + \sigma B} \right) \right) \right].\end{aligned}\quad (168)$$

The derivative of the logarithmic coefficient is

$$\begin{aligned}\frac{\partial}{\partial n_i} \left(\ln \left(\frac{Z + \epsilon B}{Z + \sigma B} \right) \right) &= \left(\frac{Z + \epsilon B}{Z + \sigma B} \right)^{-1} \left[\frac{1}{Z + \sigma B} \left(\frac{\partial Z}{\partial n_i} + \epsilon \frac{\partial B}{\partial n_i} \right) - \frac{Z + \epsilon B}{(Z + \sigma B)^2} \left(\frac{\partial Z}{\partial n_i} + \sigma \frac{\partial B}{\partial n_i} \right) \right] \\ &= \frac{1}{Z + \epsilon B} \left(\frac{\partial Z}{\partial n_i} + \epsilon \frac{\partial B}{\partial n_i} \right) - \frac{1}{Z + \sigma B} \left(\frac{\partial Z}{\partial n_i} + \sigma \frac{\partial B}{\partial n_i} \right) \\ &= \frac{Z + \sigma B}{(Z + \epsilon B)(Z + \sigma B)} \left(\frac{\partial Z}{\partial n_i} + \epsilon \frac{\partial B}{\partial n_i} \right) - \frac{Z + \epsilon B}{(Z + \epsilon B)(Z + \sigma B)} \left(\frac{\partial Z}{\partial n_i} + \sigma \frac{\partial B}{\partial n_i} \right) \\ &= \frac{(\sigma - \epsilon)B}{(Z + \epsilon B)(Z + \sigma B)} \frac{\partial Z}{\partial n_i} + \frac{(\epsilon - \sigma)Z}{(Z + \epsilon B)(Z + \sigma B)} \frac{\partial B}{\partial n_i} \\ &= \frac{\epsilon - \sigma}{(Z + \epsilon B)(Z + \sigma B)} \left(Z \frac{\partial B}{\partial n_i} - B \frac{\partial Z}{\partial n_i} \right).\end{aligned}\quad (169)$$

We insert into the expression for $\partial g^R/\partial n_i$:

$$\begin{aligned}\frac{\partial g^R}{\partial n_i} &= RT \frac{\partial Z}{\partial n_i} - RT \frac{1}{Z - B} \left(\frac{\partial Z}{\partial n_i} - \frac{\partial B}{\partial n_i} \right) \\ &\quad - \frac{1}{\epsilon - \sigma} \left[\left(\frac{\partial a_m}{\partial n_i} \frac{1}{b_m} - \frac{a_m}{b_m^2} \frac{\partial b_m}{\partial n_i} \right) \ln \left(\frac{Z + \epsilon B}{Z + \sigma B} \right) + \frac{a_m}{b_m} \frac{\epsilon - \sigma}{(Z + \epsilon B)(Z + \sigma B)} \left(Z \frac{\partial B}{\partial n_i} - B \frac{\partial Z}{\partial n_i} \right) \right] \\ &= RT \frac{\partial Z}{\partial n_i} - RT \frac{1}{Z - B} \left(\frac{\partial Z}{\partial n_i} - \frac{\partial B}{\partial n_i} \right) - \frac{a_m}{b_m} \frac{1}{(Z + \epsilon B)(Z + \sigma B)} \left(Z \frac{\partial B}{\partial n_i} - B \frac{\partial Z}{\partial n_i} \right) \\ &\quad - \frac{1}{\epsilon - \sigma} \left(\frac{\partial a_m}{\partial n_i} \frac{1}{b_m} - \frac{a_m}{b_m^2} \frac{\partial b_m}{\partial n_i} \right) \ln \left(\frac{Z + \epsilon B}{Z + \sigma B} \right).\end{aligned}\quad (170)$$

3. Derivation of the logarithmic fugacity coefficients for cubic equations of state

We recall that $A = Pa_m/(R^2T^2)$ and $B = Pb_m/(RT)$ such that $A/B = a_m/(RTb_m)$. We use that relation to rewrite the three first terms:

$$\begin{aligned}
& RT \frac{\partial Z}{\partial n_i} - RT \frac{1}{Z-B} \left(\frac{\partial Z}{\partial n_i} - \frac{\partial B}{\partial n_i} \right) - \frac{a_m}{b_m} \frac{1}{(Z+\epsilon B)(Z+\sigma B)} \left(Z \frac{\partial B}{\partial n_i} - B \frac{\partial Z}{\partial n_i} \right) \\
&= \left(RT - \frac{RT}{Z-B} + \frac{a_m}{b_m} \frac{B}{(Z+\epsilon B)(Z+\sigma B)} \right) \frac{\partial Z}{\partial n_i} + \left(\frac{RT}{Z-B} - \frac{a_m}{b_m} \frac{Z}{(Z+\epsilon B)(Z+\sigma B)} \right) \frac{\partial B}{\partial n_i} \\
&= RT \left(1 - \frac{1}{Z-B} + \frac{A}{B} \frac{B}{(Z+\epsilon B)(Z+\sigma B)} \right) \frac{\partial Z}{\partial n_i} + RT \left(\frac{1}{Z-B} - \frac{A}{B} \frac{Z}{(Z+\epsilon B)(Z+\sigma B)} \right) \frac{\partial B}{\partial n_i} \\
&= RT \left(1 - \frac{1}{Z-B} + \frac{A}{(Z+\epsilon B)(Z+\sigma B)} \right) \frac{\partial Z}{\partial n_i} + RT \left(\frac{1}{Z-B} - \frac{Z}{B} \frac{A}{(Z+\epsilon B)(Z+\sigma B)} \right) \frac{\partial B}{\partial n_i}. \tag{171}
\end{aligned}$$

We now show that the expression in the left-most parentheses is zero. We modify the three terms such that they have the same denominator:

$$\begin{aligned}
1 - \frac{1}{Z-B} + \frac{A}{(Z+\epsilon B)(Z+\sigma B)} \\
&= \frac{B(Z+\epsilon B)(Z+\sigma B)(Z-B) - B(Z+\epsilon B)(Z+\sigma B) + AZ(Z-B)}{B(Z+\epsilon B)(Z+\sigma B)(Z-B)}. \tag{172}
\end{aligned}$$

Next, we write out the terms in the numerator and express it as cubic polynomial in Z :

$$\begin{aligned}
& (Z+\epsilon B)(Z+\sigma B)(Z-B) - (Z+\epsilon B)(Z+\sigma B) + A(Z-B) \\
&= [Z^3 + (\epsilon+\sigma)BZ^2 + \epsilon\sigma B^2Z - BZ^2 - (\epsilon+\sigma)B^2Z - \epsilon\sigma B^3] \\
&\quad - [Z^2 + (\epsilon+\sigma)BZ + \epsilon\sigma B^2] + [AZ - AB] \\
&= Z^3 + ((\epsilon+\sigma)B - B - 1)Z^2 + (\epsilon\sigma B^2 - (\epsilon+\sigma)B^2 - (\epsilon+\sigma)B + A)Z + (-\epsilon\sigma B^3 - \epsilon\sigma B^2 - AB) \\
&= Z^3 + (B(\epsilon+\sigma-1) - 1)Z^2 + (A - B(\epsilon+\sigma) + B^2(\epsilon\sigma - \epsilon - \sigma))Z + (-AB + (B^2 + B^3)\epsilon\sigma) \tag{173}
\end{aligned}$$

This expression is exactly the polynomial from the cubic equation of state, $q(Z)$. The compressibility factor, Z , must satisfy the cubic equation of state. The above expression is therefore zero, i.e. $q(Z) = 0$, and the first term in (171) is therefore also zero. We use this fact to rewrite the expression in the right-most parentheses in (171). That is, from

$$1 - \frac{1}{Z-B} + \frac{A}{(Z+\epsilon B)(Z+\sigma B)} = 0, \tag{174}$$

we isolate the term that contains A :

$$\frac{A}{(Z+\epsilon B)(Z+\sigma B)} = \frac{1}{Z-B} - 1. \tag{175}$$

We insert the above into the expression in the right-most parentheses in (171) and simplify:

$$\begin{aligned}
 \frac{1}{Z-B} - \frac{Z}{B} \frac{A}{(Z+\epsilon B)(Z+\sigma B)} &= \frac{1}{Z-B} - \frac{Z}{B} \left(\frac{1}{Z-B} - 1 \right) \\
 &= \frac{1}{Z-B} - \frac{Z}{B} \left(\frac{1-(Z-B)}{Z-B} \right) \\
 &= \frac{B-Z(1-(Z-B))}{B(Z-B)} \\
 &= \frac{B-Z+Z^2-BZ}{B(Z-B)} \\
 &= \frac{B(1-Z)-Z(1-Z)}{B(Z-B)} \\
 &= \frac{-(Z-B)(1-Z)}{B(Z-B)} \\
 &= \frac{Z-1}{B}.
 \end{aligned} \tag{176}$$

The expression for $\partial g^R/\partial n_i$ therefore becomes

$$\frac{\partial g^R}{\partial n_i} = RT \frac{Z-1}{B} \frac{\partial B}{\partial n_i} - \frac{1}{\epsilon-\sigma} \frac{1}{b_m} \left(\frac{\partial a_m}{\partial n_i} - \frac{a_m}{b_m} \frac{\partial b_m}{\partial n_i} \right) \ln \left(\frac{Z+\epsilon B}{Z+\sigma B} \right). \tag{177}$$

We write out and simplify the first term:

$$\begin{aligned}
 RT \frac{Z-1}{B} \frac{\partial B}{\partial n_i} &= RT \frac{Z-1}{B} \frac{P}{RT} \frac{\partial b_m}{\partial n_i} \\
 &= RT(Z-1) \frac{1}{b_m} \frac{b_i - b_m}{N} \\
 &= \frac{RT}{N} (Z-1) \left(\frac{b_i}{b_m} - 1 \right).
 \end{aligned} \tag{178}$$

The final expression for $\partial g^R/\partial n_i$ is thus

$$\frac{\partial g^R}{\partial n_i} = \frac{RT}{N} (Z-1) \left(\frac{b_i}{b_m} - 1 \right) - \frac{1}{\epsilon-\sigma} \frac{1}{b_m} \left(\frac{\partial a_m}{\partial n_i} - \frac{a_m}{b_m} \frac{\partial b_m}{\partial n_i} \right) \ln \left(\frac{Z+\epsilon B}{Z+\sigma B} \right). \tag{179}$$

We are now ready to consider the expression for $g^R(T, P, n) + N\partial g^R/\partial n_i$. We insert the expressions for $g^R(T, P, n)$ and $\partial g^R/\partial n_i$:

$$\begin{aligned}
 g^R(T, P, n) + N \frac{\partial g^R}{\partial n_i} &= \left[RT(Z-1) - RT \ln(Z-B) - \frac{1}{\epsilon-\sigma} \frac{a_m}{b_m} \ln \left(\frac{Z+\epsilon B}{Z+\sigma B} \right) \right] \\
 &\quad + N \left[\frac{RT}{N} (Z-1) \left(\frac{b_i}{b_m} - 1 \right) - \frac{1}{\epsilon-\sigma} \frac{1}{b_m} \left(\frac{\partial a_m}{\partial n_i} - \frac{a_m}{b_m} \frac{\partial b_m}{\partial n_i} \right) \ln \left(\frac{Z+\epsilon B}{Z+\sigma B} \right) \right] \\
 &= RT(Z-1) \frac{b_i}{b_m} - \frac{1}{\epsilon-\sigma} \left(\frac{a_m}{b_m} + \frac{N}{b_m} \left(\frac{\partial a_m}{\partial n_i} - \frac{a_m}{b_m} \frac{\partial b_m}{\partial n_i} \right) \right) \ln \left(\frac{Z+\epsilon B}{Z+\sigma B} \right).
 \end{aligned} \tag{180}$$

We write out and simplify the second factor in the second term:

$$\begin{aligned}
 \frac{a_m}{b_m} + \frac{N}{b_m} \left(\frac{\partial a_m}{\partial n_i} - \frac{a_m}{b_m} \frac{\partial b_m}{\partial n_i} \right) &= \frac{a_m}{b_m} + \frac{N}{b_m} \left(\frac{2}{N} \left(\sum_j z_j a_{ij} - a_m \right) - \frac{a_m}{b_m} \frac{b_i - b_m}{N} \right) \\
 &= \frac{1}{b_m} \left(a_m + 2 \left(\sum_j z_j a_{ij} - a_m \right) - a_m \frac{b_i}{b_m} + a_m \right) \\
 &= \frac{1}{b_m} \left(2 \sum_j z_j a_{ij} - a_m \frac{b_i}{b_m} \right). \tag{181}
 \end{aligned}$$

The final expression for $g^R(T, P, n) + N \partial g^R / \partial n_i$ is

$$g^R(T, P, n) + N \frac{\partial g^R}{\partial n_i} = RT(Z - 1) \frac{b_i}{b_m} - \frac{1}{\epsilon - \sigma} \frac{1}{b_m} \left(2 \sum_j z_j a_{ij} - a_m \frac{b_i}{b_m} \right) \ln \left(\frac{Z + \epsilon B}{Z + \sigma B} \right), \tag{182}$$

and we therefore obtain the following expression for the logarithmic fugacity coefficients:

$$\begin{aligned}
 \ln \phi_i(T, P, n) &= \frac{1}{RT} \left(g^R(T, P, n) + N \frac{\partial g^R}{\partial n_i} \right) \\
 &= (Z - 1) \frac{b_i}{b_m} - \frac{1}{\epsilon - \sigma} \frac{1}{RT b_m} \left(2 \sum_j z_j a_{ij} - a_m \frac{b_i}{b_m} \right) \ln \left(\frac{Z + \epsilon B}{Z + \sigma B} \right). \tag{183}
 \end{aligned}$$

References

- Callen, H. B., 1985. Thermodynamics and Introduction to Thermostatistics. John Wiley & Sons.
- Gaspar, J., Ritschel, T. K. S., Jørgensen, J. B., 2017. An efficient and rigorous thermodynamic library and optimal-control of a cryogenic air separation unit. In: Espuna, A., Graells, M., Puigjaner, L. (Eds.), 27th European Symposium on Computer Aided Process Engineering - ESCAPE 27. Barcelona, Spain.
- Michelsen, M., Møllerup, J., 2007. Thermodynamic Models: Fundamentals and Computational Aspects. Tie-Line Publications.
- Michelsen, M. L., 1980. Calculation of phase envelopes and critical points for multicomponent mixtures. Fluid Phase Equilibria 4 (1-2), 1–10.
- Peng, D.-Y., Robinson, D. B., 1976. A new two-constant equation of state. Ind. Eng. Chem. Fundam. 15 (1), 59–64.
- Ritschel, T. K. S., Gaspar, J., Capolei, A., Jørgensen, J. B., 2016. An open-source thermodynamic software library. Tech. Rep. DTU Compute Technical Report-2016-12, Department of Applied Mathematics and Computer Science, Technical University of Denmark.

REFERENCES

- Ritschel, T. K. S., Gaspar, J., Jørgensen, J. B., 2017. A thermodynamic library for simulation and optimization of dynamic processes. In: Proceedings of the 20th World Congress of the International Federation of Automatic Control.
- Soave, G., 1972. Equilibrium constants from a modified redlich-kwong equation of state. *Chemical Engineering Science* 27 (6), 1197–1203.
- Thomson, G., 1996. The DIPPR® databases. *Int. J. Thermophys.* 17 (1), 223–232.
- Wilson, G. M., 1969. A modified Redlich-Kwong equation of state, application to general physical data calculations. In: Proceedings of the 65th National AIChE Meeting. Cleveland, OH.