INAUGURAL-DISSERTATION
zur
Erlangung der Doktorwürde
der
Naturwissenschaftlich-Mathematischen Gesamtfakultät
der
Ruprecht-Karls-Universität Heidelberg

vorgelegt von
Diplom-Mathematiker Robert Kircheis
aus Bad Saarow - Pieskow

Tag der mündlichen Prüfung:
....

Structure Exploiting Parameter Estimation and Optimum Experimental Design Methods and Applications in Microbial Enhanced Oil Recovery

Gutachter: Dr. Stefan Körkel
...

Abstract

In this thesis, we advance efficient methods to solve parameter estimation problems constrained by partial differential equations (PDEs). If PDE constrained parameter estimation problems are solved by derivative based methods, here, the generalized Gauss–Newton method, and multiple shooting, the numerical effort growths drastically with the number of states. The reduced approach couples the computation of the Jacobians and the subsequent block Gaussian elimination using directional derivatives by exploiting the special structure of the constraints which arises from the shooting formulation. Thus, the computational effort is reduced to the one of single shooting. The advantages of the new method in comparison to the common approach are illustrated by means of two academic examples.

Furthermore, we are the first to adapt methods of optimum experimental design for parameter estimation to processes of microbial enhanced oil recovery. We consider a nonlinear coupled PDE model which consists of two parts. The first part, the *black oil* model, describes two phase flow through porous media and a model of convection–diffusion–reaction type depicts the transport and growth effects of bacteria, nutrients, gas and other metabolites in the two phases. A mixed discontinuous Galerkin finite element discretization is applied in space. The discretized model is solved in time by the extended *IMPES* method.

Under the assumption of rotational symmetry, we examine a one dimensional model formulation for parameter estimation and optimum experimental design. We follow the principles of internal numerical differentiation and algorithmic differentiation to evaluate the required derivatives, i.e., the derivatives of the model functions are computed by software tools and we solve the tangential problems with respect to the model parameters and the control variables. By optimum experimental design, a new experiment is planned to reduce the uncertainties of the estimated parameters. The designed experiment differs substantially from the experiments which are usually realized in practice. The confidence intervals for the estimated parameters are reduced by a factor of one hundred.

The developed methods for parameter estimation are implemented in the software package *PAREMERA* which is embedded in the optimum experimental design software *VPLAN*. The model equations for microbial enhanced oil recovery are implemented in a simulation tool which computes not only the nominal equation but also evaluates the derivatives with respect to parameters and controls up to second order.

Zusammenfassung

Ziel dieser Arbeit ist die Weiterentwicklung effizienter Methoden zur Lösung von Parameterschätzproblemen mit partiellen Differentialgleichungen (PDE) als Nebenbedingungen. Wenn PDE beschränkte Parameterschätzprobleme mit ableitungsbasierten Methoden, hier dem verallgemeinerten Gauss-Newton-Verfahren, und einer Mehrzielmethode gelöst werden, steigt der numerische Aufwand mit der Anzahl der Zustände beträchtlich. Der reduzierte Ansatz koppelt die Berechnung der Jacobimatrizen und die anschließende Block-Gauss Elimination durch die Bildung von Richtungsableitungen. Dabei wird die spezielle Struktur der Nebenbedingungen, die durch die Verwendung der Mehrzielmethode entsteht, ausgenutzt. Der numerische Aufwand reduziert sich dadurch auf den Wert des Einfachschiessens. Die Vorteile der neuen Methode werden an Hand zweier akademischer Beispiele erläutert und mit dem herkömmlichen Ansatz verglichen.

Des Weiteren behandeln wir als Erste optimale Versuchsplanung zur Parameterschätzung für Prozesse der mikrobiellen Techniken zur erweiterten Ölförderung. Wir betrachten ein nichtlinear gekoppeltes Modell partieller Differentialgleichungen, welches aus zwei Teilen besteht. Das *black oil* Modell beschreibt Zweiphasenfluß in porösen Medien und ein Modell vom Typ Konvektion-Diffusion-Reaktion beschreibt den Transport und das Wachstum der Bakterien, Nährstoffe, Gase und anderer Stoffwechselprodukte in den zwei Phasen. Wir verwenden ein gemischtes, unstetige Galerkin-Verfahren, um das Modell im Ort zu diskretisieren. Das diskretisierte Modell wird dann mit dem erweiterten *IMPES* Verfahren in der Zeit gelöst.

Unter der Annahme von Rotationssymmetrie betrachten wir ein eindimensionales Modell für die Parameterschätzung und optimale Versuchsplanung. Wir folgen den Prinzipien der internen numerischen Differentiation und der algorithmischen Differentiation, um die benötigten Ableitungen auszuwerten, d.h. die Ableitungen der Modellfunktionen werden durch Softwarewerkzeuge berechnet und wir lösen die tangentialen Gleichungen bezüglich der Modellparameter und Kontrollvariablen. Durch die optimale Versuchsplanung wird ein neues Experiment entworfen, um die statistischen Unsicherheiten der geschätzten Parameter zu reduzieren. Das geplante Experiment unterscheidet sich wesentlich von den Experimenten, die in der Praxis durchgeführt werden. Die Konfidenzintervalle verkleinern sich für die geschätzten Parameter um den Faktor hundert.

Die entwickelten Methoden zur Parameterschätzung wurden im Softwarepaket *PAREMERA* implementiert, das an das Programm zur optimalen Versuchsplanung *VPLAN* gekoppelt ist. Die Gleichungen des Modells für mikrobielle Techniken zur erweiterten Ölförderung wurde in einem Simulator implementiert, der nicht nur die Nominalgleichung löst, sondern

auch die Ableitungen bezüglich der Parameter und den Kontrollvariablen bis zur Ordnung zwei berechnet.

Contents

1 Introduction

> *The study of mathematics, like the Nile, begins in minuteness but ends in magnificence.*
>
> (Charles Caleb Colton)

In engineering and natural sciences, many processes can be described by mathematical models. In order to predict or to optimize the behavior of a process, it is essential that the corresponding model correctly reflects the real behavior of the process. Usually, the mathematical model contains quantities whose values are known only insufficiently and cannot be derived from theoretical considerations. We refer to these quantities as parameters. From measurement data the parameters can be identified which is the general idea of parameter estimation as described in, e.g., Bard [6], Bock [13, 18] and Schlöder [85].

In this thesis, we consider processes which can be modeled by partial differential equations (PDEs). The PDE is considered as a constraint for the parameter estimation problem. We restrict ourselves to direct methods, i.e., we discretize the model first and determine an optimizer for the arising discretized problem afterwards. Here, the PDE is discretized in space by a method of lines (e.g., Schiesser [83], and Deuflhard and Weiser [33]). The approach leads to a system of differential algebraic equations (DAEs). The resulting system of DAEs is parametrized by multiple shooting (Bock and Plitt [17]). We end up with a nonlinear program (NLP) which is solved by a tailored generalized Gauss–Newton method (Bock [13]).

The efficient treatment of PDE constraints is a main topic of this thesis. For a general overview of PDE constrained optimization we refer to, e.g., Hinze et al. [51] and Troeltzsch [94]. When PDEs are discretized by a method of lines, we obtain a high-dimensional system of DAEs, i.e., with a large number of coupled differential and algebraic states. In the context of multiple shooting, the application of the generalized Gauss–Newton method causes an excessive computational effort. Schlöder [85] developed a method for parameter estimation problems constrained by ordinary differential equations (ODEs) that couples the evaluation of the Jacobians for the generalized Gauss–Newton method and the subsequent condensing algorithm by directional derivatives, the *reduced approach*. Bauer [7] extended this method to constraints described by DAEs. The first application to PDE constrained parameter estimation problems was presented by Dieses [34]. Dieses considered only systems of ODEs to approximate the solution of the partial differential equation. We derive a different formulation of the reduced approach for parameter estimation problems constrained by DAEs. We consider the DAE model as an approximation of the solution of partial differential equations. The class of problems, which can be solved by this approach, is widened to the examination of multiple experiments and equality constraints that couple the experiments with each other. The advantages of the reduced approach are illustrated by means of two academic examples. The developed formulation of the reduced approach is implemented in the *FORTRAN* software package *PAREMERA*.

Measurement data are perturbed by measurement errors which leads to uncertainties in the estimated parameter values. The quality of the results of the parameter estimation can be expressed by the confidence regions of the estimated parameters. The confidence regions can be described by the variance–covariance matrix in the solution of the parameter estimation problem. Optimum experimental design (OED) refers to the determination of optimal control settings for the process and measurement points such that an information function on the variance–covariance matrix is maximized (e.g., Körkel [58], Lohmann et al. [67] and Pukelsheim [80]). The OED problem is a nonstandard optimal control problem which has to be solved by tailored methods. The control functions and the constraints are discretized and the resulting NLP is solved by a sequential quadratic programming method. The methods for OED are implemented in *VPLAN* by Körkel et al. [58].

In this thesis, the methods of parameter estimation with single shooting and optimum experimental design are applied to the process of microbial enhanced oil recovery (MEOR) for the first time. First fundamental applications of parameter estimation have been presented by Hazra and Schulz [49] for two phase flow through porous media. We consider a more complex model also for OED. Hron et al. [54] presented parameter estimation for ground water flow which can be described by a model of an equal complexity.

Oil exists in small pore spaces within reservoir rocks underneath the surface of the earth. We distinguish three types of oil recovery. After a production well is drilled into a reservoir, the natural reservoir pressure pushes the oil up to the surface. This is the so-called primary oil recovery. During the oil is recovered in the primary phase, the natural pressure decreases and soon reaches a level where the pressure is insufficient to lift the oil up to the surface. An external force is needed which is accomplished by the injection of water or gas. Thus, an economical oil production rate can be maintained. We refer to the methods as secondary oil recovery. Less then 50 percent of the original oil in place is recovered by primary and secondary production (Gao and Zerki [40]).

MEOR is classified as tertiary or enhanced oil recovery. *In-situ* microorganisms, which are stimulated by the injection of suited nutrients, e.g. sugar and yeast, mobilize residual oil trapped in the pore space. There are no reliable long-term studies, but this technique is assumed to be environmentally-friendly and comparatively cheap since existing infrastructures (e.g. drilled wells) can be used (Maudgalya et al. [72]). When bacteria are provided with nutrient, they reproduce themselves and produce metabolites. The microbial growth at the oil/water interface causes many effects (Kögler [57]). The viscosity of oil is reduced by dissolved gas which is produced by the bacteria. The permeability of the reservoir rock is reduced due to clogging and selective plugging effects and we observe a decrease of the interfacial tension (IFT) which leads to a reduction of the capillary pressure.

We consider a coupled model of partial differential equations to describe the process of microbial enhanced oil recovery. The model consists of the black oil model (Abou et al. [1], and Kou and Sun [60]), which describes two phase flow through porous media, and a transport model of convection–reaction–diffusion type for bateria, nutrients, gas and metabolites (Chang et al. [22] and Desouky et al. [31]). The model equations are discretized in space by a mixed discontinuous Galerkin finite element method (Hesthaven and Warburton [50]). For the temporal discretization, we apply the extended IMPES method (Chang et al. [22] and Berenblyum [9]). The model equation for the pressure is solved by an implicit time step while the saturation of the water phase and the remaining components are computed explicitly. The model equations as well as the time stepping scheme are implemented with the finite element library *deal.ii* (Bangerth et al. [5]). *PAREMERA* is used to identify the parameters of the considered model for microbial enhanced oil recovery. *VPLAN* is applied to solve the OED problem.

There is a considerable amount of existing open source and commercial reservoir simulation softwares: The MATLAB Reservoir Simulation Toolbox (*MRST*) by *SINTEF* Applied Mathematics [66], *BugSim* of the international research institute Stavanger [68], the Implicit Parallel Accurate Reservoir Simulator (*IPARS*) developed by Wheeler et al. at the center for subsurface modeling at the university of Texas at Austin [99, 100] , the Automatic Differentiation General Purpose Research Simulator (*AD-GPRS*) developed at Stanford University [79] and *STARS* from the Computer Modeling Group Ltd. The majority of the simulators lacks the ability to describe the microbial transport and growth inside the reservoir and its multiple effects properly. First papers on the simulation of MEOR processes can be found in, e.g. Chang et al. [22], Maudgalya [71] and Nielsen [76], but a validated model is missing. This thesis can be regarded as a first step into that direction.

1.1 Results of the thesis

We develop a new formulation of the reduced approach for parameter estimation problems constrained by DAEs which fully eliminates the algebraic constraints. We consider DAE models as an approximation for the solution of partial differential equations. The reduced approach lowers the number of required derivatives for the generalized Gauss–Newton method in the context of multiple shooting to the minimal number. The method is implemented in the parameter estimation software package *PAREMERA* which is embedded in the software package *VPLAN*. The new method is tested by means of two academic application examples.

We develop robust numerical methods for the simulation of a model for microbial enhanced oil recovery based on the extended IMPES method with respect to varying parameters, controls and boundary conditions and implement the methods in *deal.ii*.

We use derivative based optimization methods for parameter estimation and optimum experimental design. For the latter, we need to evaluate second order derivatives with respect to parameters and controls. We apply methods of internal numerical differentiation and algorithmic differentiation. For the numerical computations, we implement a software package which simulates the model equations and evaluates the required derivatives for parameter estimation and optimum experimental design. The software is interfaced with *VPLAN* to exchange all necessary quantities.

For the first time, we execute a full cycle of parameter estimation and optimum experimental design for microbial enhanced oil recovery. By optimum experimental design, we reduce the size of the confidence intervals of the estimated parameters for the MEOR model by a factor of 100.

1.2 Thesis overview

The thesis is structured as follows. In Chapter 2, we specify the basic definitions of rock and fluid properties and introduce a nonlinear coupled model for microbial enhanced oil recovery that consists of two parts. The first part describes two phase flow of water and oil through porous media while the second one models the transport phenomena of the bacteria and nutrients as well as the production of gas and other metabolites by the bacteria when provided with nutrients.

In Chapter 3, we discretize the spatial operators by using a mixed discontinuous Galerkin finite element approach. Then, we explain how the discretized model is solved by an IMPES method. The method is extended to an explicit computation of the concentrations of microbes, nutrients, gas and metabolites. We apply our methods to a transient, nonlinear 2D benchmark problem.

We derive the parameter estimation problem constrained by a system of differential algebraic equations and discuss the multiple shooting formulation by Bock et al. for this kind of problems in Chapter 4. We present the generalized Gauss–Newton method for solving parameter estimation problems and give a short overview of the restricted monotonicity test as a globalization strategy. To determine the statistical quality of the estimates, we illustrate how to compute the variance–covariance matrix and define the linear confidence regions. Furthermore, we state Bock's Local Contraction theorem and explain how to evaluate the sensitives of the states with respect to the model parameters.

Chapter 5 is concerned with optimum experimental design problems. We state the general problem formulation and define first-order necessary conditions. Additionally, we present a sequential quadratic programming method to solve the constrained optimization problem and illustrate the evaluation of the mixed second order sensitivities of the states with respect to the parameters and controls for microbial enhanced oil recovery.

In Chapter 6, we discuss a method based on the Schlöder trick to reduce the number of derivatives, which are required to compute the Jacobian for the generalized Gauss–Newton method, to the smallest possible; the reduced approach. We present the features that are implemented in the new parameter estimation software package *PAREMERA* and examine two academic application examples to point out the advantages of *PAREMERA* and the reduced approach.

Chapter 7 contains numerical results for the model for microbial enhanced oil recovery. We describe the sequential procedure of optimum experimental design and parameter estimation. We execute two cycles of the procedure. We illustrate the results of a parameter estimation computed with the methods developed in Chapter 6. Then, a new experiment is planed applying the results of Chapter 5 and the designed experiment is used to determine a statistically better estimate of the model parameters.

The thesis is completed with a summary of the main contributions and an outlook on future research and developments of the model.

1.3 Danksagung

> *Ein guter Anfang braucht Begeisterung, ein gutes Ende Disziplin.*
>
> (Prof. Dr. Quadbeck-Seeger [81])

An dieser Stelle bedanke ich mich aufrichtig und aus tiefstem Herzen bei Dr. Stefan Körkel, dass er in mir die Begeisterung für dieses Promotionsprojekt geweckt hat und für sein Vertrauen, ohne das diese Arbeit nicht hätte entstehen können. Der *Naturwissenschaftlich-Mathematischen Gesamtfakultät* und der *Fakultät für Mathematik und Informatik* der *Ruprecht-Karls-Universität Heidelberg* möchte ich für die Möglichkeit danken, meine Dissertation dort zu erarbeiten.

Des Weiteren danke ich Prof. Dr. Dres. h.c. Hans Georg Bock und Dr. Johannes Schlöder für die zahlreichen guten Ideen und Denkanstöße, die die Fertigstellung dieser Arbeit erst ermöglichten, sowie allen Mitgliedern der Arbeitsgruppe *Simulation and Optimierung* und der Nachwuchsforschungsgruppe *Optimum Experimental Design*, die immer ein offenes Ohr für meine Fragen hatten. Ich bin froh, ein Mitglied dieser Gruppen zu sein.

Ein besonderer Dank gilt außerdem der *BASF SE*, der *Heidelberg Graduate School for Mathematical and Computational Methods for the Sciences*, dem *Zentrum für wissenschaftliches Rechnen* und der Ruprecht-Karls-Universität Heidelberg für die finanzielle und wissenschaftliche Unterstützung während der Zeit meiner Promotion.

Dr. Michael Rieger, Dr. Hakan Alkan, Dr. Eva Mahler, Felix Kögler und allen Beteiligten am Projekt Microbial Enhanced Oil Recovery danke ich für die vielen Anregungen und Ideen bei der Modellierung. Unsere Treffen und Diskussionen gaben mir tiefe Einblicke in die komplexen Vorgänge dieses Prozesses.

Meinen Zimmerkollegen Dennis Janka und Dr. Andreas Schmidt danke ich für die vielen unterhaltsamen Stunden im Büro und auch außerhalb. Unsere nicht immer ganz ernsten Gespräche lassen mich auf fünf schöne Jahre zurückblicken.

Mein Dank gilt in hohem Maße Jens Fangerau und Pavel Hron für die langjährige Organisation unserer Fußballspiele und allen Mitspielern, die mir gezeigt haben, dass ich mich zu Recht gegen eine Karriere als Fußballprofi entschieden habe.

Des Weiteren danke ich Dr. Andreas Schmidt, Pavel Hron, Dr. Andreas Potschka, Dr. Simon Lenz, Dennis Janka und Dr. Kathrin Hatz für das Korrekturlesen einzelner Kapitel dieser Arbeit.

Mein persönlicher Dank gilt Sandra Dingel, meiner Mutter Kerstin Kircheis sowie meiner Schwester Eileen, deren fortwährende Liebe und Unterstützung mir das Vertrauen und die Disziplin gaben, diese Arbeit zu vollenden.

2 Microbial enhanced oil recovery

Even with the growing environmental consciousness among the general public, crude oil is and will stay the main energy source in the world with around 33 percent because of missing affordable and efficient alternatives. With today's state of the technology and the growing demand for energy, crude oil is the only non-renewable resource for which the global demand cannot be satisfied in the next decades (Bundesanstalt für Geowissenschaften und Rohstoffe [21]). Enhanced oil extraction technologies have to be applied. Therefore, a validated model is needed to evaluate the efficiency of the new methods.

The chapter briefly reviews the main definition and notations in reservoir engineering. We introduce the main model that is examined in this thesis. We consider a coupled model consisting of two parts. The basis is the black oil model as it can be found in Abou et al. [1], Hoteit and Firoozabadi [53], and Kou and Sun [60]. The second part models the microbial transport phenomena as well as the production of gas and other metabolic compounds such as acid, surfactants and polymers, see Chang et al. [22], Desouky et al. [31], and Corapcioglu and Haridas [27]. Furthermore, we state the boundary and initial conditions of the model.

2.1 The black oil model

The black oil model describes fluid flow through porous media. Before we derive the system of partial differential equations, we give some essential definitions and settle some basic properties. A more detailed description can be found in Chen [24].

2.1.1 Rock and fluid properties

Porosity is the fraction of a rock which is pore space. We consider exclusively effective porosity which includes all interconnected voids, because only interconnected pore spaces allow fluid flow. Porosity is denoted by ϕ and varies from 0 to 1.

Permeability is the capacity of a rock to transmit fluids through its interconnected pore space. It is represented as $\boldsymbol{K}$ and has the unit square meter (m^2). Usually we assume that $\boldsymbol{K}$ is a diagonal tensor, e.g., for a three dimensional model

$$\boldsymbol{K} = \operatorname{diag}(k_1, k_2, k_3).$$

This conducting capacity is sometimes referred to as absolute permeability. The porous medium is called isotropic if $k_1 = k_2 = k_3$.

Phase refers to a chemically homogeneous region of fluid that is separated from another phase by an interface, e.g., oleic (oil), aqueous (mainly water), or gas. The phases are subscripted by w and o for water and oil phase, respectively.

The **viscosity** of a fluid is a measure of the (frictional) energy dissipated when it is in motion resisting an applied shearing force. It has the unit pascal seconds (Pa·s) and is referred to as μ.

The **fluid saturation** of a phase is the ratio of the pore space that it occupies. It is indicated by S. For two phase flow of water and oil, the saturations S_w and S_o satisfy

$$S_\mathrm{w} + S_\mathrm{o} = 1. \tag{2.1}$$

Capillary pressure denotes the discontinuity in fluid pressure across an interface between any two immiscible fluids (e.g., water and oil). It results as a consequence of the interfacial tension that exists at the interface. The capillary pressure is denoted by

$$p_\mathrm{cow} = p_\mathrm{o} - p_\mathrm{w}, \tag{2.2}$$

where p_o is the pressure of the nonwetting phase (oil) and p_w is the pressure of the wetting phase (water).

Interfacial tension exists when two phases are present and is denoted by σ_ow. These phases can be gas/oil, oil/water, or gas/water. Interfacial tension is the force that holds the surface of a particular phase together and has the unit newton per meter $\left(\frac{N}{m}\right)$.

Relative permeability specifies the amount of diminishment to flow of one phase on another. In two phase flow, it is a function of the phase saturations. The relative permeabilities of water and oil phases are denoted by k_ro and k_rw, respectively.

Typical curves suitable for two phases with water displacing oil are presented in Figure 2.1. The value of the water saturation, at which water starts to flow, is called critical water saturation, S_wc, and the value at which oil stops to flow is referred to as residual oil saturation, S_or. There is a broad variety of analytical expressions for the relationship between relative permeabilities and the water saturation (Corey [28], Naar and Henderson [75], Stone [90]). We use a modified Brooks–Corey relation [20]

$$k_\mathrm{ro} = \hat{k}_\mathrm{ro} \left(\frac{S_\mathrm{o} - S_\mathrm{or}}{1 - S_\mathrm{or} - S_\mathrm{wc}} \right)^{c_\mathrm{o}}, \tag{2.3a}$$

$$k_\mathrm{rw} = \hat{k}_\mathrm{rw} \left(\frac{S_\mathrm{w} - S_\mathrm{wc}}{1 - S_\mathrm{or} - S_\mathrm{wc}} \right)^{c_\mathrm{w}}. \tag{2.3b}$$

The exponents c_o and c_w in equations 2.3 are crucial for the behavior of the water and the oil phase. Their measurement is not possible, but an accurate knowledge is desirable. In Chapter 7, we consider c_o and c_w as model parameters.

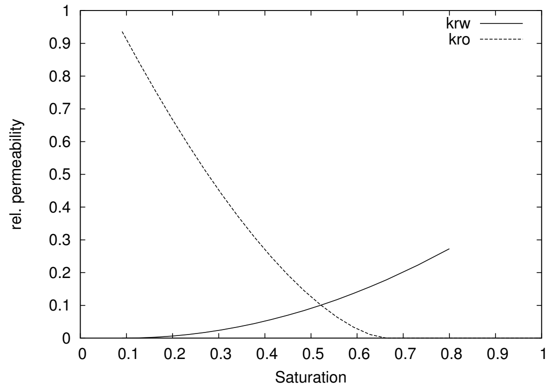


Figure 2.1: Typical curves of the relative permeabilities modeled by Equations (2.3) for a oil/water system.

The **mobility** of a phase is defined as the quotient of its relative permeability and viscosity of the phase. We define

$$\lambda_{\mathrm{o}} := \frac{k_{\mathrm{ro}}}{\mu_{\mathrm{o}}}, \quad \lambda_{\mathrm{w}} := \frac{k_{\mathrm{rw}}}{\mu_{\mathrm{w}}}.$$

Fractional flow is a quantity which determines the fractional volumetric flow rate of a phase with given pressure gradient in the presence of another phase. Symbols for water and oil in a two phase flow system are

$$f_{\mathrm{o}} := \frac{\lambda_{\mathrm{o}}}{\lambda_{\mathrm{t}}}, \quad f_{\mathrm{w}} := \frac{\lambda_{\mathrm{w}}}{\lambda_{\mathrm{t}}},$$

where $\lambda_{\mathrm{t}} := \lambda_{\mathrm{o}} + \lambda_{\mathrm{w}}$ is the total mobility.

2.1.2 Basic differential equations

Let a bounded, continuous domain $\Omega \subset \mathbb{R}^d$, $d = 1,2$ be given. The black oil model is a coupled system of nonlinear time-dependent PDEs that describes two phase flow through porous media. We consider a flow formulation according to Hoteit and Firoozabadi [53], and Kou and Sun [60] and assume that both fluids are incompressible, immiscible and that the phase densities are constant. Then, for nondeformable porous media the mass conservation within each phase is described by

$$\phi \frac{\partial S_\alpha}{\partial t} + \nabla \cdot \boldsymbol{u}_\alpha = q_\alpha, \quad \alpha = \mathrm{w}, \mathrm{o}, \tag{2.4}$$

where q_α is the external mass flow rate and $\boldsymbol{u}_\alpha$ is Darcy's velocity of phase $\alpha = w,\ o$. Darcy's velocity is determined by the extended Darcy's law:

$$\boldsymbol{u}_\alpha = -\frac{k_{\mathrm{r}\alpha}}{\mu_\alpha} \boldsymbol{K} \cdot \nabla \left(p_\alpha + \rho_\alpha g \nabla z\right), \quad \alpha = \mathrm{w}, \mathrm{o}. \tag{2.5}$$

Here, g denotes the gravity acceleration, z is the depth and ρ_α is the density of the corresponding phase.

Additionally, we consider constraint (2.1) which assures that the two phases jointly fill the pore space. We assume that the medium is isotropic. Thus, the permeability tensor can be represented as $\boldsymbol{K} = k \cdot \boldsymbol{I}_d$, where $\boldsymbol{I}_d \in \mathbb{R}^{d \times d}$ is the d-dimensional identity matrix and k is the absolute permeability, see Section 2.1.1. We apply (2.3) for the formulation of the relative permeabilities.

We add equations (2.4) and eliminate the phase saturations to get

$$\nabla \cdot (\boldsymbol{u}_\mathrm{w} + \boldsymbol{u}_\mathrm{o}) = q_\mathrm{w} + q_\mathrm{o}.$$

We apply (2.5) and define

$$\Phi_\mathrm{w} := p_\mathrm{w} + \rho_\mathrm{w} g \nabla z, \quad \Phi_\mathrm{c} := p_\mathrm{cow} - (\rho_\mathrm{o} - \rho_\mathrm{w}) g \nabla z$$

to get

$$\nabla \cdot (\boldsymbol{u}_\mathrm{a} + \boldsymbol{u}_\mathrm{c}) \equiv -\nabla \cdot \lambda_\mathrm{t} \boldsymbol{K} \nabla \Phi_\mathrm{w} - \nabla \cdot \lambda_\mathrm{o} \boldsymbol{K} \nabla \Phi_\mathrm{c} = q_\mathrm{w} + q_\mathrm{o}, \tag{2.6a}$$

$$\phi \frac{\partial S_\mathrm{w}}{\partial t} - q_\mathrm{w} = -\nabla \cdot (f_\mathrm{w} \boldsymbol{u}_\mathrm{a}) \equiv \nabla \cdot \lambda_\mathrm{w} \boldsymbol{K} \nabla \Phi_\mathrm{w}, \tag{2.6b}$$

where we used the definitions for total mobility and fractional flow given in Section 2.1.1. In equation (2.6a), we use the total velocity u_a and the capillary velocity u_c to eliminate the oil pressure p_o. Obviously it holds

$$u_\mathrm{a} + u_\mathrm{c} = u_\mathrm{w} + u_\mathrm{o}$$

The primary states of System (2.6) are the pressure p_w and the saturation of the water phase S_w.

We follow Valvatne and Blunt [96] to model the capillary pressure by the Young–Laplace equation for known shapes of the pores

$$p_\mathrm{cow} = p_\mathrm{o} - p_\mathrm{w} = \frac{2\sigma_\mathrm{ow} \cos\alpha}{R} \tag{2.7}$$

with the oil-water interfacial tension σ_ow and the receding oil water contact angle α and the inscribed radius R, see Figure 2.2. Equation (2.7) is a microscopic formulation for the capillary pressure. As we will see in Chapter 7, we have exact knowledge about the properties of the pores. This is why it is reasonable to use this formulation.

2.2 Microbial transport

The second part of the model predicts the propagation and distribution of the bacteria and the nutrients in porous media. The transport model takes into account the most important effects which influence the transport of bacteria such as diffusion, absorption, growth and decay of bacteria, and consumption of nutrients. Furthermore, the model describes the production of metabolites and accounts for permeability reduction and the changes of the viscosity of the water and the oil phase, respectively.

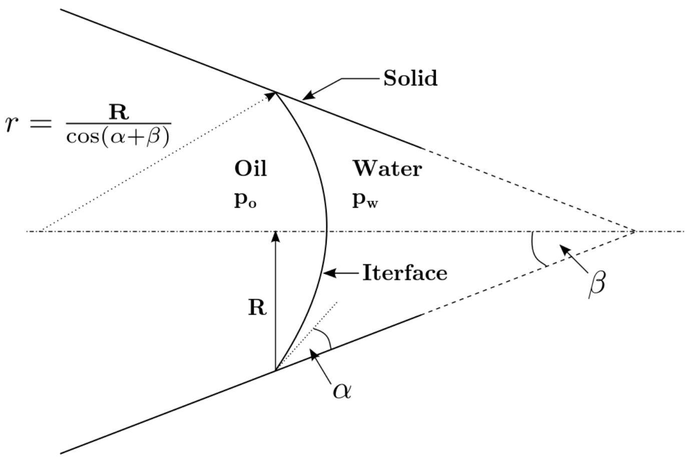


Figure 2.2: Menisci in a conical capillary, see Dullien [37]

2.2.1 Motivation

Due to batch experiments, where bacteria and nutrients have been filled into two interconnected vessels, see Figure 2.3, that have been executed by our project partners, we found out, that the bacteria do not move in the absence of an outer force. We deduce that the bacteria do not show any diffusion effects and do not move in the direction of the nutrient gradient. Because of this, we set the diffusion and chemotaxis coefficients to zero

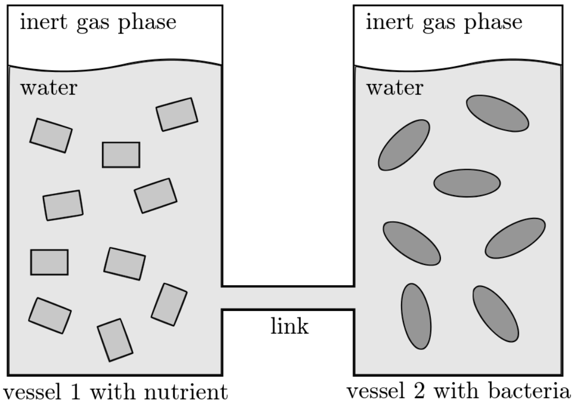


Figure 2.3: Setup of the batch experiment.

$$\boldsymbol{D} = 0, \quad k_{\mathrm{m}} = 0$$

in the equation for the microbial transport.

2.2.2 Mathematical formulation

We assume that bacteria and nutrients are solved in the water phase. Then, the microbial transport in the water phase is described by

$$\frac{\partial(\phi S_{\mathrm{w}} C)}{\partial t} = -\nabla \cdot (\boldsymbol{u}_{\mathrm{w}} C) + \phi S_{\mathrm{w}}(r_{\mathrm{g}} - r_{\mathrm{d}})C + q_{\mathrm{w}}\frac{C}{V} - k_{\mathrm{c}}\phi S_{\mathrm{w}} C, \tag{2.8}$$

see Chang et al. [22]. The primary state in (2.8) is the bacteria concentration C while C_f refers to the nutrient concentration. The equation is of convection–reaction type. The first term characterizes the transport with Darcy velocity $\boldsymbol{u}_{\mathrm{w}}$ and we have to consider several reaction terms.

$$\phi S_{\mathrm{w}}(r_{\mathrm{g}} - r_{\mathrm{d}})C$$

is the growth and decay term with growth rate r_{g} and decay rate r_{d}.

$$q_{\mathrm{w}}\frac{C}{V}$$

describes the source term with well rate q_{w} and bulk volume of the well block V. While

$$k_{\mathrm{c}}\phi S_{\mathrm{w}} C$$

relates to clogging effects, i.e., the bacteria stick together with the walls of the porous medium, with clogging rate k_{c}.

The rate r_{g}, at which new cells are formed, is limited by a maximal bacteria concentration $C_{\max}$ and the availability of nutrient. Nutrients are organic compounds such as sugar and yeast. The growth rate is then written as

$$r_{\mathrm{g}} = \left(1 - \left(\frac{C}{C_{\max}}\right)^{m}\right) \cdot F_{\mathrm{N}}, \tag{2.9}$$

where F_{N} represents the specific growth rate for a given nutrient concentration. The exponent m is a modeling parameter usually chosen between $m = [1,2]$. We use the common Monod expression [74] to compute F_{N}. The nutrient factor, with only one type of nutrient considered, can be described by

$$F_{\mathrm{N}} = \mu_{\max}\left(\frac{C_{\mathrm{f}}}{K_{\mathrm{S}} + C_{\mathrm{f}}}\right), \tag{2.10}$$

with nutrient concentration C_{f} and the half rate constant K_{S} at which half of the maximal growth rate $\mu_{\max}$ is reached. Similar to the exponents in the modified Brooks–Corey relation (2.3), it is not possible to measure $\mu_{\max}$ and K_{S} directly, but their value is very important for the behavior of the process. We add $\mu_{\max}$ and K_{S} to the vector of parameters. At the end of this chapter, we will summarize the model.

The nutrient transport in the water phase is modeled by

$$\begin{aligned}\frac{\partial(\phi S_\mathrm{w} C_\mathrm{f})}{\partial t} =& \nabla\cdot \boldsymbol{D}_\mathrm{f}\cdot\nabla(\phi S_\mathrm{w} C_\mathrm{f}) - \nabla\cdot(\boldsymbol{u}_\mathrm{w} C_\mathrm{f})\\ &-\phi S_\mathrm{w} U_\mathrm{f} r_\mathrm{g} C + q_\mathrm{w}\frac{C_\mathrm{f}}{V},\end{aligned} \tag{2.11}$$

see Chang et al. [22]. Equation (2.11) for the nutrient concentration C_f is of diffusion–convection–reaction type with corresponding diffusion coefficient $\boldsymbol{D}_\mathrm{f}$ and Darcy's velocity $\boldsymbol{u}_\mathrm{w}$. The use of nutrients is described by

$$-\phi S_\mathrm{w} U_\mathrm{f} r_\mathrm{g} C,$$

where U_f is the usage factor of the nutrients for the growth of the bacteria.

When bacteria are provided with nutrients, they do not only start to reproduce themselves but produce gas and other metabolites. We follow Lohne [68] to model these effects. The production of gas relates to

$$\phi S_\mathrm{w} U_\mathrm{g}\frac{r_\mathrm{g} C}{Y_\mathrm{g}}$$

with usage factor U_g, which illustrates how much nutrients are converted into gas, and yield factor Y_g. We assume that the produced gas does not diffuse and that is solved completely in the oil phase. Combined with the convection and source terms, the transport equation for the concentration of gas C_g is given by

$$\frac{\partial(\phi S_\mathrm{o} C_\mathrm{g})}{\partial t} = -\nabla\cdot(\boldsymbol{u}_\mathrm{o} C_\mathrm{g}) + \phi S_\mathrm{w} U_\mathrm{g}\frac{r_\mathrm{g} C}{Y_\mathrm{g}} + q_\mathrm{o}\frac{C_\mathrm{g}}{V}. \tag{2.12}$$

A similar term is defined for the production of the other metabolites

$$\phi S_\mathrm{w} U_\mathrm{m}\frac{r_\mathrm{g} C}{Y_\mathrm{m}},$$

where U_m is the usage factor and Y_m refers to the yield factor for the production of metabolites. For reasons of simplicity of the model, we do not differ between the metabolites. We assume that the metabolites are dissolved completely in the water phase. Again, we consider a diffusion–convection–reaction equation for the concentration of metabolites C_m

$$\begin{aligned}\frac{\partial(\phi S_\mathrm{w} C_\mathrm{m})}{\partial t} =& \nabla\cdot \boldsymbol{D}_\mathrm{m}\cdot\nabla(\phi S_\mathrm{w} C_\mathrm{m}) - \nabla\cdot(\boldsymbol{u}_\mathrm{w} C_\mathrm{m})\\ &+\phi S_\mathrm{w} U_\mathrm{m}\frac{r_\mathrm{g} C}{Y_\mathrm{m}} + q_\mathrm{w}\frac{C_\mathrm{m}}{V}.\end{aligned} \tag{2.13}$$

2.3 Boundary and initial conditions

We complete the model by imposing boundary and initial conditions for the states where needed. We distinguish three different kinds of boundaries: the inflow boundary Γ_in, the outflow boundary Γ_{out} and the enclosing boundary Γ_{en} with

$$\partial\Omega = \Gamma_{\text{in}} \cup \Gamma_{\text{out}} \cup \Gamma_{\text{en}} \tag{2.14}$$

and

$$\Gamma_{\text{in}} \cap \Gamma_{\text{out}} = \emptyset,\ \Gamma_{\text{in}} \cap \Gamma_{\text{en}} = \emptyset,\ \Gamma_{\text{en}} \cap \Gamma_{\text{out}} = \emptyset, \tag{2.15}$$

see Figure 2.4 for a 2D illustration. Note that for $\Omega \subset \mathbb{R}^1$ we have $\Gamma_{\text{en}} = \emptyset$. The inflow boundary is the part of $\partial\Omega$ where the water enriched with bacteria and nutrient, respectively, is injected while the outflow boundary is the part where the oil water mixture flows out. The enclosing boundary is a shell where no inflow or outflow is possible.

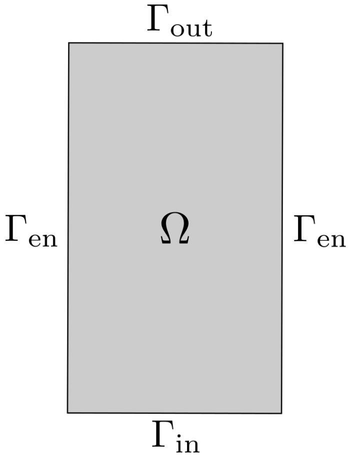


Figure 2.4: Domain in 2D.

2.3.1 Black oil model

Since there is no derivative with respect to time in Equation (2.6a), we do not need to define initial conditions for the pressure. We employ three types of boundary conditions for the p_{w}. On Γ_{out}, we formulate Dirichlet-type boundary values

$$p_{\text{w}} = p^D \text{ on } \Gamma_{\text{out}}.$$

On Γ_{in}, inhomogeneous Neumann-type boundary conditions are applied

$$\boldsymbol{n} \cdot (\boldsymbol{u}_{\text{a}} + \boldsymbol{u}_{\text{c}}) = q^{\text{in}} \text{ on } \Gamma_{\text{in}},$$

while on the enveloping boundary, homogeneous Neumann-type boundary conditions are imposed

$$\boldsymbol{n} \cdot (\boldsymbol{u}_{\text{a}} + \boldsymbol{u}_{\text{c}}) = 0 \text{ on } \Gamma_{\text{en}},$$

where $\boldsymbol{n}$ is the outward normal vector to Γ_{in} and Γ_{en}, respectively. We refer to q^{in} as inflow velocity. In Chapter 7, we use q^{in} as a control variable for the optimum experimental design problem.

We impose Dirichlet-type boundary conditions on the inflow boundary for the water saturation

$$S_{\mathrm{w}} = S_{\mathrm{w}}^{\mathrm{in}} \text{ on } \Gamma_{\mathrm{in}}.$$

On Γ_{out}, we do not need to specify any boundary conditions for the water saturations since we consider a transport problem. On the enveloping boundary, homogeneous Neumann-type boundary conditions are stated again

$$\frac{\partial S_{\mathrm{w}}}{\partial \boldsymbol{n}} = 0 \text{ on } \Gamma_{\mathrm{en}}.$$

The initial saturation at the beginning of the water flood is set to

$$S_{\mathrm{w}} = S^0 \text{ in } \Omega.$$

2.3.2 Microbial and nutrient transport

We consider transport dominant problems, i.e., with a high the Péclet number

$$Pe = \frac{L \cdot \|\boldsymbol{u}_{\mathrm{a}}\|}{D_i}, \quad i = \mathrm{f}, \mathrm{m},$$

where L is the characteristic length.

On Γ_{in}, we formulate Dirichlet-type boundary conditions for the concentration of the bacteria

$$C = C^{\mathrm{in}} \text{ on } \Gamma_{\mathrm{in}}$$

and for the nutrient concentration

$$C_{\mathrm{f}} = C_{\mathrm{f}}^{\mathrm{in}} \text{ on } \Gamma_{\mathrm{in}}.$$

The variables C^{in} and $C_{\mathrm{f}}^{\mathrm{in}}$ will are considered as control functions in Chapter 7. As mentioned above, we examine transport dominant problems. Thus, we do not have to define any boundary conditions for the bacteria and nutrient concentrations on the outflow boundary.

Homogeneous Neumann-type boundary conditions are applied for both concentration on the enveloping boundary

$$\frac{\partial C}{\partial \boldsymbol{n}} = 0, \quad \frac{\partial C_{\mathrm{f}}}{\partial \boldsymbol{n}} = 0 \text{ on } \Gamma_{\mathrm{en}}.$$

The initial concentrations of bacteria and nutrients at the beginning of the time interval are set to

$$C = 0, \quad C_{\mathrm{f}} = 0 \text{ in } \Omega.$$

Since the equations for the gas and metabolites of the same type as Equations (2.8) and (2.11), respectively, we impose the same type of boundary conditions. On Γ_{in}, we formulate homogeneous Dirichlet-type conditions

$$C_{\mathrm{g}} = 0, \quad C_{\mathrm{m}} = 0 \text{ on } \Gamma_{\mathrm{out}}$$

and on Γ_{out}, we do not have to impose boundary conditions. Since there is no flow through the enveloping boundary, we have

$$\frac{\partial C_\mathrm{g}}{\partial \boldsymbol{n}} = 0, \quad \frac{\partial C_\mathrm{m}}{\partial \boldsymbol{n}} = 0 \text{ on } \Gamma_\mathrm{en}.$$

The initial gas and metabolites concentrations at $t = t_0$ are set to

$$C_\mathrm{g} = 0, \quad C_\mathrm{m} = 0 \text{ on } \Omega.$$

2.4 Effects of the bacteria

We follow Lohne [68] to characterize the effects of the bacteria and the metabolites on the fluid and rock properties. Since the viscosity of gas is lower than the viscosity of oil, the dissolved gas lowers the viscosity of the oil phase and thus raises its mobility. Let μ_g be the viscosity of gas and let $\hat{\mu}_\mathrm{o}$ be the viscosity of pure oil. Then, the viscosity of the oil phase is computed as

$$\mu_\mathrm{o} = \frac{\hat{\mu}_\mathrm{o}}{\left[1 - \left(1 - \left(\frac{\hat{\mu}_\mathrm{o}}{\mu_\mathrm{g}}\right)^{0.25}\right) C_\mathrm{g}\right]^4}. \tag{2.16}$$

A contrary behavior is observed for the aqueous phase. The viscosity of the metabolites μ_m is higher than the viscosity of water $\hat{\mu}_\mathrm{w}$. We deduce that the viscosity of the aqueous phase increases with the concentration of metabolites as

$$\mu_\mathrm{w} = \frac{\hat{\mu}_\mathrm{w}}{\left[1 - \left(1 - \left(\frac{\hat{\mu}_\mathrm{w}}{\mu_\mathrm{m}}\right)^{0.25}\right) C_\mathrm{m}\right]^4}. \tag{2.17}$$

Additionally, we consider the effect that the clogged or immobile bacteria reduce the permeability of the porous medium. Let N_b denote the the number of clogged bacteria and R_b refers to the average radius of one microbe. Then the reduced permeability K' is computed by

$$\frac{K'}{K} = \left[1 + 2.483\sqrt{\frac{K}{\phi}} N_\mathrm{b} \pi R_\mathrm{b}^2\right]^2, \tag{2.18}$$

see Lohne [68].

2.5 Summarization of the model equations

In the introduction, we mentioned that we want to apply methods of parameter estimation and optimum experimental design to our model for microbial enhanced oil recovery. In

Sections 2.1.2 and 2.2.2, we have defined the vector of parameters that we want to identify. We refer to them as

$$\boldsymbol{\theta} = (c_{\mathrm{o}}, c_{\mathrm{w}}, \mu_{\max}, K_{\mathrm{S}})^T . \tag{2.19}$$

The control variables that are optimized by OED are defined in Section 2.3

$$q = q^{\mathrm{in}}, \tag{2.20a}$$

$$\boldsymbol{u}(t) = \left(C^{\mathrm{in}}, C_{\mathrm{f}}^{\mathrm{in}}\right)^T . \tag{2.20b}$$

To improve the readability, we summarize our model equations (2.6), (2.8), (2.11), (2.12) and (2.13) to a more general formulation of a PDE

$$\mathscr{F}(t, \boldsymbol{\xi}(t, \boldsymbol{\theta}, \boldsymbol{q}, \boldsymbol{u}(t)), \boldsymbol{\theta}, \boldsymbol{q}, \boldsymbol{u}(t)) = 0, \tag{2.21}$$

where

$$\boldsymbol{\xi} = (p,\ S_{\mathrm{w}},\ C,\ C_{\mathrm{f}},\ C_{\mathrm{g}},\ C_{\mathrm{m}})^T$$

denotes the vector of states that depend on the choice of the parameters and controls.

3 Numerical simulation of microbial enhanced oil recovery

In this chapter, we deduce the solution techniques that we apply to solve the model equations derived in Chapter 2. We develop a mixed discontinuous Galerkin finite element method that is used to discretize the model in space. Additionally, we introduce the IMPES method and extend the method. The extended IMPES method is applied to solve the model in time. We explain how the length of the time step is determined. We perform a 2D simulation example of the full model in the final section.

3.1 Finite element approach

In this section, we explain the discretization in space of the model equations (2.6), (2.8), (2.11), (2.12) and (2.13). Since we consider convection dominated transport problems, we apply a mixed discontinuous Galerkin finite element method (DG-FEM). Finite element methods (FEMs) are superior in complex geometries to, e.g., finite difference methods and allow a profound mathematical analysis. There is a broad variety of text books concerning FEMs. For an overview of FEMs calculus, see for instance Braess [19] or Grossmann and Roos [47]. For an introduction to DG-FEMs, we refer to Hesthaven and Warburton [50]. DG-FEMs combine features of the finite element and the finite volume framework. By using a space of basis and test functions, DG-FEMs emulate the finite element methods and still satisfies the equation in a sense closer to the finite volume methods (FVMs).

The first analysis of DG-FEMs was presented in Reed and Hill [82]. The application to nonlinear conservation laws was first presented in Chavent and Salzano [23]. We use the FEMs library deal.ii, an open source software which is widely used in the community of scientific computing, see Bangerth et al. [5].

Since we do not demand continuity along the intersection of two elements, we have two or more solutions at the same physical location along the trace of an element. In the following, we refer to the information of the current element by a superscript "−" and to the information of the neighbor element by a superscript "+". With this notation, we define the average

$$\{\{a\}\} = \frac{a^- + a^+}{2},$$

where a can be a scalar or a vector. Likewise, we define jumps along a normal vector, $\boldsymbol{n}$, as

$$[\![a]\!] = \boldsymbol{n}^- a^- + \boldsymbol{n}^+ a^+, \; [\![\boldsymbol{a}]\!] = \boldsymbol{n}^- \cdot \boldsymbol{a}^- + \boldsymbol{n}^+ \cdot \boldsymbol{a}^+.$$

Note that we have different definitions for the jumps depending on whether a is a scalar or a vector.

We decompose the domain Ω by a regular, quadrilateral triangulation $\mathscr{T}$ consisting of elements T_h. We use isoparametric bilinear and constant elements and denote the underlying polynomial spaces with

$$\begin{aligned} Q_1(T_h) &= \operatorname{span}\{1\}, \\ Q_2(T_h) &= \operatorname{span}\{1, x_1, x_2\}, \end{aligned}$$

for a 2D model. The bilinear elements have their support points in the corners of the cells T_h. We apply an H^1-conform finite element ansatz

$$\begin{aligned} V_h = \{&\varphi_h \in H^1(\Omega): \; \varphi_{h|_T} \in Q_2(T_h), \; T_h \in \mathscr{T}, \; \varphi_h \text{ continuous along the edges of } T_h \\ &\text{and } \varphi_h = 0 \text{ on } \Gamma_{\text{out}}\}, \end{aligned}$$

for the pressure p_w. For the water saturation S_w, the bacteria concentration C, the nutrient concentration C_f, the gas concentration C_g and the metabolite concentration C_m, we consider

$$U_h = \{\sigma_h \in H^1(\Omega): \; \sigma_{h|_T} \in Q_1(T_h), \; T_h \in \mathscr{T}, \; \sigma_h = 0 \text{ on } \Gamma_{\text{out}} \cup \Gamma_{\text{in}}\}.$$

Here, we use the partitioning of the boundary from Section 2.3

$$\partial\Omega = \Gamma_{\text{in}} \cup \Gamma_{\text{out}} \cup \Gamma_{\text{en}}.$$

On the parts of the boundary where we impose Dirichlet-type boundary conditions, Γ_{out} for φ_h and $\Gamma_{\text{in}} \cup \Gamma_{\text{out}}$ for σ_h, the ansatz functions are set to zero. On the remaining parts of the boundary Γ_{en} and $\Gamma_{\text{in}} \cup \Gamma_{\text{en}}$, respectively, Neumann-type conditions have to be fulfilled.

We do not consider mass flow. In the following, we set

$$q_\text{w} = 0, \quad q_\text{o} = 0.$$

We formulate the weak form of Equation (2.6a) and integrate by parts

$$0 = (-\nabla \cdot [\lambda_\text{t} \boldsymbol{K} \cdot \nabla \Phi_{\text{w},h}], \varphi_h)_\Omega + (-\nabla \cdot [\lambda_\text{n} \boldsymbol{K} \cdot \nabla \Phi_\text{c}], \varphi_h)_\Omega \tag{3.1}$$

$$\begin{aligned} &= (\lambda_\text{t} \boldsymbol{K} \cdot \nabla \Phi_{\text{w},h}, \nabla \varphi_h)_\Omega + (\lambda_\text{n} \boldsymbol{K} \cdot \nabla \Phi_\text{c}, \nabla \varphi_h)_\Omega \\ &\quad - (\boldsymbol{n} \cdot [\lambda_\text{t} \boldsymbol{K} \cdot \nabla \Phi_{\text{w},h}], \varphi_h)_{\Gamma_{\text{in}}} - (\boldsymbol{n} \cdot [\lambda_\text{n} \boldsymbol{K} \cdot \nabla \Phi_\text{c}], \varphi_h)_\Omega. \end{aligned} \tag{3.2}$$

From Darcy's law (2.5) it follows

$$0 = (\lambda_\text{t} \boldsymbol{K} \cdot \nabla \Phi_{\text{w},h}, \nabla \varphi_h)_\Omega + (\lambda_\text{n} \boldsymbol{K} \cdot \nabla \Phi_\text{c}, \nabla \varphi_h)_\Omega$$

$$
\begin{aligned}
&+ (\boldsymbol{n} \cdot [u_{a,h} + u_{c,h}], \varphi_h)_{\Gamma_{\text{in}}} \\
&= (\lambda_{\text{t}} \boldsymbol{K} \cdot \nabla \Phi_{\text{w},h}, \nabla \varphi_h)_\Omega + (\lambda_{\text{n}} \boldsymbol{K} \cdot \nabla \Phi_{\text{c}}, \nabla \varphi_h)_\Omega + (q^{\text{in}}, \varphi_h)_{\Gamma_{\text{in}}}.
\end{aligned} \tag{3.3}
$$

The weak form of the mass conservation law (2.6b) results in

$$
\begin{aligned}
\left(\phi \frac{\partial S_{\text{w},h}}{\partial t}, \sigma_h\right)_\Omega &= -\left(\nabla \cdot [f_{\text{w}} \boldsymbol{u}_{a,h}], \sigma_h\right)_\Omega \\
&= \sum_{T_h \in \mathscr{T}} \left((f_{\text{w}} \boldsymbol{u}_{a,h}, \nabla \sigma_h)_{T_h} - (f_{\text{w}} \boldsymbol{n} \cdot \boldsymbol{u}_{a,h}, \sigma_h)_{\partial T_h}\right)
\end{aligned} \tag{3.4}
$$

Since we consider discontinuous spaces and discontinuous functions are not defined on the boundary of the cells, we have to specify how to evaluate the terms on the interfaces between the cells. We evaluate the second term on the right hand side of the weak saturation equation (3.4) following the approach of Li and Bangerth [65] by using an upwind stabilization:

$$
\begin{aligned}
\left(f_{\text{w}}(S_{\text{w},h}) \boldsymbol{n} \cdot \boldsymbol{u}_{a,h}, \sigma_h\right)_{\partial T_h} &= \left(f_{\text{w}}(S^+_{\text{w},h})(\boldsymbol{n} \cdot \boldsymbol{u}^+_{a,h}), \sigma_h\right)_{\partial T^+_{h,a}} \\
&\quad + \left(f_{\text{w}}(S^-_{\text{w},h})(\boldsymbol{n} \cdot \boldsymbol{u}^-_{a,h}), \sigma_h\right)_{\partial T^-_{h,a}},
\end{aligned} \tag{3.5}
$$

where $\partial T^+_{h,a} := \{x \in \partial T_h,\ \boldsymbol{n} \cdot \boldsymbol{u}_{a,h}(x) < 0\}$ denotes the inflow boundary and the outflow boundary is identified by $\partial T^-_{h,a} := \partial T_h \backslash \partial T^+_{h,a}$. The quantities $S^-_{\text{w},h}, \boldsymbol{u}^-_{a,h}$ then correspond to the values of these variables on the boundary of the present cell, whereas $S^+_{\text{w},h}, \boldsymbol{u}^+_{a,h}$ (needed on the inflow part of the boundary of T_h) are quantities taken from the boundary of the neighboring cell.

We apply the same considerations to the equation for the transport of the bacteria (2.8) in order to obtain

$$
\begin{aligned}
\left(\frac{\partial(\phi S_{\text{w},h} C_h)}{\partial t}, \sigma_h\right)_\Omega &= \sum_{T_h \in \mathscr{T}} \left((\boldsymbol{u}_{\text{w},h} C_h, \nabla \sigma_h)_{T_h} - (C_h \boldsymbol{n} \cdot \boldsymbol{u}_{\text{w},h}, \sigma_h)_{\partial T_h}\right) \\
&\quad + \left(\phi S_{\text{w},h}[r_{\text{g}} - r_{\text{d}}] C_h, \sigma_h\right)_\Omega - (k_{\text{c}} \phi S_{\text{w},h} C_h, \sigma_h)_\Omega.
\end{aligned} \tag{3.6}
$$

Again, we have to specify the meaning of the terms on the boundaries of the cells. We use a stabilization similar to (3.5)

$$
\left(C_h \boldsymbol{n} \cdot \boldsymbol{u}_{\text{w},h}, \sigma_h\right)_{\partial T_h} = \left(C^+_h (\boldsymbol{n} \cdot \boldsymbol{u}^+_{\text{w},h}), \sigma_h\right)_{\partial T^+_{h,w}} + \left(C^-_h (\boldsymbol{n} \cdot \boldsymbol{u}^-_{\text{w},h}), \sigma_h\right)_{\partial T^-_{h,w}}, \tag{3.7}
$$

with the inflow boundary $\partial T^+_{h,w} := \{x \in \partial T_h,\ \boldsymbol{n} \cdot \boldsymbol{u}_{\text{w},h}(x) < 0\}$ and the outflow boundary $\partial T^-_{h,w} := \partial T_h \backslash \partial T^+_{h,w}$.

For the nutrient transport (2.11), we account also for diffusion effects. First, we apply the product rule and get

$$
\frac{\partial(\phi S_{\text{w}} C_{\text{f}})}{\partial t} = \nabla \cdot [\boldsymbol{D}_{\text{f}} \cdot \nabla(\phi S_{\text{w}} C_{\text{f}})] - \nabla \cdot (\boldsymbol{u}_{\text{w}} C_{\text{f}}) - \phi S_{\text{w}} U_{\text{f}} r_{\text{g}} C
$$

$$= \nabla \cdot [\boldsymbol{D}_\mathrm{f} \cdot (\phi C_\mathrm{f} \nabla S_\mathrm{w} + \phi S_\mathrm{w} \nabla C_\mathrm{f})] - \nabla \cdot (\boldsymbol{u}_\mathrm{w} C_\mathrm{f}) - \phi S_\mathrm{w} U_\mathrm{f} r_\mathrm{g} C. \tag{3.8}$$

We follow Hesthaven and Warburton [50] to formulate the weak formulation. We rewrite (3.8) as a problem of first order in C_f

$$\boldsymbol{q}_1 = \sqrt{S_\mathrm{w} \boldsymbol{D}_\mathrm{f}} \cdot \nabla C_\mathrm{f}, \tag{3.9}$$

$$\begin{aligned} \frac{\partial(\phi S_\mathrm{w} C_\mathrm{f})}{\partial t} &= \phi \nabla \cdot [C_\mathrm{f} \boldsymbol{D}_\mathrm{f} \cdot \nabla S_\mathrm{w}] + \phi \nabla \cdot \left[\sqrt{S_\mathrm{w} \boldsymbol{D}_\mathrm{f}} \cdot \boldsymbol{q}_1\right] \\ &\quad - \nabla \cdot (\boldsymbol{u}_\mathrm{w} C_\mathrm{f}) - \phi S_\mathrm{w} U_\mathrm{f} r_\mathrm{g} C \end{aligned} \tag{3.10}$$

to obtain a system which can be discretized using techniques for the conservation laws. We obtain the weak formulation for (3.9)

$$\begin{aligned} (\boldsymbol{q}_{1,h}, \boldsymbol{\eta}_h)_\Omega &= (\sqrt{S_{\mathrm{w},h} \boldsymbol{D}_\mathrm{f}} \cdot \nabla C_{\mathrm{f},h}, \boldsymbol{\eta}_h)_\Omega \\ &= \left(\nabla C_{\mathrm{f},h}, \sqrt{S_{\mathrm{w},h} \boldsymbol{D}_\mathrm{f}} \cdot \boldsymbol{\eta}_h\right)_\Omega \\ &= \sum_{T_h \in \mathscr{T}} \Big(- \left(C_{\mathrm{f},h}, \nabla \cdot \left[\sqrt{S_{\mathrm{w},h} \boldsymbol{D}_\mathrm{f}} \cdot \boldsymbol{\eta}_h\right]\right)_{T_h} \\ &\qquad + \left(\boldsymbol{n} \cdot \left(\sqrt{S_{\mathrm{w},h} \boldsymbol{D}_\mathrm{f}} C_{\mathrm{f},h}\right)^*, \boldsymbol{\eta}_h\right)_{\partial T_h} \Big). \end{aligned} \tag{3.11}$$

Here and in the following, $\boldsymbol{\eta}_h$ denotes the multidimensional version of σ_h that corresponds to the dimension of Ω. Since the diffusion parameter $\tilde{\boldsymbol{D}}_\mathrm{f} = \sqrt{S_\mathrm{w} \boldsymbol{D}_\mathrm{f}}$ is only piecewise smooth, we define the numerical flux according to

$$\left(\sqrt{S_{\mathrm{w},h} \boldsymbol{D}_\mathrm{f}} C_{\mathrm{f},h}\right)^* = \{\{\sqrt{S_{\mathrm{w},h} \boldsymbol{D}_\mathrm{f}} C_{\mathrm{f},h}\}\} + \tfrac{1}{2} [\![\sqrt{S_{\mathrm{w},h} \boldsymbol{D}_\mathrm{f}}]\!] \, C_{\mathrm{f},h}^+, \tag{3.12}$$

see Hesthaven and Warburton [50]. The weak formulation of (3.10) then reads

$$\begin{aligned} \left(\frac{\partial(\phi S_{\mathrm{w},h} C_{\mathrm{f},h})}{\partial t}, \sigma_h\right) &= - (\phi C_{\mathrm{f},h} \boldsymbol{D}_{\mathrm{f},h} \cdot \nabla S_{\mathrm{w},h}, \nabla \sigma_h)_\Omega \\ &\quad + \sum_{T_h \in \mathscr{T}} \Big(- \left(\phi \sqrt{S_{\mathrm{w},h} \boldsymbol{D}_\mathrm{f}} \cdot \boldsymbol{q}_{1,h}, \nabla \sigma_h\right)_{T_h} \\ &\qquad + \left(\phi \boldsymbol{n} \cdot \left(\sqrt{S_{\mathrm{w},h} \boldsymbol{D}_\mathrm{f}} \cdot \boldsymbol{q}_{1,h}\right)^*, \sigma_h\right)_{\partial T_h} \Big) \\ &\quad + \sum_{T_h \in \mathscr{T}} \left((\boldsymbol{u}_{\mathrm{w},h} C_{\mathrm{f},h}, \nabla \sigma_h)_{T_h} - (C_{\mathrm{f},h} \boldsymbol{n} \cdot \boldsymbol{u}_{\mathrm{w},h}, \sigma_h)_{\partial T_h} \right) \\ &\quad - (\phi S_{\mathrm{w},h} U_\mathrm{g} r_\mathrm{g} C_\mathrm{h}, \sigma_h)_\Omega. \end{aligned} \tag{3.13}$$

Once more we have to give a meaning to the terms at the interfaces of the cells. For the specification of the diffusion term, we refer to Hesthaven and Warburton [50]. We consider the numerical flux

$$\left(\sqrt{S_{\mathrm{w},h} \boldsymbol{D}_\mathrm{f}} \cdot \boldsymbol{q}_{1,h}\right)^* = \{\{\sqrt{S_{\mathrm{w},h} \boldsymbol{D}_\mathrm{f}} \cdot \boldsymbol{q}_{1,h}\}\} + \tfrac{1}{2} [\![\sqrt{S_{\mathrm{w},h} \boldsymbol{D}_\mathrm{f}}]\!] \cdot \boldsymbol{q}_{1,h}^+. \tag{3.14}$$

We evaluate the transport over the cell boundary as before in Equations (3.5) and (3.7)

$$\left(C_{\mathrm{f},h}\boldsymbol{n}\cdot\boldsymbol{u}_{\mathrm{w},h},\sigma_h\right)_{\partial T_h}=\left(C^+_{\mathrm{f},h}(\boldsymbol{n}\cdot\boldsymbol{u}^+_{\mathrm{w},h}),\sigma_h\right)_{\partial T^+_{h,w}}+\left(C^-_{\mathrm{f},h}(\boldsymbol{n}\cdot\boldsymbol{u}^-_{\mathrm{w},h}),\sigma_h\right)_{\partial T^-_{h,w}},\tag{3.15}$$

where we use the same definitions for $\partial T^+_{h,w}$ and $\partial T^-_{h,w}$ as in (3.7).

The equations for transport of the gas (2.12) and the metabolites (2.13) are of the same type as (2.8) and (2.11), respectively. Hence we apply the same arguments as above to obtain the corresponding weak formulations

$$\begin{aligned}\left(\frac{\partial(\phi S_{\mathrm{o},h}C_{\mathrm{g},h})}{\partial t},\sigma_h\right)_\Omega=&\sum_{T_h\in\mathscr{T}}\left((\boldsymbol{u}_{\mathrm{o},h}C_{\mathrm{g},h},\nabla\sigma_h)_{T_h}-(C_{\mathrm{g},h}\boldsymbol{n}\cdot\boldsymbol{u}_{\mathrm{o},h},\sigma_h)_{\partial T_h}\right)\\&+\left(\phi S_{\mathrm{w},h}U_\mathrm{g}\frac{r_\mathrm{g}C_h}{Y_\mathrm{g}},\sigma_h\right)_\Omega\end{aligned}\tag{3.16}$$

with the already known transport over the cell boundaries

$$\left(C_{\mathrm{g},h}\boldsymbol{n}\cdot\boldsymbol{u}_{\mathrm{o},h},\sigma_h\right)_{\partial T_h}=\left(C^+_{\mathrm{g},h}(\boldsymbol{n}\cdot\boldsymbol{u}^+_{\mathrm{o},h}),\sigma_h\right)_{\partial T^+_{h,\mathrm{o}}}+\left(C^-_{\mathrm{g},h}(\boldsymbol{n}\cdot\boldsymbol{u}^-_{\mathrm{o},h}),\sigma_h\right)_{\partial T^-_{h,\mathrm{o}}}.\tag{3.17}$$

Note, that we use a slightly different definition of the inflow and the outflow boundary of the cells. Since all the gas is dissolved in the oleic phase, we use u_o instead of u_w to define $\partial T^-_{h,\mathrm{o}}$ and $\partial T^+_{h,\mathrm{o}}$, respectively.

For Equation (2.13), we simply repeat the steps that we have executed for Equation (2.11) to recover its weak formulation. We transform Equation (2.13) into a system of first order in C_m

$$\boldsymbol{q}_2=\sqrt{S_\mathrm{w}\boldsymbol{D}_\mathrm{m}}\cdot\nabla C_\mathrm{m},\tag{3.18}$$

$$\begin{aligned}\frac{\partial(\phi S_\mathrm{w}C_\mathrm{m})}{\partial t}=&\phi\nabla\cdot[C_\mathrm{m}\boldsymbol{D}_\mathrm{m}\cdot\nabla S_\mathrm{w}]+\phi\nabla\cdot\left[\sqrt{S_\mathrm{w}\boldsymbol{D}_\mathrm{m}}\cdot\boldsymbol{q}_2\right]\\&-\nabla\cdot(\boldsymbol{u}_\mathrm{w}C_\mathrm{m})-\phi S_\mathrm{w}U_\mathrm{m}\frac{r_\mathrm{g}C}{Y_\mathrm{m}}\end{aligned}\tag{3.19}$$

and obtain the weak formulation of (3.18)

$$\begin{aligned}(\boldsymbol{q}_{2,h},\boldsymbol{\eta}_h)_\Omega=&(\sqrt{S_{\mathrm{w},h}\boldsymbol{D}_\mathrm{m}}\cdot\nabla C_{\mathrm{m},h},\boldsymbol{\eta}_h)_\Omega\\=&\sum_{T_h\in\mathscr{T}}\Big(-\left(C_{\mathrm{m},h},\nabla\cdot\left[\sqrt{S_{\mathrm{w},h}\boldsymbol{D}_\mathrm{m}}\cdot\boldsymbol{\eta}_h\right]\right)_{T_h}\\&+\left(\boldsymbol{n}\cdot\left(\sqrt{S_{\mathrm{w},h}\boldsymbol{D}_\mathrm{m}}C_{\mathrm{m},h}\right)^*,\boldsymbol{\eta}_h\right)_{\partial T_h}\Big),\end{aligned}\tag{3.20}$$

where we consider the numerical flux

$$\left(\sqrt{S_{\mathrm{w},h}\boldsymbol{D}_\mathrm{m}}C_{\mathrm{m},h}\right)^*=\{\{\sqrt{S_{\mathrm{w},h}\boldsymbol{D}_\mathrm{m}}C_{\mathrm{m},h}\}\}+\tfrac{1}{2}\left[\!\left[\sqrt{S_{\mathrm{w},h}\boldsymbol{D}_\mathrm{m}}\right]\!\right]C^+_{\mathrm{m},h}.\tag{3.21}$$

The weak formulation of Equation (3.19) looks as follows

$$\begin{aligned}\left(\frac{\partial(\phi S_{w,h}C_{\mathrm{m},h})}{\partial t},\sigma_h\right) = &-\left(\phi C_{\mathrm{m},h}\boldsymbol{D}_\mathrm{m}\cdot\nabla S_{\mathrm{w},h},\nabla\sigma_h\right)_\Omega \\ &+\sum_{T_h\in\mathscr{T}}\Big(-\left(\phi\sqrt{S_{\mathrm{w},h}\boldsymbol{D}_\mathrm{m}}\cdot\boldsymbol{q}_{2,h},\nabla\sigma_h\right)_{T_h} \\ &\qquad+\left(\phi\boldsymbol{n}\cdot\left(\sqrt{S_{\mathrm{w},h}\boldsymbol{D}_\mathrm{m}}\cdot\boldsymbol{q}_{2,h}\right)^*,\sigma_h\right)_{\partial T_h}\Big) \\ &+\sum_{T_h\in\mathscr{T}}\left(\left(\boldsymbol{u}_{\mathrm{w},h}C_{\mathrm{m},h},\nabla\sigma_h\right)_{T_h}-\left(C_{\mathrm{m},h}\boldsymbol{n}\cdot\boldsymbol{u}_{\mathrm{w},h},\sigma_h\right)_{\partial T_h}\right) \\ &+\left(\phi S_{\mathrm{w},h}U_\mathrm{g}\frac{r_\mathrm{g}C_h}{Y_\mathrm{g}},\sigma_h\right)_\Omega .\end{aligned} \tag{3.22}$$

We choose the numerical flux according to

$$\left(\sqrt{S_{\mathrm{w},h}\boldsymbol{D}_\mathrm{m}}\cdot\boldsymbol{q}_{2,h}\right)^* = \left\{\!\!\left\{\sqrt{S_{\mathrm{w},h}\boldsymbol{D}_\mathrm{m}}\cdot\boldsymbol{q}_{2,h}\right\}\!\!\right\}+\tfrac{1}{2}\left[\!\left[\sqrt{S_{\mathrm{w},h}\boldsymbol{D}_\mathrm{m}}\right]\!\right]\cdot\boldsymbol{q}_{2,h}^+, \tag{3.23}$$

see Hesthaven and Warburton [50], and

$$\left(C_{\mathrm{m},h}\boldsymbol{n}\cdot\boldsymbol{u}_{\mathrm{w},h},\sigma_h\right)_{\partial T_h} = \left(C_{\mathrm{m},h}^+(\boldsymbol{n}\cdot\boldsymbol{u}_{\mathrm{w},h}^+),\sigma_h\right)_{\partial T_{h,w}^+}+\left(C_{\mathrm{m},h}^-(\boldsymbol{n}\cdot\boldsymbol{u}_{\mathrm{w},h}^-),\sigma_h\right)_{\partial T_{h,w}^-}, \tag{3.24}$$

with the established definitions of $\partial T_{h,w}^-$ and $\partial T_{h,w}^+$.

The model equations in the weak formulation can be summarized as the following problem: Find $(\Phi_{\mathrm{w},h},S_{\mathrm{w},h},C_h,C_{\mathrm{f},h},C_{\mathrm{g},h},C_{\mathrm{m},h})\in v^\partial+V_h\times U_h^5$, such that Equations (3.3), (3.4), (3.6), (3.13), (3.16), (3.22) are fulfilled with an arbitrary function $v^\partial\in H^1(\Omega)$. The trace of v^∂ on the boundary matches to the Dirichlet boundary values of the states Φ_w, S_w, C, C_f, C_g and C_m. With N and M the dimensions of V_h and U_h, respectively and the nodal bases $\{\phi_h^1,\dots,\phi_h^N\}$ and $\{\sigma_h^1,\dots,\sigma_h^M\}$ we represent the solutions as

$$\begin{aligned}\Phi_{\mathrm{w},h}&=\sum_{k=1}^N\Phi_{\mathrm{w},k}\phi_h^k, & S_{\mathrm{w},h}&=\sum_{k=1}^M S_{\mathrm{w},k}\sigma_h^k, & C_h&=\sum_{k=1}^M C_k\sigma_h^k\\ C_{\mathrm{f},h}&=\sum_{k=1}^M C_{\mathrm{f},k}\sigma_h^k, & C_{\mathrm{g},h}&=\sum_{k=1}^M C_{\mathrm{g},k}\sigma_h^k, & C_{\mathrm{m},h}&=\sum_{k=1}^M C_{\mathrm{m},k}\sigma_h^k.\end{aligned}$$

We apply these representations to the weak system and test with all basis functions ϕ_h^i, $i=1,\dots,M$, and σ_h^i, $i=1,\dots,M$. The differential operators ∇ and $\nabla\cdot$ act only on the space-dependent ansatz functions, for which the exact derivatives are known. We manipulate the system matrix and the right-hand side to incorporate the Dirichlet-type boundary conditions. The degrees of freedom on the boundary are removed and the solution is set to the predefined value. The inhomogeneous Neumann-type conditions are formulated directly in the weak representation while the homogeneous Neumann-type boundary conditions are imposed naturally via the weak formulation.

3.2 The classical IMPES method

The considered differential equations for the black oil model (2.6) are strongly coupled and nonlinear. There is a widespread selection of approaches to solve these problems, such as *simultaneous solution* (SS) techniques by Douglas et al. [36], *sequential implicit methods* by MacDonald and Coats [69] and *adaptive implicit methods* by Thomas and Thurnau [93]. In the following, we focus on the classical *IMplicit Pressure Explicit Saturation* (IMPES) method, which is a very powerful method for the numerical treatment of incompressible two phase flow. It was first proposed by Sheldon et al. [88] in 1959 and Stone and Garder jr. [91] in 1961, and it is still the most popular algorithm in the petroleum industry.

We state the approach as it was presented in Chen [24]. The basic idea of the IMPES method is the separation of the computation of the pressure from that of the saturation. First, Equation (2.6a) is solved for the pressure using an implicit equation, then we apply an explicit time stepping scheme to compute the saturation from (2.6b). The resulting scheme is rather easy to implement and requires less memory than other methods such as the SS method. One drawback of this method is that the classical IMPES methods require small time steps for the saturation equation in order to be stable. Especially for long time horizons and fine meshes this restriction is expensive and prohibitive.

The algorithm can be written as follows:

Algorithm 3.2.1. ***The classical IMPES method***

1. *Set* $k := 0$ *and set* $S_{\mathrm{w}}^k = S^0$
2. *Solve for* $p_{\mathrm{w},h}^{k+1}$
$$0 = \left(\lambda_{\mathrm{t}}(S_{\mathrm{w},h}^k)\boldsymbol{K}\cdot\nabla\Phi_{\mathrm{w},h}^{k+1},\nabla\varphi_h\right)_\Omega + \left(\lambda_{\mathrm{n}}(S_{\mathrm{w},h}^k)\boldsymbol{K}\cdot\nabla\Phi_{\mathrm{c}}^k,\nabla\varphi_h\right)_\Omega + (q^{\mathrm{in}},\varphi_h)_{\Gamma_{\mathrm{in}}}.$$
3. *Update*
$$\boldsymbol{u}_{a,h}^{k+1} = -\lambda_{\mathrm{t}}(S_{\mathrm{w},h}^k)\boldsymbol{K}\cdot\nabla\Phi_{\mathrm{w},h}^{k+1}.$$
4. *Choose a time step* Δt^{k+1}.
5. *Solve for* $S_{\mathrm{w},h}^{k+1}$
$$\begin{aligned}\left(\phi S_{\mathrm{w},h}^{k+1},\sigma_h\right)_\Omega =& \Delta t^{k+1}\sum_{T_h\in\mathscr{T}}\Bigg(\left(f_{\mathrm{w}}(S_{\mathrm{w},h}^k)\boldsymbol{u}_{a,h}^{k+1},\nabla\sigma_h\right)_\Omega \\ &\qquad - \left(f_{\mathrm{w}}\left((S_{\mathrm{w},h}^k)^+\right)\boldsymbol{n}\cdot(\boldsymbol{u}_{a,h}^{k+1})^+,\sigma_h\right)_{\partial T_{h,a}^+} \\ &\qquad - \left(f_{\mathrm{w}}\left((S_{\mathrm{w},h}^k)^-\right)\boldsymbol{n}\cdot(\boldsymbol{u}_{a,h}^{k+1})^-,\sigma_h\right)_{\partial T_{h,a}^-}\Bigg) \\ &+ \left(\phi S_{\mathrm{w},h}^k,\sigma_h\right)_\Omega\end{aligned}$$

6. *If*

$$t^{k+1} = t^k + \Delta t^{k+1} \geqslant t_{end}$$

stop. Otherwise:

7. *Set* $k := k+1$. *Go to 2.*

To solve the equations in Step 2 and 5 of Algorithm 3.2.1 we apply a conjugate gradient (CG) method that is implemented in the SolverCG class of *deal.ii* without preconditioning. For more information about CG methods we refer to Nocedal and Wright [77]. In step 2, we solve the implicit pressure equation which depends only on the water saturation S_{w}^k that was computed in the previous step. Then, we apply an explicit time step to determine S_{w}^{k+1} which depends only on the old saturation S_{w}^k and the just updated $\boldsymbol{u}_{\mathrm{a}}^{n+1}$. By using the IMPES method, we never have to iterate for the nonlinearities of the system as we would have if we use a fully implicit method.

3.3 Choosing a time step

For the choice of the time step, we follow Li and Bangerth [65]. In 1928, Richard Courant, Kurt Friedrichs, and Hans Lewy stated a condition for the numerical stability of the solution of hyperbolic partial differential equations, see Courant et al. [29]. The Courant–Friedrichs–Lewy (CFL) number

$$c = \frac{u\Delta t}{h} \tag{3.25}$$

describes how many cells are crossed by an observed quantity in one time step. Here, h denotes the diameter of a cell, u refers to the velocity of the particle and Δt is the length of the time step.

If one uses an explicit time stepping scheme for hyperbolic transport equations, as we do, when we solve the saturation equation (3.4), a general rule of thumb is to choose the time step Δt^{k+1} in a manner that the particles can not travel a distance larger than the diameter of a single cell within one time step. This corresponds to a CFL number smaller or equal to 1. Coats [25, 26] showed that the choice of

$$c \leqslant 1$$

ensures non-oscillatory stability for the solution. In other words, here, we should choose

$$\Delta t^{k+1} \leqslant \frac{h}{\|\boldsymbol{u}_{\mathrm{a}}^{k+1}(\boldsymbol{x})\|_2}.$$

Fortunately, this quantity is rather easy to calculate in the current setting. If we recall Algorithm 3.2.1, we update Δt^{k+1} in Step 4 right after we have updated $\boldsymbol{u}_{\mathrm{a}}^{k+1}$. We loop over all quadrature points of the domain and determine the maximal norm of the velocity. Then we set the time step to

$$\Delta t^{k+1} := \frac{\min\limits_{T_h \in \mathscr{T}} h_T}{\max\limits_{\boldsymbol{x}} \|\boldsymbol{u}_{\mathrm{a}}^{k+1}(\boldsymbol{x})\|_2}. \tag{3.26}$$

It can easily be shown that larger time steps lead to grid points where the water saturation is larger than one or less than zero. Since the saturation corresponds to the fraction of water that fills the pore space our model would not be evaluable anymore. If we use a time step according to (3.26), this should not happen anyhow. To be on the safe side, we run a routine called *project_back_saturation*, see Li and Bangerth [65], at the end of each time step. As the name suggests it projects the water saturation back onto the interval $[S_{\mathrm{wc}}, 1 - S_{\mathrm{or}}]$ if necessary. We have to do so since the functions (2.3b), (2.3a) for the relative permeabilities and the fractional flow $f_{\mathrm{w}}(S_{\mathrm{w}})$ are not evaluable outside this range.

3.4 The extended IMPES method

We still have to specify how to solve the equations for bacteria (3.6), nutrients (3.13), gas (3.16) and metabolites (3.22). The considered equations are dominated by the convection term. Hence, it is reasonable to use an explicit time stepping scheme for (3.6), (3.13), (3.16) and (3.22) as well. We have to pay attention to the coupled time derivative of the saturation and the respective states on the left hand side of all of these equations. To overcome this issue, we extend the IMPES method by inserting the updated saturations S_{w}^{n+1} and $S_{\mathrm{o}}^{n+1} = 1 - S_{\mathrm{w}}^{n+1}$, that we have computed in Step 5 of Algorithm 3.2.1, into Equations (3.6), (3.13), (3.16) and (3.22), see Berenblyum [9] and Nielsen [76]. This approach is sometimes also referred to as *IMPEC* method where the capital letter C stands for components. The computation of $C_h^{k+1}, C_{\mathrm{f},h}^{k+1}, C_{\mathrm{g},h}^{k+1}$, and $C_{\mathrm{m},h}^{k+1}$, is then added to Algorithm 3.2.1 after Step 5 has been executed.

Algorithm 3.4.1. ***Extension of the classical IMPES***

5.1 Solve for C_h^{k+1}

$$\begin{aligned}
\left(\phi S_{\mathrm{w},h}^{k+1} C_h^{k+1}, \sigma_h\right)_\Omega = & \Delta t^{k+1} \sum_{T_h \in \mathscr{T}} \Bigg(\left(\boldsymbol{u}_{\mathrm{w},h}^{k+1} C_h^k, \nabla \sigma_h\right)_\Omega \\
& \qquad - \left((C_h^k)^+ \boldsymbol{n} \cdot (\boldsymbol{u}_{\mathrm{w},h}^{k+1})^+, \sigma_h\right)_{\partial T_{h,w}^+} \\
& \qquad - \left((C_h^k)^- \boldsymbol{n} \cdot (\boldsymbol{u}_{\mathrm{w},h}^{k+1})^-, \sigma_h\right)_{\partial T_{h,w}^-} \Bigg) \\
& + \Delta t^{k+1} \left(\phi S_{\mathrm{w},h}^{k+1} [r_{\mathrm{g}} - r_{\mathrm{d}}] C_h^k, \sigma_h\right)_\Omega - \Delta t^{k+1} \left(k_{\mathrm{c}} \phi S_{\mathrm{w},h}^{k+1} C_h^k, \sigma_h\right)_\Omega \\
& + \left(\phi S_{\mathrm{w},h}^{k+1} C_h^k, \sigma_h\right)_\Omega
\end{aligned}$$

5.2 *Solve for* $\boldsymbol{q}_{1,h}^{k+1}$

$$
\begin{aligned}
(\boldsymbol{q}_{1,h}^{k+1}, \boldsymbol{\eta}_h)_\Omega = \sum_{T_h \in \mathscr{T}} \Bigg(& - \left(C_{\mathrm{f},h}^k, \nabla \cdot \left[\sqrt{S_{\mathrm{w},h}^{k+1} \boldsymbol{D}_\mathrm{f}} \cdot \boldsymbol{\eta}_h \right] \right)_{T_h} \\
& + \left(\boldsymbol{n} \cdot \left(\sqrt{S_{\mathrm{w},h}^{k+1} \boldsymbol{D}_\mathrm{f} C_{\mathrm{f},h}^k} \right)^*, \boldsymbol{\eta}_h \right)_{\partial T_h} \Bigg).
\end{aligned}
$$

5.3 *Solve for* $C_{\mathrm{f},h}^{k+1}$

$$
\begin{aligned}
\left(\phi S_{\mathrm{w},h}^{k+1} C_{\mathrm{f},h}^{k+1}, \sigma_h \right) = & - \Delta t^{k+1} (\phi C_{\mathrm{f},h}^k \boldsymbol{D}_{\mathrm{f},h} \cdot \nabla S_{\mathrm{w},h}^{k+1}, \nabla \sigma_h)_\Omega \\
& + \Delta t^{k+1} \sum_{T_h \in \mathscr{T}} \Bigg(- \left(\phi \sqrt{S_{\mathrm{w},h}^{k+1} \boldsymbol{D}_\mathrm{f}} \cdot \boldsymbol{q}_{1,h}, \nabla \sigma_h \right)_{T_h} \\
& \qquad + \left(\phi \boldsymbol{n} \cdot \left(\sqrt{S_{\mathrm{w},h}^{k+1} \boldsymbol{D}_\mathrm{f}} \cdot \boldsymbol{q}_{1,h} \right)^*, \sigma_h \right)_{\partial T_h} \Bigg) \\
& + \Delta t^{k+1} \sum_{T_h \in \mathscr{T}} \Bigg(\left(\boldsymbol{u}_{\mathrm{w},h}^{k+1} C_{\mathrm{f},h}^k, \nabla \sigma_h \right)_\Omega \\
& \qquad - \left((C_{\mathrm{f},h}^k)^+ \boldsymbol{n} \cdot (\boldsymbol{u}_{\mathrm{w},h}^{k+1})^+, \sigma_h \right)_{\partial T_{h,w}^+} \\
& \qquad - \left((C_{\mathrm{f},h}^k)^- \boldsymbol{n} \cdot (\boldsymbol{u}_{\mathrm{w},h}^{k+1})^-, \sigma_h \right)_{\partial T_{h,w}^-} \Bigg) \\
& - \Delta t^{k+1} \left(\phi S_{\mathrm{w},h}^{k+1} U_\mathrm{f} r_\mathrm{g} C_h^k, \sigma_h \right)_\Omega \\
& + \left(\phi S_{\mathrm{w},h}^{k+1} C_{\mathrm{f},h}^k, \sigma_h \right)_\Omega
\end{aligned}
$$

5.4 *Solve for* $C_{\mathrm{g},h}^{k+1}$

$$
\begin{aligned}
\left(\phi S_{\mathrm{o},h}^{k+1} C_{\mathrm{g},h}^{k+1}, \sigma_h \right)_\Omega = & \Delta t^{k+1} \sum_{T_h \in \mathscr{T}} \Bigg(\left(\boldsymbol{u}_{\mathrm{o},h}^{k+1} C_{\mathrm{g},h}^k, \nabla \sigma_h \right)_\Omega \\
& \qquad - \left((C_{\mathrm{g},h}^k)^+ \boldsymbol{n} \cdot (\boldsymbol{u}_{\mathrm{o},h}^{k+1})^+, \sigma_h \right)_{\partial T_{h,o}^+} \\
& \qquad - \left((C_{\mathrm{g},h}^k)^- \boldsymbol{n} \cdot (\boldsymbol{u}_{\mathrm{o},h}^{k+1})^-, \sigma_h \right)_{\partial T_{h,o}^-} \Bigg) \\
& + \Delta t^{k+1} \left(\phi S_{\mathrm{w},h}^{k+1} U_\mathrm{g} \frac{r_\mathrm{g} C_h^k}{Y_\mathrm{g}}, \sigma_h \right)_\Omega \\
& + \left(\phi S_{\mathrm{o},h}^{k+1} C_{\mathrm{g},h}^k, \sigma_h \right)_\Omega
\end{aligned}
$$

5.5 Solve for $\boldsymbol{q}_{2,h}^{k+1}$

$$(\boldsymbol{q}_{2,h}^{k+1},\boldsymbol{\eta}_h)_\Omega = \sum_{T_h\in\mathscr{T}}\left(-\left(C_{\mathrm{m},h}^k,\nabla\cdot\left[\sqrt{S_{\mathrm{w},h}^{k+1}\boldsymbol{D}_\mathrm{m}}\cdot\boldsymbol{\eta}_h\right]\right)_{T_h} + \left(\boldsymbol{n}\cdot\left(\sqrt{S_{\mathrm{w},h}^{k+1}\boldsymbol{D}_\mathrm{m}}C_{\mathrm{m},h}^k\right)^*,\boldsymbol{\eta}_h\right)_{\partial T_h}\right).$$

5.6 Solve $C_{\mathrm{m},h}^{k+1}$

$$\begin{aligned}\left(\phi S_{\mathrm{w},h}^{k+1}C_{\mathrm{m},h}^{k+1},\sigma_h\right) =& -\Delta t^{k+1}(\phi C_{\mathrm{m},h}^k\boldsymbol{D}_\mathrm{m}\cdot\nabla S_{\mathrm{w},h}^{k+1},\nabla\sigma_h)_\Omega \\ &+\Delta t^{k+1}\sum_{T_h\in\mathscr{T}}\left(-\left(\phi\sqrt{S_{\mathrm{w},h}^{k+1}\boldsymbol{D}_\mathrm{m}}\cdot\boldsymbol{q}_{2,h},\nabla\sigma_h\right)_{T_h}\right. \\ &\qquad\left.+\left(\phi\boldsymbol{n}\cdot\left(\sqrt{S_{\mathrm{w},h}^{k+1}\boldsymbol{D}_\mathrm{m}}\cdot\boldsymbol{q}_{2,h}\right)^*,\sigma_h\right)_{\partial T_h}\right) \\ &+\Delta t^{k+1}\sum_{T_h\in\mathscr{T}}\left(\left(\boldsymbol{u}_{\mathrm{w},h}^{k+1}C_{\mathrm{m},h}^k,\nabla\sigma_h\right)_\Omega\right. \\ &\qquad-\left(\left(C_{\mathrm{m},h}^k\right)^+\boldsymbol{n}\cdot(\boldsymbol{u}_{\mathrm{w},h}^{k+1})^+,\sigma_h\right)_{\partial T_{h,w}^+} \\ &\qquad\left.-\left(\left(C_{\mathrm{m},h}^k\right)^-\boldsymbol{n}\cdot(\boldsymbol{u}_{\mathrm{w},h}^{k+1})^-,\sigma_h\right)_{\partial T_{h,w}^-}\right) \\ &+\Delta t^{k+1}\left(\phi S_{\mathrm{w},h}^{k+1}U_\mathrm{g}\frac{r_\mathrm{g}C_h^k}{Y_\mathrm{g}},\sigma_h\right)_\Omega \\ &+\left(\phi S_{\mathrm{w},h}^{k+1}C_{\mathrm{m},h}^k,\sigma_h\right)_\Omega\end{aligned}$$

Note, in Equations (2.11) and (2.13) we consider diffusion. In LeVeque [64], it is shown that for explicit time stepping schemes the time step has to respect the following constraint

$$\Delta t \leqslant \frac{h^2}{2D},$$

where D denotes the diffusion constant. In our setting we have to use $D = \max\{D_\mathrm{f}, D_\mathrm{m}\}$. As mentioned in section 2.3.2, we consider only convection-dominated processes with a high Péclet number. Therefore, we neglect the constraint.

We use the same CG algorithm to compute the solutions of Step 5.1–5.6 in Algorithm 3.4.1 as we did in Step 2 and 5 of Algorithm 3.2.1.

3.5 Numerical simulation of microbial transport - 2D example

We conclude this chapter with a simulation of a 2D example to illustrate that our model is appropriate to describe the underlying processes. Due to the rotational symmetry and homogeneity of the core and the boundary and initial condition, we are actually concerned with a 1D model. But to show that the methods are independent of the dimension, we execute the following computations in 2D. We assume that there is no capillary pressure.

We consider a rectangular domain $\Omega = [0,5] \times [0,31]$ with $\Gamma_{\text{in}} = [0,5] \times 0$, $\Gamma_{\text{out}} = [0,5] \times 31$ and $\Gamma_{\text{en}} = \partial\Omega \backslash \{\Gamma_{\text{in}} \cup \Gamma_{\text{out}}\}$. The time horizon is set to $t_{\text{end}} = 240$. We consider minutes and centimeter as units for time and space, respectively.

We consider one control variable, the injection velocity $q^{\text{in}} = -0.212758715 (\approx 100\frac{ml}{h})$, that is constant in time and two control functions, namely the injection concentration of bacteria C^{in} and nutrients C_{f}^{in}. We start by injecting pure water without any bacteria or nutrients. At $t = 135$, we add bacteria to the injected water until we start the first nutrient injection at $t = 165$. The second nutrient injection is started at $t = 190$ and ends at $t = 215$. It follows a final water flood. The sequences for the three different controls are listed below:

$$q^{\text{in}} = -0.212758715, \ t \in [0,240]$$

$$C^{\text{in}} = \begin{cases} 0.0, & t \in [0,135], \\ 0.01, & t \in (135,165], \\ 0.0, & t \in (165,240], \end{cases}$$

$$C_{\text{f}}^{\text{in}} = \begin{cases} 0.0, & t \in [0,165], \\ 0.08, & t \in (165,190], \\ 0.05, & t \in (190,215], \\ 0.0, & t \in (215,240], \end{cases}$$

The initial water saturation is set to $S^0 = 0.079$ and the initial concentration of the bacteria, nutrients, gas and metabolites are set to zero. The injected water saturation is set to $S_{\text{w}}^{\text{in}} = 0.95$ and the pressure at the outflow boundary is equal standard pressure. We simulate our model equations with the parameter settings listed in Table 3.1. We use a uniformly refined grid of 4096 cells and 41089 degrees of freedom. The results at $t \in \{0, 135, 165, 190, 215, 240\}$ are shown in Figures 3.1–3.6. We do not show the results for the axillary states q_1 and q_2. The order of the subfigures is the same for all figures. From the top left to the bottom right, we show the pressure of water, the water saturation and the concentrations of bacteria, nutrients, gas and metabolites.

Figure 3.1 illustrates the initial values of the water saturation, the four concentrations and the computed pressure profile at $t = 0$, cf. step 2 of Algorithm 3.2.1.

The states at $t = 135$, right before the start of the bacteria injection, are shown in Figure 3.2. There are no bacteria, nutrients etc. present. The pressure profile is flatter than in

$\hat{k}_{\text{ro}}$	$\hat{k}_{\text{rw}}$	c_{o}	c_{w}	$\hat{\mu}_{\text{o}}$	$\hat{\mu}_{\text{w}}$	μ_{g}	μ_{m}	K	S_{wc}	S_{or}
1.18	0.39	2.0	1.6	42.17	1.18	0.02	20.10	10.47	0.037	0.05

ϕ	C_{max}	K_{S}	μ_{max}	r_{d}	k_{c}	u_f	u_g	u_m	Y_g	Y_m
0.399	0.05	0.05	8.3	0.25	0.0	0.5	0.45	0.05	3.0	1.0

Table 3.1: Parameter settings

the beginning which leads to a slower replacement of the oil. We observe a high water saturation at the inflow boundary which decreases in direction of the outflow boundary.

In Figure 3.3, we depict the primary states of the model at $t = 165$ when we stop the injection of the bacteria. Since the bacteria are not provided with nutrients, no gas and metabolites have been produced. The pressure profile is even flatter than before and the water saturation has increased. The concentration of bacteria is comparatively high at the inflow boundary and lessen in direction of the outflow boundary.

After the first nutrient injection at $t = 190$, we have a uniform concentration of bacteria with a minimum at the outflow boundary. The nutrients have migrated into the core. Due to the presence of gas and metabolites, the pressure profile is stepper and more oil has been replaced. The concentration of gas is low with a peak at the inflow boundary. The metabolite concentration has its maximum to the right over the middle of the core because of convection effects.

Figure 3.5 illustrates the states at the end of the second nutrient injection $t = 215$. The pressure profile is nearly unchanged. More oil has been replaced. The core is almost evenly filled with the maximal concentration of bacteria. Especially in the middle of the core, most of the nutrients have been consumed. The concentration of gas has slightly risen but its peak has become higher. Since the oil concentration has reached the residual oil saturation at the bottom half of the core, there are no convection effects to observe. We notice a medium concentration of metabolites at the inflow boundary which increases in direction of the outflow boundary.

In Figure 3.6, we present the results of the simulation at the end of the time horizon. Bacteria, nutrients and metabolites have been flushed out of the bottom half of the core. The pressure profile is flatter than in Figure 3.5 due to the absence of metabolites at the bottom half. The gas concentration is almost unchanged.

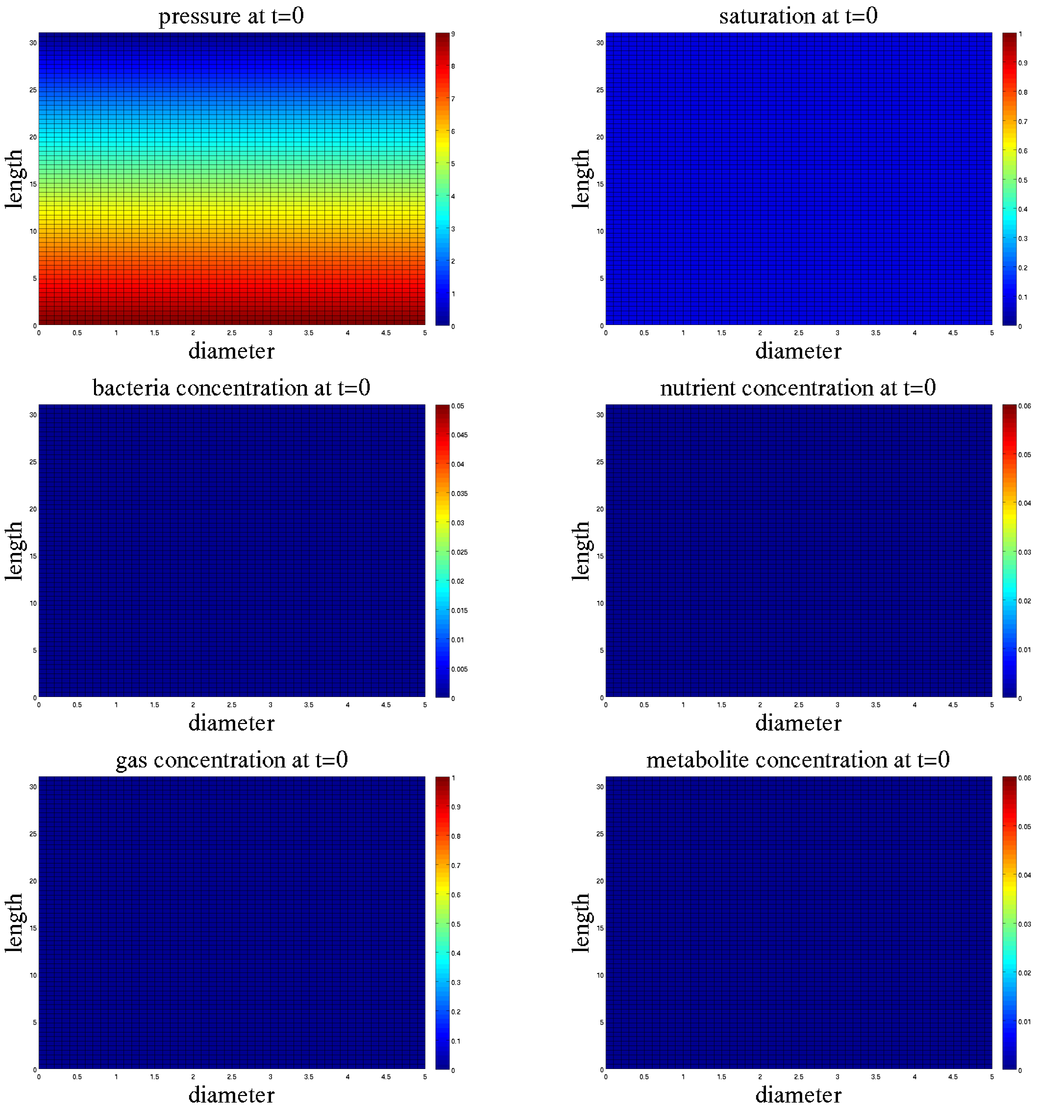
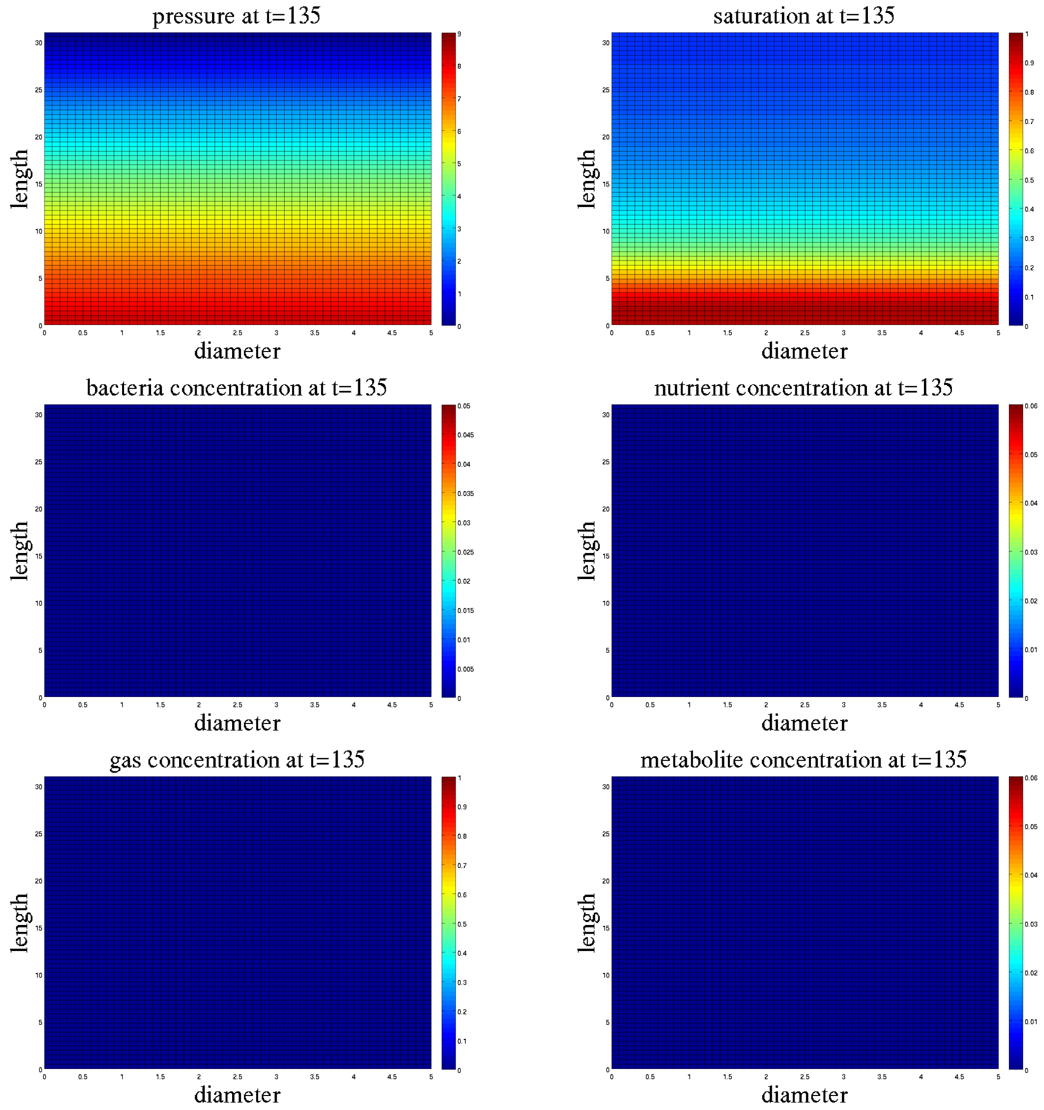


Figure 3.1: States at $t = 0$. Computed initial pressure and initial values for saturation, bacteria, nutrient, gas and metabolite concentration.


pressure at t=135

saturation at t=135

bacteria concentration at t=135

nutrient concentration at t=135

gas concentration at t=135

metabolite concentration at t=135

Figure 3.2: States at $t = 135$. Pressure profile flatter. Water saturation rises from the inflow boundary.

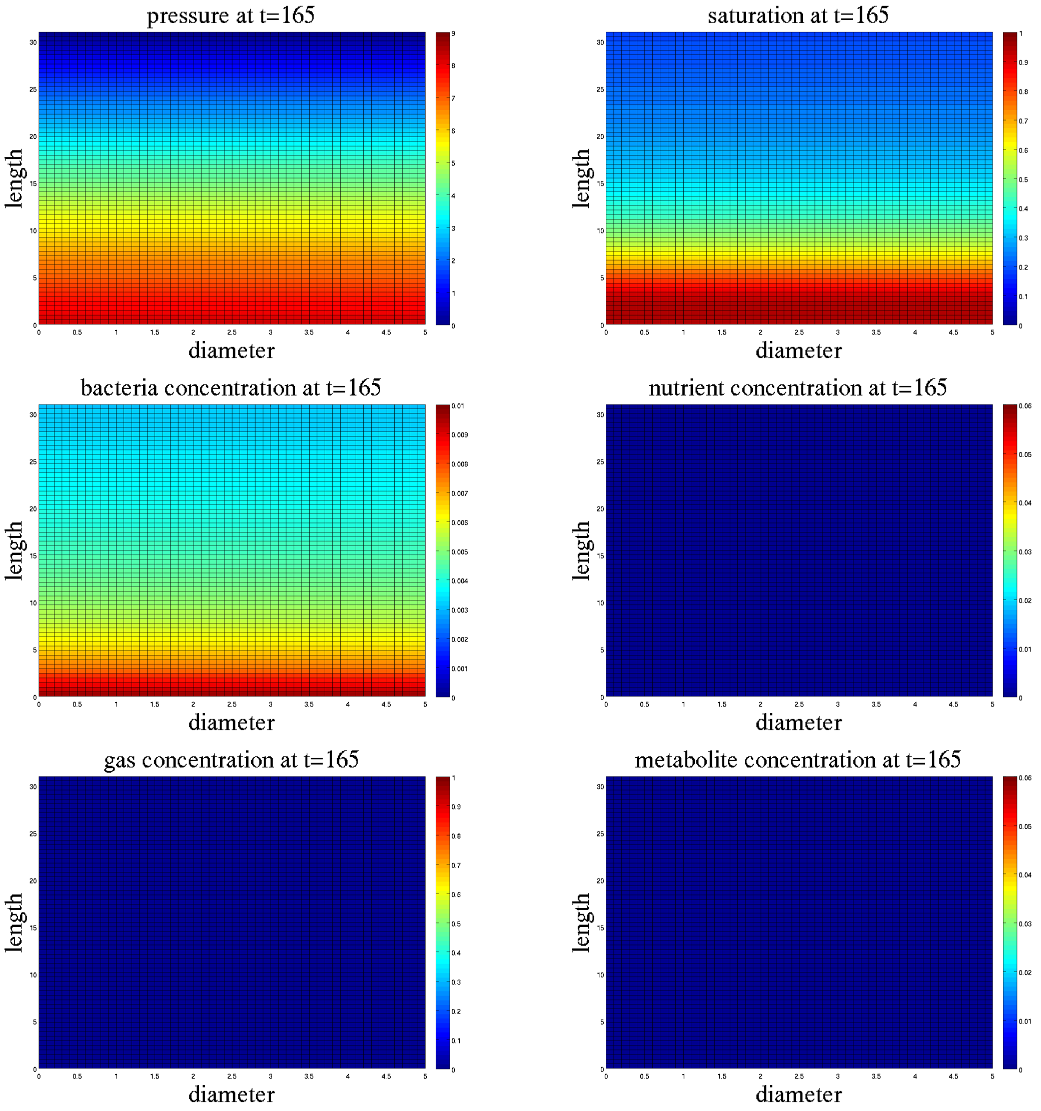


Figure 3.3: States at $t = 165$. Bacteria have been injected into the core.

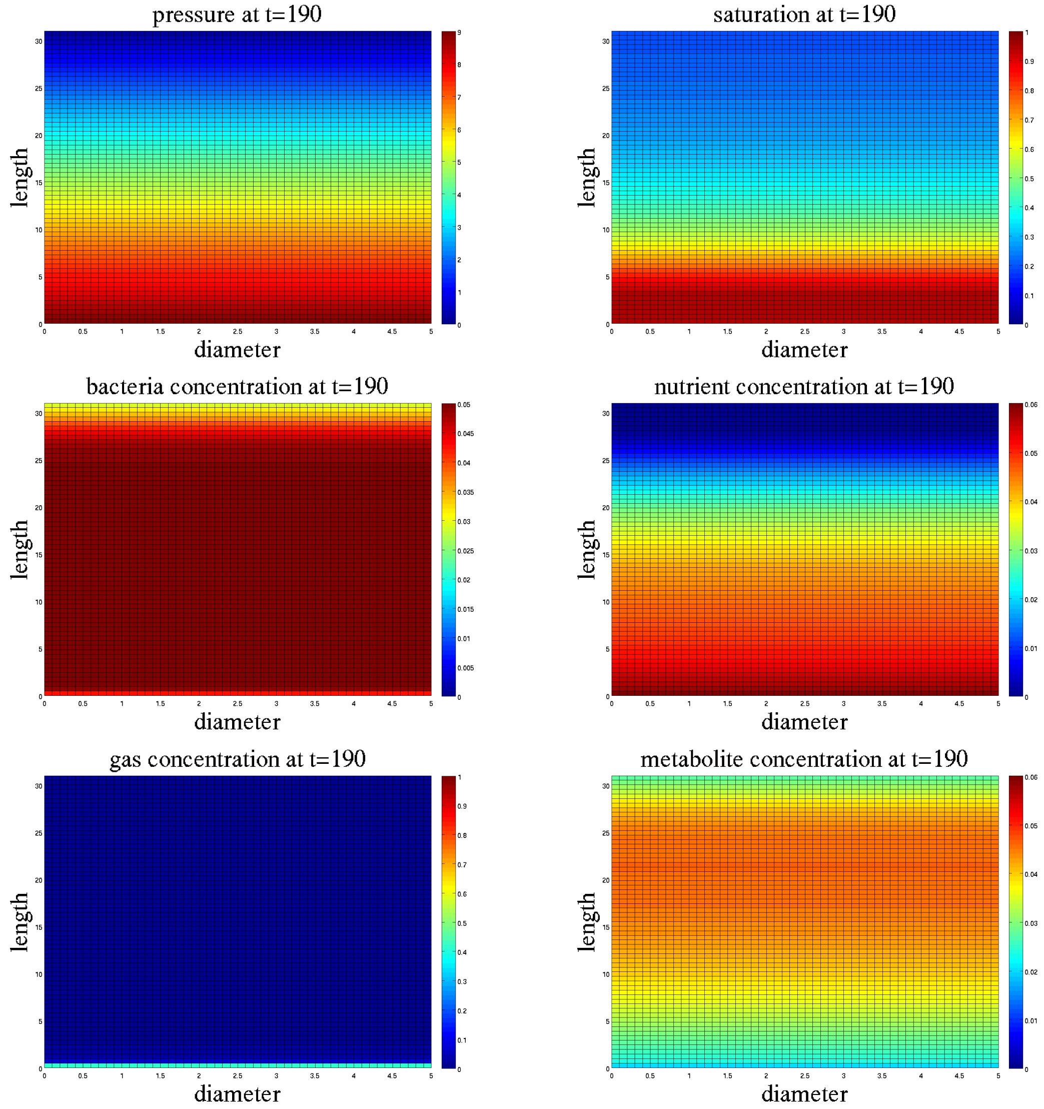


Figure 3.4: States at $t = 190$. End of the first nutrient injection. Pressure profile is steeper. High gas concentration at the inflow boundary (low oil saturation). Metabolites have been transported to the right half of the core.

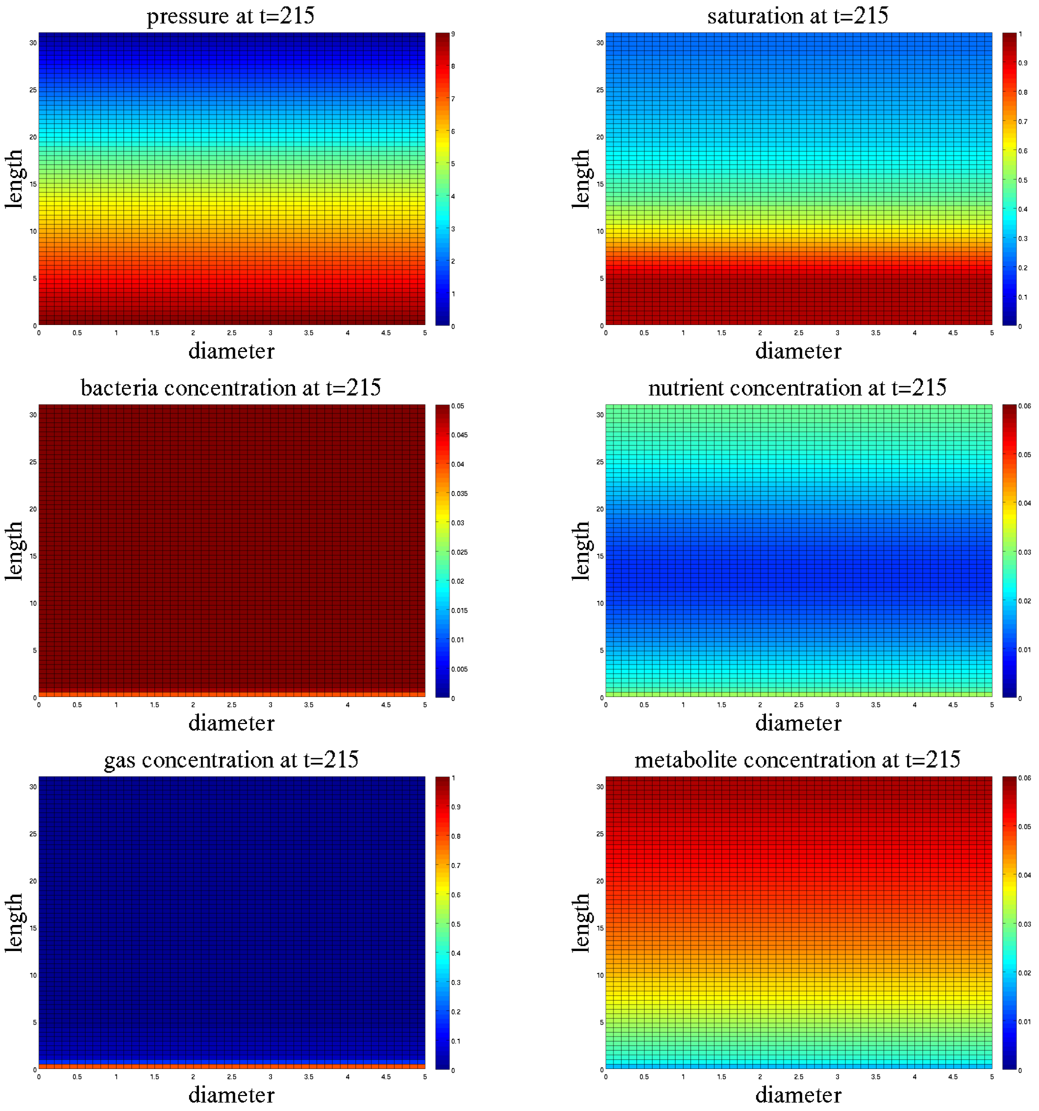


Figure 3.5: States at $t = 215$. Beginning of the terminal water flood. Bacteria fill the core. Nutrient almost consumed in the middle. High metabolite concentration.

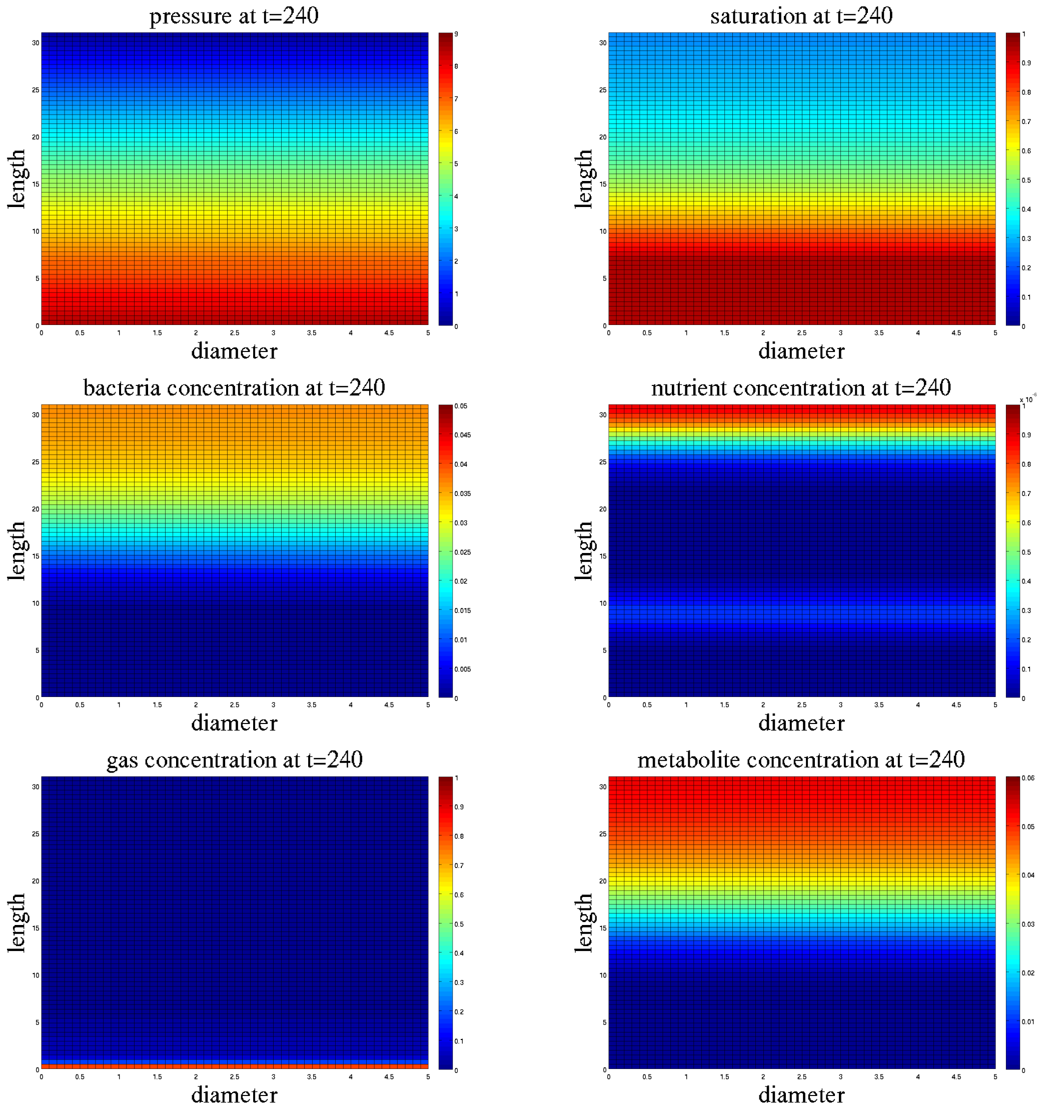


Figure 3.6: States at $t = 240$. End of the time horizon. Bacteria, nutrient and metabolites have been flushed out of the left side of the core or have been consumed.

4 Parameter estimation

Models that describe processes, e.g., in chemistry or physics, like the one for microbial enhanced oil recovery presented in Chapter 2 usually contain quantities whose values are not known exactly. We refer to those quantities as parameters. In order to predict or to optimize process behavior, an exact knowledge of the main parameters of the system is desirable. While the parameters can not be measured directly in most cases, it is possible to observe other quantities of the system. By fitting model to measurement data the parameters can be estimated.

In this chapter, we formulate a mathematical problem whose solution is a good approximation of the true parameters with respect to a specific criterion. First, we introduce the dynamical model and the measurement model for a considered process, and derive the parameter estimation problem. We explain how the problem is parametrized and state the generalized Gauss–Newton method as well as a local convergence theorem. We analyze the solution in a statistical sense in anticipation of optimum experimental design and motivate the use of a damping strategy. The chapter is concluded by a short introduction to parameter estimation for microbial enhanced oil recovery.

In 1987, Bock [13] presented a method to solve parameter estimation problems constrained by systems of ordinary differential equations (ODE) by a boundary value problem optimization approach. An extension to differential algebraic equations was presented by Bock et al. [18]. For a detailed overview, we refer to Körkel [58]. In this chapter, we will follow these methods.

Here and in the following, we omit the highlighting of vector-valued variables because it does not help to increase comprehensibility and readability.

4.1 Constrained parameter estimation problems

In this section, we derive the constrained parameter estimation problem. We refer to Seber and Wild [87] for the underlying statistical principles. Similar approaches can be found in Bock [13], Körkel [58], Bock et al. [15], Weiler [98] and Lenz [62].

4.1.1 Differential algebraic equations

We consider processes described by a system of linear-implicit DAEs

$$A(t,y(t),z(t),\theta,q,u(t))\dot{y}(t) = f(t,y(t),z(t),\theta), \tag{4.1a}$$
$$0 = g(t,y(t),z(t),\theta,). \tag{4.1b}$$

Herein f and g are arbitrary functions of time $t \in [t_0, t_{\text{end}}]$, dependent variables $y(t) \in \mathbb{R}^{n_y}$, which are defined by the ordinary differential equations (4.1a), dependent variables $z(t) \in \mathbb{R}^{n_z}$, which are determined by the algebraic equations (4.1b), and parameters $\theta \in \mathbb{R}^{n_\theta}$. Please note that system (4.1) may be obtained from partial differential equations discretized by a method of lines, see Schiesser [83], and Deuflhard and Weiser [33].

Usually, we consider initial value problems. The initial conditions often depend on the parameters:

$$y(t_0) = y_0(\theta). \tag{4.2}$$

The initial values of the algebraic states may be chosen in a way such that we have consistency

$$g(t_0, y(t_0), z(t_0), \theta) = 0$$

for the algebraic equations (4.1b) at $t = t_0$.

For the solvability of a DAE, the *differentiation index* is a crucial criterion, see Hairer and Wanner [48]. We evaluate the derivative of the algebraic equations (4.1b) of a linear-implicit DAE with respect to t and obtain

$$0 = g_t(t,y(t),z(t),\theta) + g_y(t,y(t),z(t),\theta)\dot{y}(t) + g_z(t,y(t),z(t),\theta)\dot{z}(t).$$

If g_z is regular, we perform algebraic manipulations with respect to $\dot{z}$

$$\dot{z}(t) = g_z(t,y(t),z(t),\theta)^{-1}\left(g_t(t,y(t),z(t),\theta) + g_y(t,y(t),z(t),\theta)\dot{y}(t)\right).$$

With (4.1a), we get an ODE system for $\dot{y}$ and $\dot{z}$. In that case, we say that the DAE has (differentiation) index 1. In general, the differentiation index of an DAE corresponds to the minimal number of differentiations with respect to t that have to be executed to transform a DAE into an ODE.

Definition 4.1.1. *Equation* (4.1) *has differentiation index m if m is the minimal number of analytical differentiations*

$$g = 0,\ \frac{\partial g}{\partial t} = 0,\ \ldots,\ \frac{d^m g}{dt^m} = 0 \tag{4.3}$$

such that Equations (4.3) *allow us to extract by algebraic manipulations an explicit ordinary differential system.*

Consequently, an ODE always has index 0. We assume that the considered differential algebraic equations (4.1) have differentiation index 1.

Additionally to the model equations 4.1, interior point constraints have to be fulfilled at m_r time points. We refer to these constraints in the form:

$$0 = \sum_{k=1}^{m_r} r_k \left(y(t_k), z(t_k), \theta\right), \tag{4.4}$$

with $r : I\!R^{n_y+n_z} \times I\!R^{n_\theta} \rightarrow I\!R^{n_r}$. In the following, we assume that the interior point constraints (4.4) include the initial conditions for the differential states(4.2).

4.1.2 Model responses and measurements

As mentioned above, usually a dynamic model contains parameters, whose true values θ^* are unknown, but there are measurement instruments that allow us to determine other observables of the system. Applying these methods to the process, we obtain a vector of measurements $\eta_j,\ j = 1,\dots,m$. We refer to a function h_j that describes the measurement process as the model response and we assume

$$\eta_j = h_j(t_j, y^*(t_j), z^*(t_j), \theta^*) + \varepsilon_j, \quad j = 1,\dots,m. \tag{4.5}$$

Here, y^* and z^* refer to the states computed by evaluating the model equations (4.1) for the true parameters θ^*. We assume that the model responses h_j describe the measurement process correctly. Then, ε_j is the measurement error for the j-th measurement.

The measurement errors are random variables, i.e., for two sets of measurement values $\left(\eta_j\right)_{j=1,\dots,m}$ and $\left(\hat{\eta}_j\right)_{j=1,\dots,m}$ that have been observed for two identical experiments executed in exactly the same way, we have almost surely

$$\varepsilon_j \neq \hat{\varepsilon}_j, \quad j = 1,\dots,m,$$

if ε_j and $\hat{\varepsilon}_j$ denote the corresponding measurement errors. Thus, it holds $\eta_j \neq \hat{\eta}_j,\ j = 1,\dots,m$. We assume the measurement errors to be additive, independent and normally distributed

$$\varepsilon_j \sim \mathcal{N}\left(0, \sigma_j^2\right),\ j = 1,\dots,m, \tag{4.6}$$

with mean 0 and known variance σ_j^2. Following the argumentation of Bard [6], we define the joint probability density function

$$f_\theta(\varepsilon) = (\sqrt{2\pi})^{-m} \prod_{j=1}^{m} \sigma_j^{-1} \exp\left(-\frac{\varepsilon_j^2}{2\sigma_j^2}\right).$$

For any (feasible) vector of parameters θ, it is possible to define the residuals

$$R(\theta, \eta_j) := \eta_j - h(t_j, y(t_j), z(t_j), \theta), \quad j = 1,\dots,m. \tag{4.7}$$

From Equation (4.5), it follows

$$R(\theta^*, \eta_j) = \varepsilon_j, \; j = 1, \dots, m.$$

We apply the definition of the residuals (4.7) to define the likelihood function. If the measurement errors are distributed as described above, the likelihood function is defined as

$$L(\theta|\eta) := (\sqrt{2\pi})^{-m} \prod_{j=1}^{m} \sigma_j^{-1} \exp\left(-\frac{\left(\eta_j - h_j(t_j, y(t_j), z(t_j), \theta)\right)^2}{2\sigma_j^2} \right) \tag{4.8}$$

with $\eta = (\eta_1, \dots, \eta_m)^T$.

4.1.3 Maximum likelihood estimation

The argumentations in Sections 4.1.1 and 4.1.2 hold if we consider n_{ex} experiments, too. We extend the notation of the previous subsections by the superscript i with $i = 1, \dots, n_{\text{ex}}$. The likelihood function for n_{ex} experiments then reads as follows:

$$L(\theta|\eta) := \prod_{i=1}^{n_{\text{ex}}} (\sqrt{2\pi})^{-m_i} \prod_{j=1}^{m_i} \left(\sigma_j^i\right)^{-1} \exp\left(-\frac{\left(\eta_j^i - h_j^i(t_j^i, y^i(t_j^i), z^i(t_j^i), \theta)\right)^2}{2\left(\sigma_j^i\right)^2} \right). \tag{4.9}$$

As the name suggests, the maximum likelihood estimation is to find parameters $\hat{\theta}$ such that the likelihood function (4.9) is maximized:

$$\begin{aligned}
\hat{\theta} &= arg \max_{\theta \in I\!R^{n_\theta}} L(\theta|\eta) \\
&= arg \max_{\theta \in I\!R^{n_\theta}} \log L(\theta|\eta) \\
&= arg \max_{\theta \in I\!R^{n_\theta}} -\frac{1}{2} \sum_{i=1}^{n_{\text{ex}}} \sum_{j=1}^{m_i} \frac{(\eta_j^i - h_j^i(t_j^i, \xi^i(t_j^i), \theta))^2}{2(\sigma_j^i)^2} + \underbrace{\sum_{i=1}^{n_{\text{ex}}} \sum_{j=1}^{m_i} \log \frac{1}{\sqrt{2\pi(\sigma_j^i)^2}}}_{\text{independent of } \theta} \\
&= arg \min_{\theta \in I\!R^{n_\theta}} \frac{1}{2} \sum_{i=1}^{n_{\text{ex}}} \sum_{j=1}^{m_i} \frac{(\eta_j^i - h_j^i(t_j^i, \xi^i(t_j^i), \theta))^2}{2(\sigma_j^i)^2}
\end{aligned}$$

with $\xi_j^i = (y_j^i, z_j^i)$.

Considering the model equations (4.1) and the interior point constraints (4.4) as additional constraints of the problem, we obtain the constrained parameter estimation problem

$$\min_{\theta, y, z} \quad \frac{1}{2} \sum_{i=1}^{n_{\text{ex}}} \sum_{j=1}^{m_i} \left(\frac{\eta_j^i - h_j^i(t_j^i, \xi^i(t_j^i), \theta)}{\sigma_j^i} \right)^2 \tag{4.10a}$$

$$\text{s.t.} \quad A^i(t,y^i(t),z^i(t),\theta)\dot{y}^i(t) = f^i(t,y^i(t),z^i(t),\theta), \tag{4.10b}$$

$$0 = g^i(t,y^i(t^i),z^i(t),\theta), \qquad i = 1,\dots,n_{\text{ex}}, \tag{4.10c}$$

$$0 = \sum_{k=1}^{m_r^i} r_k^i\left(y^i(t_k^i),z^i(t_k^i),\theta\right). \tag{4.10d}$$

4.2 Direct shooting parametrization of parameter estimation problems

So far, we consider a infinite dimensional optimization problem in function space. For numerical examinations, we have to transform the parameter estimation problem (4.10) into a finite dimensional one. Therefore, the dynamic model (4.1) has to be parametrized.

The following method is applied for all n_{ex} experiments. Without loss of generality, we present the methods for the first experiment and omit the superscript 1.

The most obvious approach is the single shooting method. The *relaxed* system of differential algebraic equation is solved

$$\begin{aligned} A(t,y(t),z(t),\theta,q,u(t))\dot{y}(t) &= f(t,y(t),z(t),\theta) \\ 0 &= g(t,y(t),z(t),\theta,) - \beta(t)\cdot g(t_0,y(t_0),z(t_0),\theta) \end{aligned}$$

with the initial values

$$y(t_0) = y_0(\theta), \quad z(t_0) = s_0^z$$

and a monotonic decreasing function $\beta(t)$ with $\beta(t_0) = 1$ and $\lim_{t\to t_{\text{end}}} \beta(t) = 0$. This results in a representation of the solution at every time point $t \in [t_0, t_{\text{end}}]$ which depends on the parameters θ and the parametrization variables s_0^z

$$\begin{pmatrix} y(t) \\ z(t) \end{pmatrix} = \psi(t; s_0^z, p).$$

We refer to ψ as the nominal trajectory. The consistency condition

$$0 = g(t_0, y(t_0), s_0^z, \theta)$$

is considered as an additional constraint of the optimization problem (4.10).

Bock proved [13] that single shooting may not be suitable to parametrize the dynamic model of a parameter estimation problem. The initial value problem may be ill-conditioned, i.e. small perturbations of the initial values may lead to large perturbations of the solution of the parameter estimation problem in a nonlinear way. This is described by the following Lemma, that can be found in Bock et al. [14].

Lemma 4.2.1. *Let $f \in C^0(D)$ and Lipschitz continuous on $\bar{D} = [a,b] \times \{\|y - y_0\| \leqslant K\} \subset D$ with Lipschitz constant $L < \infty$. The solutions y, w of*

$$\begin{aligned} \dot{y}(t) &= f(t,y(t)), & y(t_0) &= y_0 \\ \dot{w}(t) &= f(t,y(t)) + \delta f(t,y(t)), & w(t_0) &= w_0 = y_0 + \delta y_0 \end{aligned}$$

both exist on $[t_0,t_f] \subset [a,b]$ in $\bar{D}$. Under the two assumptions $\|\delta y_0\| \leqslant \varepsilon_1$ and $\|\delta f(t,w(t))\| \leqslant \varepsilon_2$ for all $(t,w(t))$, the deviation $\delta y(t) := w(t) - v(t)$ satisfies

$$\|\delta y(t)\| \leqslant \varepsilon_1 \exp(L \cdot (t - t_0)) + \varepsilon_2 \exp(L \cdot (t - t_0))(t - t_0)$$

The proof uses Gronwall's Lemma [3].

In other words, small perturbations of the initial values or the model equations, mainly by the unknown parameter values, may propagate exponentially in time.

We suggest the use of the direct multiple shooting method to overcome this problem. In the beginning, this method was applied to solve boundary value problems, e.g., see Stoer and Bulirsch [89]. Bock and Plitt [17] successfully applied this method to the parametrization of the model equations for solving parameter estimation problems.

We decompose the time interval into $n_{\text{ms}} + 1$ subintervals

$$t_0 = \tau_0 < \tau_1 < \ldots < \tau_{n_{\text{ms}}} < \tau_{n_{\text{ms}}+1} = t_{\text{end}}$$

and solve the relaxed DAE system

$$\begin{aligned} A(t,y(t),z(t),\theta)\dot{y}(t) &= f(t,y(t),z(t),\theta), \\ 0 &= g(t,y(t),z(t),\theta) - \beta(t) \cdot g(\tau_k,y(\tau_k),z(\tau_k),\theta). \end{aligned}$$

on the subintervals $t \in [\tau_k, \tau_{k+1})$, $k = 0,\ldots,n_{\text{ms}}$. Here, we use an interval-wise definition of the monotonic decreasing function β with $\beta(t) \in [0,1]$, $\beta(\tau_k) = 1$ and $\lim\limits_{t \to \tau_{k+1}} \beta(t) = 0$. For each subinterval we have initial conditions

$$y(\tau_k) = s_k^y, \quad z(\tau_k) = s_k^z, \quad k = 0,\ldots,n_{\text{ms}}$$

and $s_0^y = y_0(\theta)$. The parameterization of the solution of (4.1) is illustrated in Figure 4.1 for $n_y = 1$ and $n_z = 0$. The shooting variables $s_k = (s_k^y, s_k^z)$, $k = 0,\ldots,n_{\text{ms}}$ are additional degrees of freedom of the problem. This approach leads to a piecewise continuous representation of the nominal trajectories of the model equations (4.1)

$$\begin{pmatrix} y(t) \\ z(t) \end{pmatrix} = \psi(t; s_k, \theta), \quad t \in [\tau_k, \tau_{k+1}), \quad k = 0,\ldots,n_{\text{ms}}. \tag{4.11}$$

Since we want the trajectories to be continuous and consistent, we have to demand that the continuity conditions for the differential states

$$0 = c(\tau_{k+1}, s_k, s_{k+1}, \theta) := \psi^y(\tau_{k+1}; s_k, \theta) - s_{k+1}^y, \quad k = 0,\ldots,n_{\text{ms}} - 1 \tag{4.12}$$

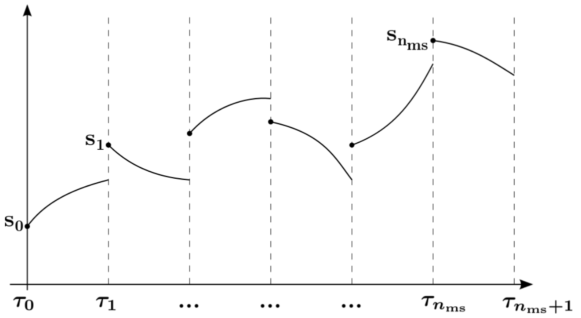


Figure 4.1: Illustration of the parameterized solution of an ODE using multiple shooting.

and the consistency conditions for the algebraic states

$$0 = g(\tau_k, s_k, \theta), \quad k = 0, \dots, n_{\text{ms}} \tag{4.13}$$

are fulfilled. In the following, we use a closed form for the interior point constraints

$$r^i\left(\psi^i(t_1^i), \dots, \psi^i(t_{m_r^i}^i), \theta\right) := \sum_{k=1}^{m_r^i} r_k^i\left(\psi^i(t_k^i), \theta\right). \tag{4.14}$$

To improve readability, we omit the dependencies of ψ from s and θ. By using the parametrized formulation of the trajectory and adding the constraints (4.12) and (4.13) to the problem formulation (4.10), we obtain a finite dimensional, nonlinear constrained least-squares problem

$$\min_{\theta,s} \quad \tfrac{1}{2}(\eta - h(\psi(s,\theta),\theta)^T \Sigma^{-2} (\eta - h(\psi(s,\theta),\theta)) \tag{4.15a}$$

$$\begin{aligned}
\text{s.t.} \quad 0 =& c^i(\tau_{k+1}^i, s_k^i, s_{k+1}^i, \theta), & k = 0, \dots, n_{\text{ms}}^i - 1 \quad & (4.15\text{b}) \\
0 =& g^i(\tau_k^i, s_k^i, \theta), & k = 0, \dots, n_{\text{ms}}^i \quad & (4.15\text{c}) \\
& & i = 1, \dots, n_{\text{ex}} \quad & \\
0 =& r^i\left(\psi^i(t_1^i), \dots, \psi^i(t_{m_r^i}^i), \theta\right), & & (4.15\text{d})
\end{aligned}$$

with $s = (s_1^1, \dots, s_{n_{\text{ms}}^1}^1, \dots, s_1^{n_{\text{ex}}}, \dots, s_{n_{\text{ms}}^{n_{\text{ex}}}}^{n_{\text{ex}}})$. Here, we use an abbreviated form for the vector of all measurement values

$$\eta := (\eta_1^1, \dots, \eta_{m_1}^1, \dots, \eta_1^{n_{\text{ex}}}, \dots, \eta_{m_{n_{\text{ex}}}}^{n_{\text{ex}}})^T.$$

The vector of all model responses $h(\psi(s,\theta),\theta)$ and variance matrix Σ as a diagonal matrix of the standard deviations are structured according to this definition.

4.3 Generalized Gauss–Newton method

By defining

$$F_1(s,\theta) := \Sigma^{-1}(\eta - h(\xi(s,\theta),\theta)) \tag{4.16a}$$

$$F_2(s,\theta) := \begin{pmatrix} \left(c^i(\tau^i_{k+1}, s^i_k, s^i_{k+1}, \theta)\right)_{\substack{k=0,\dots,n^i_{\mathrm{ms}}-1\\ i=1,\dots,n_{\mathrm{ex}}}} \\ \left(g^i(\tau^i_k, s^i_k, \theta)\right)_{\substack{k=0,\dots,n^i_{\mathrm{ms}}\\ i=1,\dots,n_{\mathrm{ex}}}} \\ \left(r^i\left(\psi^i(t^i_1),\dots,\psi^i(t^i_{m^i_r}),\theta\right)\right)_{i=1,\dots,n_{\mathrm{ex}}} \end{pmatrix}, \tag{4.16b}$$

we get a shorter formulation for (4.15)

$$\min_{s,\theta} \quad \tfrac{1}{2}\|F_1(s,\theta)\|_2^2 \tag{4.17a}$$

$$\text{s.t.} \quad 0 = F_2(s,\theta). \tag{4.17b}$$

We apply the generalized Gauss–Newton method as it was presented by Bock [13] to solve (4.17). Problem (4.17) is solved iteratively by examining a linearization. For the linearization, the evaluation of the Jacobians

$$J_1 = \frac{\partial F_1}{\partial(s,\theta)} \tag{4.18a}$$

$$J_2 = \frac{\partial F_2}{\partial(s,\theta)} \tag{4.18b}$$

and

$$J = \begin{pmatrix} J_1 \\ J_2 \end{pmatrix}$$

is needed. The following outline describes the generalized Gauss–Newton method.

Algorithm 4.3.1. ***Generalized Gauss–Newton method***

1. *Set* $l := 0$. *Choose an initial guess* (s^0, θ^0) *and a termination tolerance* TOL.

2. *Solve the linear constrained least-squares problem*

$$\min_{\Delta s^l, \Delta\theta^l} \quad \frac{1}{2}\left\| F_1(s^l,\theta^l) + J_1(s^l,\theta^l)\begin{pmatrix}\Delta s^l \\ \Delta\theta^l\end{pmatrix}\right\|_2^2 \tag{4.19a}$$

$$s.t. \quad 0 = F_2(s^l,\theta^l) + J_2(s^l,\theta^l)\begin{pmatrix}\Delta s^l \\ \Delta\theta^l\end{pmatrix}. \tag{4.19b}$$

3. *Update the iterate*

$$\begin{pmatrix} s^{l+1} \\ \theta^{l+1}\end{pmatrix} := \begin{pmatrix} s^l \\ \theta^l \end{pmatrix} + \begin{pmatrix}\Delta s^l \\ \Delta\theta^l\end{pmatrix}.$$

4. *If*

$$\left\|\begin{pmatrix}\Delta s^l \\ \Delta\theta^l\end{pmatrix}\right\|_2 < TOL,$$

terminate, otherwise:

5. *Set $l := l+1$. Go to 2.*

We assume that problem (4.19) is regular in all points (s,θ) where we have to evaluate F and J. Let $(s,\theta) \in I\!R^n$, $F_1(s,\theta) \in I\!R^{n_1}$ and $F_2(s,\theta) \in I\!R^{n_2}$. We demand that the following regularity conditions are fulfilled:

1. **Constraint Qualification (CQ)**

$$\operatorname{rank} J_2(s,\theta) = n_2.$$

2. **Positive Definiteness (PD)**

$$\operatorname{rank} J(s,\theta) = n.$$

Conditions (CQ) and (PD) are equivalent to the condition, that $J_1(s,\theta)^T J_1(s,\theta)$ is positive definite on the kernel of $J_2(s,\theta)$. Under these assumptions, we formulate a Lemma about the representation of the solution of (4.19) that can be found in Bock et al. [14].

Lemma 4.3.2. *Let (CQ) and (PD) hold. The solution of the constrained linear least-squares problem*

$$\min_{\Delta s,\Delta\theta} \quad \frac{1}{2}\left\| F_1 + J_1 \begin{pmatrix} \Delta s \\ \Delta\theta \end{pmatrix} \right\|_2^2 \tag{4.20a}$$

$$s.t. \quad 0 = F_2 + J_2 \begin{pmatrix} \Delta s \\ \Delta\theta \end{pmatrix} \tag{4.20b}$$

can be represented as

$$\begin{pmatrix} \Delta s^* \\ \Delta\theta^* \end{pmatrix} = -\begin{pmatrix} I & 0 \end{pmatrix} \begin{pmatrix} J_1^T J_1 & J_2^T \\ J_2 & 0 \end{pmatrix}^{-1} \begin{pmatrix} J_1^T & 0 \\ 0 & I \end{pmatrix} \begin{pmatrix} F_1 \\ F_2 \end{pmatrix}$$

and the Lagrangian multiplier is

$$\lambda^* = \begin{pmatrix} 0 & I \end{pmatrix} \begin{pmatrix} J_1^T J_1 & J_2^T \\ J_2 & 0 \end{pmatrix}^{-1} \begin{pmatrix} J_1^T & 0 \\ 0 & I \end{pmatrix} \begin{pmatrix} F_1 \\ F_2 \end{pmatrix}.$$

Proof. Because of (CQ) and (PD),

$$\begin{pmatrix} J_1^T J_1 & J_2^T \\ J_2 & 0 \end{pmatrix}$$

is regular. The Lagrangian of the constrained linear least-squares problem is

$$\begin{aligned} \mathscr{L}(\Delta s,\Delta\theta,\lambda) =& \frac{1}{2} F_1^T F_1 + F_1^T J_1 \begin{pmatrix} \Delta s \\ \Delta\theta \end{pmatrix} + \frac{1}{2} \begin{pmatrix} \Delta s^T & \Delta\theta^T \end{pmatrix} J_1^T J_1 \begin{pmatrix} \Delta s \\ \Delta\theta \end{pmatrix} \\ & - \lambda^T \left(F_2 + J_2 \begin{pmatrix} \Delta s \\ \Delta\theta \end{pmatrix} \right). \end{aligned}$$

Setting the gradient of the Lagrangian to zero

$$0 = \nabla\mathscr{L}(\Delta s^*, \Delta\theta^*, \lambda^*) = \begin{pmatrix} J_1^T F_1 + J_1^T J_1 \begin{pmatrix} \Delta s^* \\ \Delta\theta^* \end{pmatrix} - J_2^T \lambda^* \\ F_2 + J_2 \begin{pmatrix} \Delta s^* \\ \Delta\theta^* \end{pmatrix} - J_2^T \lambda^* \end{pmatrix},$$

which here is necessary and sufficient for optimality because of (CQ) and (PD), results in

$$\begin{pmatrix} \Delta s^* \\ \Delta\theta^* \\ -\Delta\lambda^* \end{pmatrix} = \begin{pmatrix} J_1^T J_1 & J_2^T \\ J_2 & 0 \end{pmatrix}^{-1} \begin{pmatrix} J_1^T & 0 \\ 0 & I \end{pmatrix} \begin{pmatrix} F_1 \\ F_2 \end{pmatrix}.$$

□

The generalized inverse of J is defined by

$$J^+ = (I\ 0) \begin{pmatrix} J_1^T J_1 & J_2^T \\ J_2 & 0 \end{pmatrix}^{-1} \begin{pmatrix} J_1^T & 0 \\ 0 & I \end{pmatrix} \tag{4.21}$$

and we obtain the representation

$$\begin{pmatrix} \Delta s^* \\ \Delta\theta^* \end{pmatrix} = -J^+ F$$

with

$$F = \begin{pmatrix} F_1 \\ F_2 \end{pmatrix} \tag{4.22}$$

for the solution of the linear constrained least-squares problem (4.20).

4.3.1 Local convergence of Newton type method

In the next subsections, we use v as an acronym for the variables (s, θ) of problem (4.17). For the Newton type methods, the increment is computed as the solution of a linearized problem according to

$$\Delta v^l = -M(v^l)F(v^l).$$

We distinguish

- *Newton's method* with
$$M(v) = J^{-1}(v)$$
for nonlinear systems of equations $F(v^k) = 0$,

- *Quasi–Newton methods* with an approximation of the inverse (based on secant updates, see, e.g., Nocedal and Wright [77])
$$M(v) \approx J^{-1}(v),$$

- and the *generalized Gauss–Newton method* with

  $$M(v) = J^+(v)$$

  for constrained nonlinear least-squares problems.

We state the local contraction theorem, proved by Bock [13], to define the local convergence behavior of Newton type methods.

Theorem 4.3.3. *[Local Contraction theorem] Let* $F : \mathbb{R}^m \supseteq D \to \mathbb{R}^n,\ F \in C^2(D, \mathbb{R}^n)$, $J := \frac{dF}{dv}$*. For all* $v, w \in D,\ \vartheta \in [0,1]$ *with* $w - v = -M(v)F(v)$ *:*

1. *There exists* $\omega \leqslant \infty$*, such that the Lipschitz condition*

   $$\|M(w)\,(J(v + \vartheta(w - v)) - J(v))\,(w - v)\| \leqslant \omega \cdot \vartheta \cdot \|w - v\|^2$$

   holds.

2. *There exists* $\kappa(v) \leqslant \kappa < 1$*, such that the compatibility condition*

   $$\|M(w)R(v)\| \leqslant \kappa(v)\|v - w\|$$

   for the residual $R(v) := F(v) - J(v)M(v)F(v)$ *is satisfied.*

Let $v^0 \in D$ *be given with*

$$\begin{aligned}
\Delta v^l &:= -M(v^l)F(v^l) \\
\delta_0 &:= \kappa + \frac{\omega}{2}\|\Delta v^0\| < 1,\ \ \delta_k := \kappa + \frac{\omega}{2}\|\Delta v^l\|, \\
D_0 &:= \left\{ v : \|v - v^0\| \leqslant \frac{\|\Delta v^0\|}{1 - \delta_0} \right\} \subseteq D.
\end{aligned}$$

Then:

1. *The iteration* $v^{l+1} = v^l + \Delta v^l$ *is well defined and stays in* D_0.
2. *There exists an* $v^* \in D_0$*, such that* v^l *converges to* v^* *for* $l \to \infty$.
3. *The following a priori estimate holds:*

   $$\|v^{l+j} - v^*\| \leqslant \frac{{\delta_l}^j}{1 - \delta_l}\|\Delta v^l\|.$$

4. *The following a posteriori estimate holds:*

   $$\|\Delta v^{l+1}\| \leqslant \delta_l\|\Delta v^l\| = \kappa\|\Delta v^l\| + \frac{\omega}{2}\|\Delta v^l\|^2.$$

It follows directly that Newton's method for unconstrained nonlinear problems with $M(v) = J(v)^{-1}$ converges locally quadratically because $R(v) = 0$ and therefore $\kappa = 0$. Applied to the Gauss–Newton methods for constrained nonlinear least-squares problems, we draw the following conclusions, that can be found in Bock et al. [14].

Corollary 4.3.4. *(Local convergence of Gauss-Newton methods) Let* $F = \begin{pmatrix} F_1 \\ F_2 \end{pmatrix} : \mathbb{R}^m \supseteq D \to \mathbb{R}^n$, $F \in C^2(D, \mathbb{R}^n)$, $J := \frac{dF}{dv}$. *Let (CQ) and (PD) hold and*

$$J^+(v)F(v) = \begin{pmatrix} I & 0 \end{pmatrix} \begin{pmatrix} J_1^T(v)J_1(v) & J_2^T(v) \\ J_2(v) & 0 \end{pmatrix}^{-1} \begin{pmatrix} J_1^T(v) & 0 \\ 0 & I \end{pmatrix} \begin{pmatrix} F_1(v) \\ F_2(v) \end{pmatrix}$$

be the solution of the linearized constrained least-squares problem

$$\begin{aligned} \min_{\Delta v} \quad & \tfrac{1}{2} \|F_1(v) + J_1(v)\|_2^2 \\ s.t. \quad 0 =& F_2(v) + J_2(v). \end{aligned}$$

Let $J^+(w)$ be bounded in a neighborhood of the solution v^* of the nonlinear problem, $\|J^+(w)\| \leqslant \beta$, and let J satisfy the following Lipschitz condition

$$\|J(v + \vartheta(w - v)) - J(v))(w - v)\| \leqslant \gamma \cdot \vartheta \cdot \|w - v\|^2.$$

J^+ is continuously differentiable, thus

$$\|J^+(w) - J^+(v)\| \leqslant L \cdot \|w - v\|.$$

Let $\|R(v)\| \leqslant \rho$. Then the three conclusions hold:

1. *$\omega = \beta \cdot \gamma$ is large if*
   - *$\|J'\|$ respectively $\|F''\|$ is large, i.e. the problem is very nonlinear.*
   - *$\|J^+(w)\|$ is large, i.e. $\begin{pmatrix} J_1^T J_1 & J_2^T \\ J_2 & 0 \end{pmatrix}$ is almost singular.*
2. *Due to*

   $$\begin{aligned} J^+(v)R(v) &= J^+(v)(F(v) - J(v)J^+(v)F(v)) \\ &= (J^+(v) - J^+(v)J(v)J^+(v))F(v) = 0, \end{aligned}$$

   it holds

   $$\|J^+(w)R(v)\| = \|(J^+(w) - J^+(v))R(v)\| \leqslant \rho \cdot L \cdot \|w - v\|.$$

   Hence $\kappa = \rho \cdot L < 1$ if
   - *the residual R is small,*
   - *J^+ satisfies a Lipschitz condition respectively the first derivative of J^+ is small.*
3. *If the starting value is close to the solution, where "close" is determined by*

   $$\kappa + \frac{\omega}{2} \|\Delta v^0\| < 1,$$

   then the Gauss–Newton method converges, and the convergence is linear with convergence rate $\kappa < 1$.

4.3.2 Statistical analysis of the solution

Since measurement data are random variables, the estimated parameters as the solution of the generalized Gauss–Newton method will be a random variable, too.

We consider the linearized problem (4.19) in the solution $\hat{v}$ of the Gauss–Newton method. We have, $\Delta v = -J^{+}F$ is multivariate normally distributed with mean

$$\mathbb{E}\left(\Delta v\right) = \mathbb{E}\left(-J^{+}\begin{pmatrix}F_1\\F_2\end{pmatrix}\right) = -J^{+}\mathbb{E}\begin{pmatrix}F_1\\F_2\end{pmatrix} = -J^{+}\begin{pmatrix}0\\0\end{pmatrix} = 0$$

and variance–covariance matrix

$$\begin{aligned} C &= \mathbb{E}\left(\Delta v \Delta v^T\right) \\ &= \mathbb{E}\left(J^{+}\begin{pmatrix}F_1F_1^T & F_1F_2^T\\ F_2F_1^T & F_2F_2^T\end{pmatrix}J^{+^T}\right) \\ &= J^{+}\begin{pmatrix}\mathbb{E}\left(F_1F_1^T\right) & 0\\ 0 & 0\end{pmatrix}J^{+^T} \\ &= J^{+}\begin{pmatrix}I & 0\\ 0 & 0\end{pmatrix}J^{+^T} \\ &= (I\;0)\begin{pmatrix}J_1^TJ_1 & J_2^T\\ J_2 & 0\end{pmatrix}^{-1}\begin{pmatrix}J_1^TJ_1 & 0\\ 0 & 0\end{pmatrix}\begin{pmatrix}J_1^TJ_1 & J_2^T\\ J_2 & 0\end{pmatrix}^{-T}\begin{pmatrix}I\\0\end{pmatrix}. \end{aligned} \tag{4.23}$$

Here, we used that

$$\mathbb{E}\left(F_1F_1^T\right) = \Sigma^{-1}\mathbb{E}\left(\eta\eta^T\right)\Sigma^{-1} = \Sigma^{-1}\Sigma^2\Sigma^{-1} = I$$

for the vector of measurement values η and $\mathbb{E}\left(F_1F_2^T\right) = 0$, $\mathbb{E}\left(F_2F_1^T\right) = 0$, $\mathbb{E}\left(F_2F_2^T\right) = 0$, since F_2 is not a random variable.

The $(100\cdot\alpha)\%$ confidence region is the region where the true parameters θ^* lie with probability $\alpha \in [0,1]$.

In Section 4.3, we stated that the nonlinear constraint least-squares problem (4.17) has n variables and n_2 equality constraints. Let

$$\gamma^2(\alpha) := \chi^2_{n-n_2}(1-\alpha) \tag{4.24}$$

be the quantile of the χ^2 distribution for α with $n-n_2$ degrees of freedom.

Since the true parameters are unknown, we examine an approximation of the $(100\cdot\alpha)$ confidence region for the solution $\hat{v}$ of the generalized Gauss–Newton

$$\mathscr{G}(\alpha,\hat{v}) := \left\{v \in I\!R^n :\; F_2(v) = 0,\; \|F_1(v)\|_2^2 - \|F_1(\hat{v})\|_2^2 \leqslant \gamma^2(\alpha)\right\}.$$

This region is not easy to compute because of the nonlinearity of the least-squares function and the constraints. Thus, we consider the linear approximation

$$\begin{aligned} \mathscr{G}_L(\alpha,\hat{v}) := \{v \in I\!R^n :\; & F_2(\hat{v}) + J_2(\hat{v})(v-\hat{v}) = 0, \\ & \|F_1(\hat{v}) + J_1(\hat{v})(v-\hat{v})\|_2^2 - \|F_1(\hat{v})\|_2^2 \leqslant \gamma^2(\alpha)\} \end{aligned} \tag{4.25}$$

Bock [13] has shown that the linearized confidence region can be expressed by the formula

$$\bar{\mathscr{G}}_L(\alpha,\hat{v}) := \left\{ v \in I\!R^n : \; v - \hat{v} = -J^+(\hat{v}) \begin{pmatrix} \delta w \\ 0 \end{pmatrix}, \; \delta w \in I\!R^{n_1}, \; \|\delta w\|_2 \leqslant \gamma(\alpha) \right\}. \tag{4.26}$$

Following Körkel [58], we use the last results to define estimates for the confidence intervals of the single parameters.

Corollary 4.3.5. *Let*

$$\Theta_i := \gamma(\alpha) \cdot \sqrt{C_{ii}(\hat{v})}, \quad i = 1, \ldots, n,$$

with $\gamma(\alpha)$ *as in* (4.24) *and* $\sqrt{C_{ii}(\hat{v})}$ *the square root of the i-th element of the main diagonal of the variance–covariance matrix* (4.23)*. Then it holds: The linearized confidence interval* $\mathscr{G}_L(\alpha,\hat{v})$ *is enclosed by the cuboid with side length* $2 \cdot \Theta_i$ *:*

$$\mathscr{G}_L(\alpha,\hat{v}) \subseteq [\hat{v}_1 - \Theta_1, \hat{v}_1 + \Theta_1] \times \ldots \times [\hat{v}_n - \Theta_n, \hat{v}_n + \Theta_n].$$

Proof. Let $e_i \in I\!R^n$ be the i-th unit vector. With (4.26) we have for $v \in G_L(\alpha,\hat{v})$:

$$\begin{aligned} |v_i - \hat{v}_i| &\leqslant \left\| \left(J^+(\hat{v}) \begin{pmatrix} I \\ 0 \end{pmatrix} \right)^T e_i \right\|_2 \cdot \|\delta w\|_2 \\ &= \left(e_i^T J^+(\hat{v}) \begin{pmatrix} I \\ 0 \end{pmatrix} \begin{pmatrix} I & 0 \end{pmatrix} J^+(\hat{v})^T e_i \right)^{\frac{1}{2}} \cdot \|\delta w\|_2 \\ &\leqslant \sqrt{C_{ii}(\hat{v})} \cdot \gamma(\alpha), \quad i = 1, \ldots, n. \end{aligned}$$

□

4.3.3 Computation of the Jacobian

In the following two subsections, we consider a single experiment. The extension to multiple experiments is straightforward.

As mentioned above, we have to compute the Jacobian of problem (4.17)

$$J_1 = \frac{\partial F_1}{\partial(s,\theta)}, \quad J_2 = \frac{\partial F_2}{\partial(s,\theta)},$$

in every iteration of the generalized Gauss–Newton algorithm. In the context of multiple shooting, the Jacobian has the following structure

$$J = \begin{pmatrix} J_1 \\ J_2 \end{pmatrix} = \left(\begin{array}{ccccc} D_0^1 & D_1^1 & \cdots & D_{n_{\mathrm{ms}}}^1 & D_\theta^1 \\ \hline G_0 & \begin{pmatrix} -I_{n_y} & 0 \end{pmatrix} & & & G_0^\theta \\ & \ddots & \ddots & & \vdots \\ & & G_{n_{\mathrm{ms}}-1} & \begin{pmatrix} -I_{n_y} & 0 \end{pmatrix} & G_{n_{\mathrm{ms}}-1}^\theta \\ H_0 & & & & H_0^\theta \\ & H_1 & & & H_1^\theta \\ & & \ddots & & \vdots \\ & & & H_{n_{\mathrm{ms}}} & H_{n_{\mathrm{ms}}}^\theta \\ D_0^2 & D_1^2 & \cdots & D_{n_{\mathrm{ms}}}^2 & D_\theta^2 \end{array}\right), \tag{4.27}$$

with

- the derivatives of the least-squares functional

$$D_k^1 := \frac{\partial F_1}{\partial s_i}, \quad D_\theta^1 := \frac{\partial F_1}{\partial \theta}, \quad k = 0, \ldots, n_{\mathrm{ms}},$$

- the derivatives of the continuity conditions

$$G_k := \frac{\partial \psi^y}{\partial s_k}(\tau_{k+1}; s_k, \theta), \quad G_k^\theta := \frac{\partial \psi^y}{\partial \theta}(\tau_{k+1}; s_k, \theta), \quad k = 0, \ldots, n_{\mathrm{ms}} - 1,$$

- the derivatives of the consistency conditions

$$H_k := \frac{\partial g}{\partial s_k}(\tau_k, s_k, \theta), \quad H_k^\theta := \frac{\partial g}{\partial \theta}(\tau_k, s_k, \theta), \quad k = 0, \ldots, n_{\mathrm{ms}},$$

- and the derivatives of the interior point constraints

$$D_k^2 := \frac{\partial r}{\partial s_k}, \quad D_\theta^2 := \frac{\partial r}{\partial \theta}, \quad k = 0, \ldots, n_{\mathrm{ms}}.$$

Note, that for $n_{\mathrm{ms}} = 0$ and consequently $s_0 = s_0^z$ we obtain the Jacobian of the single shooting case. For the evaluation of the block matrices G and H, we apply the approach of forward internal numerical differentiation (IND), i.e., the incorporation of the derivative computation into the integrator scheme. Thereby, the derivatives are determined as solutions of the variational differential equations (VDE) with respect to the shooting variables s_k

$$\begin{aligned} A(t, \psi(t), \theta)\dot{G}_k(t) = & \frac{\partial f}{\partial \xi}(t, \psi(t), \theta) \begin{pmatrix} G_k(t) \\ H_k(t) \end{pmatrix}, \quad G_k(\tau_k) = I, \quad k = 0, \ldots, n_{\mathrm{ms}}, \\ & - \frac{\partial A}{\partial \xi}(t, \psi(t), \theta) \end{aligned}$$

$$0 = \frac{\partial g}{\partial \xi}(t, \psi(t), \theta) \begin{pmatrix} G_k(t) \\ H_k(t) \end{pmatrix}$$

and with respect to the parameters

$$\begin{aligned} A(t, \psi(t), \theta)\dot{G}_k^\theta(t) =& \frac{\partial f}{\partial \theta}(t, \psi(t), \theta) \begin{pmatrix} G_k^\theta(t) \\ H_k^\theta(t) \end{pmatrix} + \frac{\partial f}{\partial \theta}(t, \psi(t), \theta), \quad k = 0, \dots, n_{\mathrm{ms}} \\ & - \left(\frac{\partial A}{\partial \xi}(t, \psi(t), \theta) \begin{pmatrix} G_k^\theta(t) \\ H_k^\theta(t) \end{pmatrix} + \frac{\partial A}{\partial \theta}(t, \psi(t), \theta) \right) \\ 0 =& \frac{\partial g}{\partial \theta}(t, \psi(t), \theta) \begin{pmatrix} G_k^\theta(t) \\ H_k^\theta(t) \end{pmatrix} + \frac{\partial g}{\partial \theta}(t, \psi(t), \theta), \\ G_k^\theta(\tau_k) =& \begin{cases} \frac{\partial y_0}{\partial \theta}(\theta) & , k = 0 \\ 0 & , else \end{cases} . \end{aligned}$$

These equations form a system of differential algebraic equations and can be solved as a system together with the DAE system (4.1) which we refer to as the nominal DAE. For the concrete realization of this method we refer to, e.g., Bock [11], Bauer [7], Albersmeyer [2] and Lenz [62].

If the initial values of the algebraic states are inconsistent, a relaxed formulation has to be applied here as well, i.e., we differentiate the relaxed formulation of the nominal problem according to the above settings.

The required derivatives of the model functions f, g, h and r are computed by algorithmic differentiation (AD), see Griewank [46]. In comparison with finite differences or symbolic differentiation, AD has two main advantages:

(i) No suffering from numerical truncation errors.

(ii) The evaluation is done efficiently, i.e. already computed expressions are not evaluated twice.

4.3.4 Condensing

The multiple shooting formulation induces a comparatively large, but very sparse Jacobian with a special structure, compare Equation (4.27). Bock [13] developed a factorization algorithm that exploits the special structure of the Jacobian for ODE constrained parameter estimation problems and manipulates the system by a sequence of block Gauss eliminations; the *condensing* method. Due to the index-1-assumption for the algebraic equations (4.1b) this method can be extended to parameter estimation problems with DAE constraints, see Leineweber [61].

We separate $H_k = \begin{pmatrix} H_k^y & H_k^z \end{pmatrix}$ with

$$H_k^y = \frac{\partial g}{\partial s_k^y}(\tau_k, s_k, \theta), \quad H_k^y = \frac{\partial g}{\partial s_k^z}(\tau_k, s_k, \theta), \quad k = 0, \dots, n_{\mathrm{ms}}$$

and transform the corresponding part in the constrained linear least-squares problem (4.20) with respect to Δs_k^z :

$$\Delta s_k^z = -\left(H_k^z\right)^{-1}\left[H_k^y \Delta s_k^y + H_k^\theta \Delta\theta + g(\tau_k, s_k, \theta)\right], \quad k = 0, \ldots, n_{\mathrm{ms}}. \tag{4.28}$$

We do not compute the inverse of H_k^z but use a decomposition algorithm, e.g., QR or singular value decomposition. Equation (4.28) is applied to (4.20) and we obtain a modified representation of the Jacobian and the right-hand side of the constrained linear least-squares problem which does not depend on Δs_k^z, $k = 0, \ldots, n_{\mathrm{ms}}$

$$\hat{J} = \begin{pmatrix} \hat{D}_0^1 & \hat{D}_1^1 & \cdots & \hat{D}_{n_{\mathrm{ms}}}^1 & \hat{D}_\theta^1 \\ \hat{G}_0 & -I_{n_y} & & & \hat{G}_0^\theta \\ & \ddots & \ddots & & \vdots \\ & & \hat{G}_{n_{\mathrm{ms}}-1} & -I_{n_y} & \hat{G}_{n_{\mathrm{ms}}-1}^\theta \\ \hat{D}_0^2 & \hat{D}_1^2 & \cdots & \hat{D}_{n_{\mathrm{ms}}}^2 & \hat{D}_\theta^2 \end{pmatrix}, \quad \hat{F} = \begin{pmatrix} \hat{F}_1 \\ \hat{c}(\tau_1) \\ \vdots \\ \hat{c}(\tau_{n_{\mathrm{ms}}}) \\ \hat{r} \end{pmatrix}. \tag{4.29}$$

Here, we suppressed the dependencies of $\hat{c}$ with respect to s_k, s_{k+1} and θ. Because of the identity matrices I_{n_y} on the diagonal of $\hat{J}$, cf. (4.29), the elimination is straightforward. We multiply the rows of $\hat{J}$ which belong to the continuity constraints by the corresponding submatrices D_k^i, $i = 1, 2$, $k = 1, \ldots, n_{\mathrm{ms}}$ and sum up the results. The condensing algorithm factorizes the system, which is defined by (4.29), to the condensed system for the increments of the first shooting variable Δs_0^y and the parameters $\Delta\theta$:

$$\min_{\Delta s_0^y, \Delta\theta} \quad \frac{1}{2}\left\| u_1 + E_1 \Delta s_0^y + E_1^\theta \Delta\theta \right\|_2^2 \tag{4.30a}$$

$$\text{s.t.} \quad 0 = u_2 + E_2 \Delta s_0^y + E_2^\theta \Delta\theta. \tag{4.30b}$$

The submatrices of system (4.30) are computed in the following way, see Schlöder [85],

$$E_i = \hat{D}_0^i + \sum_{k=1}^{n_{\mathrm{ms}}} \hat{D}_k^i \prod_{j=1}^{k} \hat{G}_{k-j}, \qquad i = 1, 2, \tag{4.31a}$$

$$E_i^p = \hat{D}_\theta^i + \sum_{k=1}^{n_{\mathrm{ms}}} \hat{D}_k^i \left(\sum_{j=0}^{k-1} \left(\prod_{l=1}^{k-1-j} \hat{G}_{k-l} \right) \hat{G}_j^\theta \right), \qquad i = 1, 2 \tag{4.31b}$$

$$u_1 = \hat{F}_1 + \sum_{k=1}^{n_{\mathrm{ms}}} \hat{D}_k^1 \left(\sum_{j=0}^{k-1} \left(\prod_{l=1}^{k-1-j} \hat{G}_{k-l} \right) \hat{c}(\tau_k) \right), \tag{4.31c}$$

$$u_2 = \hat{r} + \sum_{k=1}^{n_{\mathrm{ms}}} \hat{D}_k^2 \left(\sum_{j=0}^{k-1} \left(\prod_{l=1}^{k-1-j} \hat{G}_{k-l} \right) \hat{c}(\tau_k) \right), \tag{4.31d}$$

with $\prod_{k=i}^{j} A_k = I$ for $j < i$ and $\prod_{k=i}^{j} A_k = A_i \cdot A_{i+1} \cdot \ldots \cdot A_j$.

After we have solved (4.30) for Δs_0^y and $\Delta\theta$, the remaining increments are computed by forward recursion

$$\Delta s_k^z = -\left(H_k^z\right)^{-1}\left[H_k^y \Delta s_k^y + H_k^\theta \Delta\theta + g(\tau_k, s_k, \theta)\right], \qquad k = 0, \dots, n_{\text{ms}}, \tag{4.32a}$$

$$\Delta s_{k+1}^y = \hat{G}_k \Delta s_k^y + \hat{G}_k^\theta \Delta\theta + \hat{c}(\tau_{k+1}), \qquad k = 0, \dots, n_{\text{ms}} - 1. \tag{4.32b}$$

The common approach, e.g., see Bock [12], Körkel [58], is to compute all submatrices D, G and H of the Jacobian (4.27) and the right-hand side. Afterwards, the condensing algorithm is applied.

For the computation of the right hand side, we need to compute the nominal trajectory of the differential algebraic equation (4.1) while $n_y + n_z + n_\theta$ additional variational differential equations have to be evaluated for the evaluation of the derivatives. The effort increases with the number of state variables and parameters. For high-dimensional problems this method becomes very expensive.

In Chapter 6, we develop an approach to lower the computational effort.

4.3.5 Restricted monotonicity test

The convergence results presented in Section 4.3.1 especially in Lemma (4.3.4) hold only for small residuals, i.e., if we start sufficiently close to the solution. If this is not the case, we need to apply a globalization strategy.

We suggest the application the restricted monotonicity test (RMT) developed by Bock et al. [16]. The computation of the quantities that are necessary for the evaluation of the RMT are very costly. Here, we illustrate the practical realization.

The costly evaluation of the curvature information is approximated by

$$\omega(\alpha) = \frac{2\left\|J(v^l)^+\left(F(v^l + \alpha\Delta v^l) - (1-\alpha)F(v^l)\right)\right\|}{\alpha^2 \|v^l\|^2}. \tag{4.33}$$

The evaluation of (4.33) involves the calculation of the current increment

$$\Delta v^l = -J^+(v^l)F(v^l),$$

what has already been done, and of

$$\Delta \bar{v}^l = -J^+(v^l)F(v^l + \alpha v^l),$$

which we would have had to compute for a line search strategy, too. For the step length α^k, we demand the less restrictive condition

$$\delta_* \leqslant \alpha^l \cdot \omega(\alpha^l)\left\|\Delta v^l\right\| \leqslant \delta^* \tag{4.34}$$

to hold, with $\delta_* < \delta < \delta^*$ for a prescribed constant $\delta < 2$.

By a root finding strategy for

$$\alpha \cdot \omega(\alpha) \left\| \Delta v^l \right\| - \delta = 0,$$

we determine an α^k such that condition (4.34) is fulfilled. Bock et al. [16] suggest

$$\alpha^l_{\text{start}} := \min \left(1, \frac{\delta}{\omega(\alpha^{l-1}) \left\| \Delta v^l \right\|)} \right)$$

as a good starting value for the computation of α^l, where we use the curvature information of the previous iteration.

Though there is no proof of global convergence, the RMT shows promising results in practical applications and does not lead to iteration cycling for the well known example by Ascher and Osbourne [4].

Furthermore, in the neighborhood of the solution this approach leads to full step iterations ($\alpha = 1$) and the application of the restricted monotonicity test does not result in convergence of the generalized Gauss–Newton towards statistically instable solutions.

The algorithm for the practical realization of the restricted monotonicity test looks the following:

Algorithm 4.3.6. *Let the current iterate v^l, the step for the Gauss–Newton iteration Δv^l and the previous curvature information ω^{l-1} be given.*

1. *Set $k = 0$ and compute a candidate for the step length $\alpha^l_k = \min\left(1, \frac{1}{\omega^{l-1}\|\Delta v^l\|)}\right)$*
2. *Calculate the simplified increment $\Delta\bar{v}^l = -J^+(v^l)F(v^l + \alpha^l_k \Delta v^l)$*
3. *Compute the curvature information $\omega^l_k = \frac{2\|\Delta\bar{v}^l - (1-\alpha^l_k)v^l)\|}{\left(\alpha^l_k\right)^2 \|v^l\|^2}$*
4. *If the monotonicity test*

$$\delta_* \leqslant \alpha^l_k \cdot \omega^l_k \left\| \Delta v^l \right\| \leqslant \delta^*$$

is satisfied, update $v^{l+1} = v^l + \alpha^l_k \Delta v^l$ and set $\omega^l = \omega^l_k$. Otherwise, set $\omega^{l-1} = \omega^l_k$ go to step 1.

4.4 Parameter estimation for microbial enhanced oil recovery

Since a partial differential equation can be considered as a system of DAEs after being discretized by a method of lines, all the results and methods presented above are valid for

PDE constrained parameter estimation problems as well. A similar approach for adjoint derivatives can be found in Weiler [98].

As mentioned in Section 4.1.2, we examine model responses of the form:

$$h_j(t_j, \xi_j(\theta), \theta), \quad j = 1, \ldots, m$$

where the states ξ_j are computed by solving the general PDE model equations (2.21) from Section 2.5

$$\mathscr{F}_j(t, \xi_j(\theta), \theta) = 0. \tag{4.35}$$

Here, we omit the dependencies on the controls. The index j of $\mathscr{F}$ symbolizes that the configurations, e.g., the controls, are the same as in the experimental setup for obtaining η_j. In every iteration of the generalized Gauss–Newton method, we solve (4.35) applying the methods presented in Chapter 3. For the computation of the Jacobians (4.18a), we need to evaluate the derivatives of the model response h and the model equations $\mathscr{F}$ with respect to states ξ and parameters θ

$$\partial_\xi h,\ \partial_\theta h,\ \partial_\xi \mathscr{F},\ \partial_\theta \mathscr{F}.$$

As mentioned before, for the computation of the derivatives we choose algorithmic differentiation in forecast of the optimum experimental design problems, we want to solve, where second order derivatives are required. Computing the Jacobians (4.18a) requires the differentiation of h in every unit direction of $\mathbb{R}^{n_\theta}$. To determine n_θ parameters without regularization for the number of measurements, it has to hold $m \geqslant n_\theta$. Griewank [46] has shown that in this case the forward mode of AD is suitable. Then computing $\partial_\theta h_j$ results in

$$\frac{dh_j}{d\theta} = \frac{\partial h_j}{\partial \xi}\xi_{\theta,j} + \frac{\partial h_j}{\partial \theta},$$

where $\xi_{\theta,j} \in \mathbb{R}^{\hat{n} \times n_\theta}$, $j = 1, \ldots, m$ and $\hat{n}$ is the number of states. The sensitivities are derived by solving the variational differential equation

$$0 = \frac{\partial \mathscr{F}_j}{\partial \xi}\xi_{\theta,j} + \frac{\partial \mathscr{F}_j}{\partial \theta}. \tag{4.36}$$

The VDEs are of the same type as our model equations (2.21). Therefore, it is reasonable to apply the methods presented in Chapter 3 for the solution of the variational differential equations (4.36), too. We exert a mixed discontinuous Galerkin finite element method for the discretization in space and apply corresponding formulations of Algorithms 3.2.1 and 3.4.1 for the solution in time.

5 Optimum experimental design

Usually, the prediction of variance–covariance matrix depends on control variables $q \in \mathbb{R}^{n_q}$, and control functions, $u(t) \in \mathbb{R}^{n_u}$, e.g., temperature and feeding rates, and the choice of measurement points. By varying the influencing factors, we design experiments that allow an improved estimation of parameters.

By the methods of experimental design, we try to determine an experimental setup which permits the best identification of the parameters with respect to a later defined objective function. We follow the approaches of Körkel [58] and Lohmann et al. [67]. We state the general problem formulation. We introduce a sequential quadratic programming method and explain how the required derivatives are evaluated.

For the computations, we use the software package *VPLAN* developed by Körkel [58]. VPLAN offers the possibility to exploit structures that arise when multiple experiments are designed, an interface to *ADIFOR* by Bischof et al. [10] for the evaluation of the derivatives by algorithmic differentiation and an interface to *SNOPT* by Gill et al. [44] to solve the optimum experimental design problem.

In this chapter, we consider unconstrained parameter estimation problems, i.e., $F = F_1$, see Equation (4.22).

5.1 The general problem formulation

As mentioned before, the prediction of the variance–covariance matrix depends on the control variables $q \in \mathbb{R}^{n_q}$, that are constant in time and the control functions $u(t) \in \mathbb{R}^{n_u}$, that vary over time. Constraints may be given for both of them. In the form of lower and upper bounds

$$\begin{aligned} q_i &\in [\underline{q}_i, \overline{q}_i], \ i = 1, \dots, n_q, \\ u_i(t) &\in [\underline{u}_i, \overline{u}_i], \ i = 1, \dots, n_u \end{aligned}$$

or as nonlinear control constraints

$$c(q, u(t)) \in [\underline{c}, \overline{c}]$$

with $c : \mathbb{R}^{n_q + n_u} \to \mathbb{R}^{n_c}$.

Besides the choice of the controls, the quality of the parameter estimation depends on the selection of measurement points. Due to various given constraints, it may not be possible to measure whenever and as often as we want to. Therefore, it is necessary to determine the optimal placement for the available measurement methods.

Consider a set of possible time points, $t_0 \leqslant t_1 \leqslant \ldots \leqslant t_m \leqslant t_{end}$, where a measurement may be taken with corresponding model response $h_j(t_j, \xi(t_j), \theta, q),\ j = 1, \ldots, m$. Thereby, it is possible that two or more time points are equal or that we have the same measurement functions at different time points. For each possible measurement, we define a weight

$$w_j \in \{0,1\},\ j = 1, \ldots, m \tag{5.1}$$

with $w_j = 1$ if the measurement is taken and $w_1 = 0$ else. Similar to the controls, lower and upper bounds can be given to model the minimal and maximal number of measurements per time point, measurement type etc.

$$\underline{w}_k \leqslant \sum_{j \in J_k} w_j \leqslant \overline{w}_k,\ k = 1, \ldots, n_J \tag{5.2}$$

for suitable index sets $J_k \subseteq \{1, \ldots, m\},\ k = 1, \ldots, n_J$. In addition, we may have to deal with cost constraints. If we assume a linear cost model with cost coefficients c_j^w for the measurement j, the total measurement costs are described by

$$\sum_{j=1}^{m} c_j^w \cdot w_j.$$

We summarize all possible constraints in the form

$$c_i(q, u(t), w) = 0,\ i \in \mathscr{E}, \tag{5.3a}$$

$$c_i(q, u(t), w) \geqslant 0,\ i \in \mathscr{I}. \tag{5.3b}$$

The predicted variance–covariance matrix at the solution $\hat{\theta}$ of the parameter estimation problem

$$C = \left(J^T J\right)^{-1},$$

is a function of the design variables q, u and w.
We use information functions ϕ to rate the quality of the parameter estimation, see Pukelsheim [80]. The most common are:

- The trace criterion (A-criterion):

$$\phi_A(C) = \frac{1}{n_\theta} \mathrm{trace}(C).$$

- The determinant criterion (D-criterion):

$$\phi_D(C) = \left(\det(C)\right)^{n_\theta}.$$

- The largest eigenvalue criterion (E-criterion):

$$\phi_E(C) = \max\{\lambda : \lambda - \text{ eigenvalue of } C\}.$$

Another possibility is to use the approximation of the confidence intervals of the parameters, see Theorem 4.3.5:

- The confidence interval criterion (M-criterion):

$$\phi_M(C) = \max\{\sqrt{C_{ii}},\ i = 1,\dots,n_\theta\}.$$

Using the confidence ellipsoid (4.25), we interpret the different criteria geometrically. The A-criterion is proportional to the average length of the semiaxes of the confidence ellipsoid, the D-criterion to its volume, the E-criterion to the squared length of the semimajor axis and the M-criterion to the side length of a bounding box around the ellipsoid, see Figure 5.1.

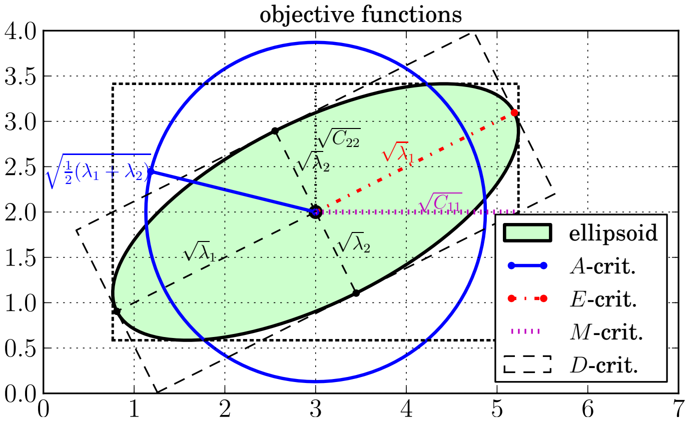


Figure 5.1: Visualization of the optimum experimental design criteria taken from Walter [97].

Then, the nonlinear optimum experimental design problem looks as follows:

$$\min_{q,u,w,x} \quad \phi(C) \tag{5.4a}$$

with $C = (J^T J)^{-1}$ and the Jacobian J evaluated at the solution of the parameter estimation problem $\min\limits_{\theta} \frac{1}{2}\|F(\theta)\|_2^2$

$$\text{s.t.} \quad 0 = \mathscr{F}_j(t, \xi(\theta), \theta, q, u(t)), \tag{5.4b}$$

$$0 = \frac{\partial \mathscr{F}_j}{\partial \xi}\xi_{\theta,j} + \frac{\partial \mathscr{F}_j}{\partial \theta}, \qquad j = 1,\dots,m, \tag{5.4c}$$

$$0 = c_i(q, u(t), w), \qquad i \in \mathscr{E}, \tag{5.4d}$$

$$0 \leqslant c_i(q, u(t), w), \qquad i \in \mathscr{I}, \tag{5.4e}$$

$$w \in \{0,1\}^m, \tag{5.4f}$$

where $\mathscr{F}_j$ are the underlying model equations as defined in (4.35).

5.2 Sequential quadratic programming

We follow the notations of Nocedal and Wright [77] throughout this section. Körkel [58] presented how the infinite dimensional problem (5.4) is transformed into a finite dimensional one. Thereby, the control functions u are discretized by piecewise constant, piecewise linear or the like polynomial approximations and the integral measurement weights w are relaxed to the interval $w \in [0,1]^m$.

Let $\hat{u}$ be the discretized version of u and let $\hat{w}$ be the relaxation of w. By assuming our model equations to be continuously differentiable with respect to $q, \hat{u}$ and $\hat{w}$, we rewrite (5.4) in the standard form of nonlinear optimal control problems

$$\min_{\hat{q}} \quad \phi(\hat{q}) \tag{5.5a}$$

$$\text{s.t.} \quad 0 = c_i(\hat{q}), \quad i \in \mathscr{E}, \tag{5.5b}$$

$$0 \leqslant c_i(\hat{q}), \quad i \in \mathscr{I}, \tag{5.5c}$$

with the setting $\hat{q} = (q, \hat{u}, \hat{w})$ and continuously differentiable functions ϕ and c_i. We define the Lagrangian

$$\mathscr{L}(\hat{q}) = \phi(\hat{q}) - \sum_{i \in \mathscr{E}} \lambda_i c_i(\hat{q}) - \sum_{i \in \mathscr{I}} \mu_i c_i \tag{5.6}$$

with Lagrange multipliers $\lambda_i \in I\!R$, $i \in \mathscr{E}$, and $\mu_i \in I\!R^+$, $i \in \mathscr{I}$.

Definition 5.2.1. *(Nocedal and Wright [77]) The active set $\mathscr{A}(\hat{q})$ at any feasible point $\hat{q}$ consists of the equality constraint indices from $\mathscr{E}$ together with the indices of the inequality constraints i for which $c_i(\hat{q}) = 0$; that is,*

$$\mathscr{A}(\hat{q}) = \mathscr{E} \cup \{ j \in \mathscr{I} \mid c_i(\hat{q}) = 0 \}.$$

At a feasible point $\hat{q}$, the inequality constraint $i \in \mathscr{I}$ is said to be active if $c_i(\hat{q}) = 0$ and inactive if the strict inequality $c_i(\hat{q}) > 0$ is satisfied. We define a condition on the active set to guarantee the existence of a solution of (5.5).

Definition 5.2.2. *If the set*

$$\{ \nabla c_i(\hat{q}) : \, i \in \mathscr{A}(\hat{q}) \}$$

is linearly independent, then the linear independence constraint qualification (LICQ) condition holds.

With Definitions 5.2.1 and 5.2.2, we state the first-order necessary conditions.

Theorem 5.2.3. *Let $\hat{q}^*$ be a local solution of* (5.5) *and let the functions ϕ and c_j be continuously differentiable. Furthermore, suppose that (LICQ) holds at $\hat{q}^*$. Then, there are*

Lagrange multipliers $\lambda^* = (\lambda_i)_{i\in\mathscr{E}}$ *and* $\mu^* = (\mu_i)_{i\in\mathscr{I}}$ *such that the following conditions are satisfied*

$$
\begin{aligned}
0 =&\nabla_{\hat{q}^*}\mathscr{L}(\hat{q}^*,\lambda^*,\mu^*), && && (5.7a)\\
0 =&c_i(\hat{q}^*), && \forall i\in\mathscr{E}, && (5.7b)\\
0 \leqslant&c_i(\hat{q}^*), && \forall i\in\mathscr{I}, && (5.7c)\\
0 \leqslant&\mu_i^*, && \forall i\in\mathscr{I}, && (5.7d)\\
0 =&\mu_i^* c_i(\hat{q}^*), && \forall i\in\mathscr{I}. && (5.7e)
\end{aligned}
$$

For a proof, we refer to Nocedal and Wright [77]. Conditions (5.7) are referred to as the Karush–Kuhn–Tucker (KKT) conditions. With the set of linearized feasible directions

$$\mathscr{D} := \{d:\ d^T\nabla c_i(\hat{q}) = 0,\ \forall\ i\in\mathscr{E} \text{ and } d^T\nabla c_i(\hat{q}) \geqslant 0, \forall\ i\in\mathscr{A}(\hat{q})\cap\mathscr{I}\}$$

and the critical cone

$$\mathscr{C}(\hat{q}^*,\lambda^*,\mu^*) := \{\hat{d}\in\mathscr{D}(\hat{q}^*):\ \nabla c_i(\hat{q}^*)^T\hat{d} = 0, \forall\ i\in\mathscr{A}(\hat{q}^*)\cap\mathscr{I}\}$$

we formulate second-order sufficient conditions for twice continuously differentiable functions ϕ and c_i :

Theorem 5.2.4. *Let the KKT conditions* (5.7) *be satisfied for some feasible point* $\hat{q}^*$ *and Lagrange multipliers* λ^* *and* μ^**. If*

$$\hat{d}^T\nabla^2_{\hat{q}^*\hat{q}^*}\mathscr{L}(\hat{q}^*,\lambda^*,\mu^*)\hat{d} > 0, \quad \forall d\in\mathscr{C}(\hat{q}^*,\lambda^*,\mu^*),\ d\neq 0,$$

then $\hat{q}^*$ *is a strict local solution for* (5.5).

We solve problem (5.4) by applying a sequential quadratic programming (SQP) method, which is a powerful approach for solving nonlinear constrained optimal control problems. SQP methods are treated in many textbooks concerning nonlinear optimization, see for instance Fletcher [39], McCormick [73], Nocedal and Wright[77], and Ulbrich [95]. Furthermore, there is a broad variety of practical implementations at hand. We use the software package SNOPT, see Gill et al. [43, 44]. The idea is to solve problem (5.5) iteratively. As the name suggests, we have to solve a quadratic subproblem in each iteration whose objective function is the sum of the gradient of ϕ and the Hessian of the Lagrangian function (5.6) and whose constraints are linearizations of the nonlinear constraints (5.5b) and (5.5c).

The following algorithm describes the basic approach of the SQP method.

Algorithm 5.2.5. ***(SQP method)***

1. *Set* $k := 0$. *Start with an initial guess* $(\hat{q}^0,\lambda^0,\mu^0)$.

2. *Compute the function values*

$$\phi^k := \phi(\hat{q}^k), \quad c_i^k := c(\hat{q}^k), \quad i \in \mathscr{E} \cup \mathscr{I}$$

the gradients

$$\nabla\phi^k := \nabla\phi(\hat{q}^k), \quad \nabla c^k := \nabla c(\hat{q}^k), \quad i \in \mathscr{E} \cup \mathscr{I}$$

and an approximation of the Hessian of the Lagrangian function

$$H^k \approx \nabla_{\hat{q}}^2 \mathscr{L}(\hat{q}^k, \lambda^k, \mu^k). \tag{5.8}$$

3. *Determine the increment d^k as solution of the quadratic program*

$$\begin{aligned} \min_d \quad & \tfrac{1}{2} d^T H^k d + \nabla\phi^{k^T} d \\ s.t. \quad 0 =& c^k + \nabla\hat{q}^{k^T} d, \qquad i \in \mathscr{E} \\ 0 \leqslant& c^{k^T} + \nabla c^{k^T} d, \qquad i \in \mathscr{I}. \end{aligned}$$

Compute the corresponding multipliers $\tilde{\lambda}^k$ and $\tilde{\mu}^k$.

4. *Determine the step size $\alpha^k \in [0,1]$ and update:*

$$\hat{q}^{k+1} := \hat{q}^k + \alpha^k \cdot d^k, \tag{5.9a}$$

$$\lambda^{k+1} := \lambda^k + \alpha^k(\tilde{\lambda}^k - \lambda^k), \tag{5.9b}$$

$$\mu^{k+1} := \mu^k + \alpha^k(\tilde{\mu}^k - \mu^k). \tag{5.9c}$$

5. *If some convergence criterion is not satisfied continue with step 2 and $k := k+1$.*

If the exact Hessian is used in (5.8), the SQP method is locally equivalent to Newton's method and converges locally quadratically. In general, we will not compute the exact Hessian. For large-scale problems, its computation is too costly for an efficient treatment of the problem. One of the most common Hessian approximation strategies is the Broyden–Fletcher–Goldfarb–Shanno (BFGS) update. BFGS is a symmetric rank two update for which one can proof superlinear convergence using the theorem of Dennis and Moré [30].

In SNOPT, a limited memory BFGS, where only a history of the past $\hat{n}$ updates is maintained, together with an Active-Set method is implemented. For further details on Active-Set methods, see Nocedal and Wright [77]. For the determination of the step size α^k in (5.9) by line search methods for the globalization of convergence, we refer to the literature as well, cf. Nocedal and Wright [77], and Geiger and Kanzow [41]. As convergence criterion, we choose, that the KKT conditions are satisfied up to a predefined tolerance, i.e., the first order necessary conditions are satisfied up to the predefined tolerance.

5.3 Derivatives

Since SNOPT uses BFGS updates to approximate the Hessian of the Lagrange function, we need to compute only the gradient of the objective ϕ and the constraints $c_i,\ i \in \mathscr{E} \cup \mathscr{I}$ of problem (5.5). The formulas for the directional derivatives of ϕ are taken from Körkel [58]. A directional derivative with direction δx is defined by

$$\frac{df}{dx}\delta x := \lim_{h\to 0} \frac{f(x+h\delta x)-f(x)}{h}.$$

The derivatives of the information function ϕ in direction $\delta C \in \mathbb{R}^{n_\theta \times n_\theta}$ of the variance–covariance matrix $C \in \mathbb{R}^{n_\theta \times n_\theta}$ of the optimum experimental design problem are

(i) A-criterion:

$$\begin{aligned}\frac{d\phi_A}{dC} &= \frac{1}{n_\theta}\frac{d\,\mathrm{trace} C}{dC}\delta C = \frac{1}{n_\theta}\sum_{i,j}\frac{d\,\mathrm{trace} C}{dC_{ij}}\delta C_{ij} = \frac{1}{n_\theta}\sum_{i,j,k}\frac{d\,\mathrm{trace} C_{kk}}{dC_{ij}}\delta C_{ij}\\ &= \frac{1}{n_\theta}\sum_{i,j,k}\delta_{ki}\delta_{kj}\delta C_{ij} = \frac{1}{n_\theta}\delta C_{kk} = \frac{1}{n_\theta}\mathrm{trace}\delta C.\end{aligned}$$

(ii) D-criterion: With

$$\frac{d}{dC}\det(C)\delta C = \sum_{i,j}\det(C)(C^{-1})_{ij}\delta C_{ij}$$

we have

$$\begin{aligned}\frac{d\phi_D}{dC}\delta C &= \frac{d}{dC}\delta C(\det(C))^{\frac{1}{n_\theta}}\delta C\\ &= \frac{1}{n_\theta}(\det(C))^{\frac{1}{n_\theta}}\sum_{i,j}\left(C^{-1}\right)_{ij}\delta C_{ij}.\end{aligned}$$

(iii) If the largest eigenvalue has multiplicity one with corresponding normalized eigenvector v, then

$$\frac{d\phi_E}{dC}\delta C = \frac{d}{dC}\max\{\lambda :\ \lambda \text{ eigenvalue of } C\}\delta C = v^T\delta C v.$$

(iv) M-criterion: We transform the problem

$$\min \phi_M(C) = \min\max\{\sqrt{C_{ii}}, i = 1,\dots,n_\theta\}$$

into an auxiliary problem

$$\begin{array}{rl}\min & \phi_0\\ \text{s.t.} & \phi_0 \leqslant \sqrt{C_{ii}},\ i = 1,\dots,n_\theta.\end{array}$$

The constraints are added to the inequality constraints $c_i,\ i \in \mathscr{I}$ in (5.4). The derivative is then given by

$$\frac{d}{dC}(\sqrt{C_{ii}})\delta C = \frac{1}{2\sqrt{C_{ii}}}\delta C.$$

We state the following lemma which is needed for the evaluation of the derivative of the variance–covariance matrix.

Lemma 5.3.1. *For a regular matrix* $A \in I\!R^{n_\theta \times n_\theta}$ *and* $\delta A \in I\!R^{n_\theta \times n_\theta}$ *it holds*

$$\frac{dA^{-1}}{dA}\delta A = -A^{-1}\delta A A^{-1}.$$

Proof. Define $F : I\!R^{n_\theta \times n_\theta} \times I\!R \to I\!R^{n_\theta \times n_\theta}$ as

$$F(B,h) := (A + h\delta A)B - I_{n_\theta}.$$

F is continuously differentiable and it holds

(i) $F(A^{-1},0) = 0,$

(ii) $\frac{\partial F}{\partial B}(A^{-1},0) = A$ is regular.

By applying the implicit function theorem, there exists a continuously differentiable function B with

(i) $B(h) = (A + h\delta A)^{-1},$

(ii) $B'(h) = -\left(\frac{\partial F}{\partial B}(B(h),h)\right)^{-1} \frac{\partial F}{\partial h}(B(h),h) = -(A + h\delta A)^{-1}\delta A B(h).$

Hence it follows

$$\frac{dA^{-1}}{dA}\delta A = \lim_{h\to 0}\frac{B(h) - B(0)}{h} = B'(0) = -A^{-1}\delta A A^{-1}.$$

□

Following the conclusion of Lemma 5.3.1, the derivative of the variance-covariance matrix

$$C = \left(J^T J\right)^{-1}$$

in direction δJ is

$$\frac{dC}{dJ}\delta J = \frac{d\left(J^T J\right)^{-1}}{dJ}\delta J = -\left(J^T J\right)^{-1}\left(\delta J^T J + J^T \delta J\right)\left(J^T J\right)^{-1}.$$

We recall the notation for the Jacobian from (4.18a)

$$J_{ji} = -\frac{1}{\sigma_j}\frac{dh_j}{d\theta_i}(\xi(\hat{q}),\hat{q}).$$

Then, the gradient of the objective ϕ is determined by the derivatives

$$\frac{d}{d\hat{q}}\left(\frac{dh_j}{d\theta}\right) = \frac{d}{d\hat{q}}\left(\frac{\partial h_j}{\partial \xi}\xi_{\theta,j} + \frac{\partial h_j}{\partial \theta}\right)$$

$$= \left(\frac{\partial^2 h_j}{\partial \xi \partial \xi} \xi_{\hat{q},j} + \frac{\partial^2 h_j}{\partial \xi \partial \hat{q}} \right)^T \xi_{\theta,j} + \frac{\partial h_j}{\partial \xi} \xi_{\theta \hat{q},j} + \frac{\partial^2 h_j}{\partial \theta \partial \xi} \xi_{\hat{q},j} + \frac{\partial^2 h_j}{\partial \theta \partial \hat{q}}.$$

The first order sensitivities $\xi_{\hat{q},j}$ are derived by the variational differential equations

$$0 = \frac{\partial \mathscr{F}_j}{\partial \xi} \xi_{\hat{q},j} + \frac{\partial \mathscr{F}_j}{\partial \hat{q}}, \tag{5.10}$$

while second order sensitivities $\xi_{\theta \hat{q},j}$ are given by second order VDEs

$$0 = \left(\frac{\partial^2 \mathscr{F}_j}{\partial \xi \partial \xi} \xi_{\hat{q},j} + \frac{\partial^2 \mathscr{F}_j}{\partial \xi \partial \hat{q}} \right)^T \xi_{\theta,j} + \frac{\partial \mathscr{F}_j}{\partial \xi} \xi_{\theta \hat{q},j} + \frac{\partial^2 \mathscr{F}_j}{\partial \theta \partial \xi} \xi_{\hat{q},j} + \frac{\partial^2 \mathscr{F}_j}{\partial \theta \partial \hat{q}} \tag{5.11}$$

with $\xi_{\theta,j}$ and $\xi_{\hat{q},j}$ computed by (4.36) and (5.10) respectively.

As presented in section 4.4, the derivatives of model response

$$\partial_\xi h,\ \partial_\theta h,\ \partial_{\hat{q}} h,\ \partial^2_{\xi\xi} h,\ \partial^2_{\theta\xi} h,\ \partial^2_{\xi,\hat{q}} h,\ \partial^2_{\theta,\hat{q}} h,$$

the model equations

$$\partial_\xi \mathscr{F},\ \partial_\theta \mathscr{F},\ \partial_{\hat{q}} \mathscr{F},\ \partial^2_{\xi\xi} \mathscr{F},\ \partial^2_{\theta\xi} \mathscr{F},\ \partial^2_{\xi,\hat{q}} \mathscr{F},\ \partial^2_{\theta,\hat{q}} \mathscr{F},$$

and the constraints

$$\nabla c_i$$

are evaluated with the algorithmic differentiation tool ADIFOR.

Similar to (4.36), Equations (5.10) and (5.11) feature the same structure as the underlying model equations (2.21) and we apply an analog mixed DG-FEM formulation to the variational differential equations (5.10) and (5.11), respectively. The extended IMPES Algorithms 3.2.1 and 3.4.1 adopted to the weak formulation afterwards.

6 Efficient parameter estimation

In this chapter, we revisit an approach that reduces the increasing demand of computation time for high-dimensional parameter estimation problems in the context of multiple shooting. The *reduced approach* was developed by Schlöder [85] for parameter estimation problems constrained by high-dimensional ODE systems. A first extension to DAE models was presented by Bauer [7]. We consider constraints of partial differential equations as the origin of the DAE constraints and develop of different formulation of the reduced approach. A short introduction to the method and a first application to PDE constraints parameter estimation can be found in Kircheis and Körkel [55].

We derive the reduced approach from the condensed system and illustrate the efficient computation of the increment for the generalized Gauss–Newton method. We introduce the new parameter estimation software *PAREMERA* and the implemented algorithms. The last sections provides numerical results of two academic examples.

6.1 The reduced approach

We recall the finite dimensional parameter estimation problem (4.15) for n_{ex} executed experiments from Section 4.2

$$\min_{\theta,s} \quad \tfrac{1}{2}\left(\eta - h(\psi(s,\theta),\theta\right)^T \Sigma^{-2} \left(\eta - h(\psi(s,\theta),\theta)\right) \tag{6.1a}$$

$$\begin{aligned}
\text{s.t.}\quad 0 =& c^i(\tau^i_{k+1}, s^i_k, s^i_{k+1}, \theta), && k = 0,\dots,n^i_{\text{ms}} - 1 && \text{(6.1b)}\\
0 =& g^i(\tau^i_k, s^i_k, \theta), && k = 0,\dots,n^i_{\text{ms}} && \text{(6.1c)}\\
& && i = 1,\dots,n_{\text{ex}} &&\\
0 =& y^i_0(\theta) - s^{y,i}_0 && && \text{(6.1d)}\\
0 =& \sum_{i=1}^{n_{\text{ex}}} \sum_{k=1}^{m^i_r} r^i_k\left(\psi^i(t^i_k),\theta\right). && && \text{(6.1e)}
\end{aligned}$$

with the interval-wise description of the nominal trajectory

$$\begin{pmatrix} y^i(t) \\ z^i(t) \end{pmatrix} = \psi^i(t; s_k, \theta), \quad t \in [\tau^i_k, \tau^i_{k+1}), \quad k = 0,\dots,n^i_{\text{ms}}, \quad i = 1,\dots,n_{\text{ex}},$$

the continuity conditions for the differential states

$$0 = c^i(\tau^i_{k+1}, s^i_k, s^i_{k+1}, \theta) := \psi^{i,y}(\tau^i_{k+1}; s^i_k, \theta) - s^{y,i}_{k+1}, \quad k = 0, \ldots, n^i_{\text{ms}} - 1, \quad i = 1, \ldots, n_{\text{ex}}, \tag{6.2}$$

and the consistency conditions for the algebraic states

$$0 = g^i(\tau^i_k, s^i_k, \theta), \quad k = 0, \ldots, n^i_{\text{ms}}, \quad i = 1, \ldots, n_{\text{ex}}. \tag{6.3}$$

We consider a linear-implicit DAE

$$A^i(t, \psi^i(t), \theta)\dot{y}^i(t) = f^i(t, \psi^i(t), \theta), \quad \psi^{y,i}(\tau_k) = s^y_k \tag{6.4a}$$

$$0 = g^i(t, \psi^i(t), \theta), \quad t \in [\tau_k, \tau_{k+1}], \quad k = 0, \ldots, n_{\text{ms}}, \quad i = 1, \ldots, n_{\text{ex}}, \tag{6.4b}$$

as the approximation of a parametrized partial differential equation, e.g. discretized by a methods of lines.

In comparison with the parameter estimation problem (4.15) in Chapter 4, we examine a slightly different formulation of the interior point constraints. In (6.1e), we consider linearly coupled constraints that allow us to model conditions which interconnect two or more experiments. Note, that the interior point constraints (4.14)

$$0 = \sum_{k=1}^{m^i_r} r^i_k\left(\psi^i(t^i_k), \theta\right), \quad i = 1, \ldots, n_{\text{ex}}.$$

are a special case of (6.1e).

By using Equation (6.1e), e.g., it is possible to model matching conditions between single experiments of the general form

$$r(y^i(t^i_{end}), y^{i+1}(t^{i+1}_0)) = 0, \tag{6.5}$$

where the superscript i refers to the different experiments, to model discontinuities in the state equations

$$\dot{y}(t) = \begin{cases} f_1(t, y(t), \theta), & t \leqslant \hat{t}, \\ f_2(t, y(t), \theta), & t > \hat{t} \end{cases},$$

or even jumps of the trajectories. We refer to equation (6.1e) as multi-experiment interior point constraints.

We introduce new variables s^r that arise from the parametrization of the model equations. In Section 4.3, we demand that the constraint qualification condition holds, i.e., the Jacobian of the constraints J_2 is regular. Under the assumption that we add the initial conditions of the differential states (6.1d), that may depend on the variables s^r, to the multi-experiment interior point constraints, the Equations (6.2), (6.3) and (6.1e) sum up to

$$\sum_{i=1}^{n_{\text{ex}}} n^i_{\text{ms}} \cdot n^i_y + \sum_{i=1}^{n_{\text{ex}}} \left(n^i_{\text{ms}} + 1\right) \cdot n^i_z + n_r + \sum_{i=1}^{n_{\text{ex}}} n^i_y = n_r + \sum_{i=1}^{n_{\text{ex}}} \left(n^i_{\text{ms}} + 1\right) \cdot \left(n^i_y + n^i_z\right)$$

conditions. Since we have

$$n_\theta + \sum_{i=1}^{n_{\text{ex}}} \left(n_{\text{ms}}^i + 1\right) \cdot \left(n_y^i + n_z^i\right)$$

degrees of freedom, the Jacobian J_2 has full rank by all means if we establish n_r new variables s^r. The variables s^r are defined by the condition

$$\frac{\partial}{\partial s^r} \sum_{i=1}^{n_{\text{ex}}} \sum_{k=1}^{m_r^i} r_k^i \left(\psi^i(t_k^i), \theta, s^{r,i}\right) - \text{regular},$$

and add the variables s^r to the dependencies of the multi-experiment interior point constraints. Thus, it is ensured that CQ holds.

For further examinations, we add the variables s^r to the vector s and define

$$d(s,\theta) := \begin{pmatrix} \left(y_0^i(\theta, s^{r,i}) - s_0^{y,i}\right)_{i=1,\dots,n_{\text{ex}}} \\ \sum_{i=1}^{n_{\text{ex}}} \sum_{k=1}^{m_r^i} r_k^i \left(\psi^i(t_k^i), \theta, s^{r,i}\right). \end{pmatrix}. \tag{6.6}$$

Note, that the initial conditions (6.1d) may depend on the variables s^r, too. Again we introduce a shorter notation of problem (6.1) to increase readability

$$\min_{s,\theta} \quad \tfrac{1}{2} \|F_1(s,\theta)\|_2^2 \tag{6.7a}$$

$$\text{s.t.} \quad 0 = F_2(s,\theta) \tag{6.7b}$$

with the settings

$$F_1 := \Sigma^{-1}\left(\eta - h(\xi(s,\theta),\theta)\right) \tag{6.8a}$$

$$F_2 := \begin{pmatrix} \left(c^i(\tau_{k+1}^i, s_k^i, s_{k+1}^i, \theta)\right)_{\substack{k=0,\dots,n_{\text{ms}}^i-1 \\ i=1,\dots,n_{\text{ex}}}} \\ \left(g^i(\tau_k^i, s_k^i, \theta)\right)_{\substack{k=0,\dots,n_{\text{ms}}^i \\ i=1,\dots,n_{\text{ex}}}} \\ d(s,\theta) \end{pmatrix}. \tag{6.8b}$$

As described in Section 4.3, the solution of (6.7) is computed iteratively by means of a generalized Gauss–Newton method where we iteratively solve a series linearized problems instead

$$\min_{\Delta s, \Delta\theta} \quad \frac{1}{2} \left\| F_1(s,\theta) + J_1(s,\theta) \begin{pmatrix} \Delta s \\ \Delta\theta \end{pmatrix} \right\|_2^2 \tag{6.9a}$$

$$\text{s.t.} \quad 0 = F_2(s,\theta) + J_2(s,\theta) \begin{pmatrix} \Delta s \\ \Delta\theta \end{pmatrix}. \tag{6.9b}$$

We recall the Jacobian for DAE constrained parameter estimation problems and a single experiment

$$J = \begin{pmatrix} J_1 \\ J_2 \end{pmatrix} = \left(\begin{array}{cccccc} D_0^1 & D_1^1 & \cdots & D_{n_{ms}}^1 & D_{s^r}^1 & D_\theta^1 \\ \hline G_0 & \begin{pmatrix} -I_{n_y} & 0 \end{pmatrix} & & & G_0^{s^r} & G_0^\theta \\ & \ddots & \ddots & & \vdots & \vdots \\ & & G_{n_{ms}-1} & \begin{pmatrix} -I_{n_y} & 0 \end{pmatrix} & G_{n_{ms}-1}^{s^r} & G_{n_{ms}-1}^\theta \\ H_0 & & & & H_0^{s^r} & H_0^\theta \\ & H_1 & & & H_1^{s^r} & H_1^\theta \\ & & \ddots & & \vdots & \vdots \\ & & & H_{n_{ms}} & H_{n_{ms}}^{s^r} & G_0^{s^r} \\ D_0^2 & D_1^2 & \cdots & D_{n_{ms}}^2 & D_{s^r}^2 & D_\theta^2 \end{array}\right), \tag{6.10}$$

see (4.27), to illustrate the required computational effort for a single iteration of the generalized Gauss–Newton method. Here, we have added a new column for the derivatives of (6.8) with respect to s^r.

In each iteration, we have to evaluate $n_y + n_z + n_\theta + n_r$ derivatives of the underlying model equations (6.4) per shooting interval to compute the matrices G_k, $G_k^{s^r}$, G_k^θ, $k = 0, \ldots, n_{ms} - 1$ and H_k, $H_k^{s^r}$, H_k^θ, $k = 0, \ldots, n_{ms}$. For complex systems of partial differential equations, the effort is tremendous.

As discussed in Section 4.3.4, the condensed system has the form

$$\min_{\Delta s_0^y, \Delta\theta, \Delta s^r} \quad \frac{1}{2} \left\| u_1 + E_1 \Delta s_0^y + E_1^\theta \Delta\theta + E_1^r \Delta s^r \right\|_2^2 \tag{6.11a}$$

$$\text{s.t.} \quad 0 = u_2 + E_2 \Delta s_0^y + E_2^\theta \Delta\theta + E_2^r \Delta s^r. \tag{6.11b}$$

cf. (4.30). Note, that we have to consider an additional block for the variables s^r.

6.1.1 Derivation of the reduced approach

Schlöder [85] developed an approach for high-dimensional ODE systems that couples the computation of the Jacobian (6.10) and the subsequent condensing algorithm presented in Section 4.3.4. The approach uses directional derivatives which exploit the special structure of the parametrized parameter estimation problem. With the approach and the extensions for DAE constrained parameter estimation problems developed by Bauer [7], it is possible to reduce the computational effort to evaluate and solve (6.7) to the one of single shooting.

We develop a different approach than in Bauer [7] that leads to an other formulation of the reduced condensed system and increase the set of problems that is solvable by this method to described by (6.1).

Before we continue, we reformulate (if necessary we rearrange the time points) the multi-experiment interior point constraints (6.1e)

$$0 = \sum_{i=1}^{n_{\text{ex}}} \sum_{k=0}^{n_{\text{ms}}^i} \sum_{j \in \mathscr{I}(k)} r_k^i(\psi^i(t_j^i), \theta, s^r), \tag{6.12}$$

and the residuals of the measurements

$$0 = \sum_{i=1}^{n_{\text{ex}}} \sum_{k=0}^{n_{\text{ms}}^i} \sum_{j \in \mathscr{J}(k)} \hat{h}_k^i(\psi^i(t_j^i), \theta), \quad \hat{h}_k^i = \eta_k^i - h_k^i, \tag{6.13}$$

with

$$\mathscr{I}(k) = \left\{ j \mid \tau_k^i \leqslant t_j < \tau_{k+1}^i \right\},\ k = 0, \ldots, n_{\text{ms}}^i,\ i = 1, \ldots, n_{\text{ex}},$$
$$\mathscr{J}(k) = \left\{ j \mid \tau_k^i \leqslant t_j < \tau_{k+1}^i \right\},\ k = 0, \ldots, n_{\text{ms}}^i,\ i = 1, \ldots, n_{\text{ex}}.$$

We refer to Equations (6.12) and (6.13) as separability conditions.

We present the following computations exemplarily for the first experiment. For the remaining $n_{\text{ex}} - 1$ experiments the steps have to be repeated. For the sake of readability, we omit the superscript 1 and consider a single experiment formulation of the initial conditions in Equation (6.6) of $d(s, \theta)$.

We have already eliminated the continuity constraints and the consistency conditions. Because of the definition of $d(s, \theta)$ in Equation (6.6) we deduce that E_2 has $n_y + n_r$ rows. We separate Equation (6.11b) and obtain

$$0 = u_{21} + E_{21} \Delta s_0^y + E_{21}^\theta \Delta\theta + E_{21}^s \Delta s^r, \tag{6.14a}$$
$$0 = u_{22} + E_{22} \Delta s_0^y + E_{22}^\theta \Delta\theta + E_{22}^s \Delta s^r, \tag{6.14b}$$

where (6.14a) contains the first n_y rows of (6.11b) and (6.14b) includes the remaining n_r rows. From Section 4.3.4 and Equation (6.6), it follows directly that the matrix E_{21} is regular.

We eliminate the variables Δs_0^y, which are defined by the initial conditions (6.1d), and obtain the reduced condensed system for the remaining variables $(\Delta\theta,\ \Delta s^r)$

$$\min_{\Delta\theta, \Delta s^r} \quad \frac{1}{2} \left\| \tilde{u}_1 + \tilde{E}_1^\theta \Delta\theta + \tilde{E}_1^s \Delta s^r \right\|_2^2 \tag{6.15a}$$
$$\text{s.t.} \quad 0 = \tilde{u}_2 + \tilde{E}_2^\theta \Delta\theta + \tilde{E}_2^s \Delta s^r. \tag{6.15b}$$

Note, that Equation (6.15b) has only n_r rows. We define

$$M_\theta^0 := - \begin{pmatrix} E_{21} & 0 \\ H_0^y & H_0^z \end{pmatrix}^{-1} E_{21}^\theta, \tag{6.16a}$$

$$M_s^0 := -\begin{pmatrix} E_{21} & 0 \\ H_0^y & H_0^z \end{pmatrix}^{-1} E_{21}^s, \tag{6.16b}$$

$$M_r^0 := -\begin{pmatrix} E_{21} & 0 \\ H_0^y & H_0^z \end{pmatrix}^{-1} u_{21}. \tag{6.16c}$$

We refer to $M_\theta^\cdot$, $M_s^\cdot$ and $M_r^\cdot$ as *seed* matrices.

Remark Because of the index-1-assumption, the inverse matrix in Equations (6.16) is well defined.

With the definitions of the separability conditions (6.12) and (6.13), we have

$$D_\theta^2 = \frac{\mathrm{d}}{\mathrm{d}\theta} \sum_{i=1}^{n_{\mathrm{ex}}} \sum_{k=0}^{n_{\mathrm{ms}}^i} \sum_{j \in \mathscr{I}(k)} r_k^i(\psi^i(t_j^i), \theta) = \sum_{i=1}^{n_{\mathrm{ex}}} \sum_{k=0}^{n_{\mathrm{ms}}^i} \frac{\mathrm{d}}{\mathrm{d}\theta} \sum_{j \in \mathscr{I}(k)} r_k^i(\psi^i(t_j^i), \theta) = \sum_{i=1}^{n_{\mathrm{ex}}} \sum_{k=0}^{n_{\mathrm{ms}}^i} D_{\theta,k}^{2,i},$$

$$D_\theta^1 = \frac{\mathrm{d}}{\mathrm{d}\theta} \sum_{i=1}^{n_{\mathrm{ex}}} \sum_{k=0}^{n_{\mathrm{ms}}^i} \sum_{j \in \mathscr{J}(k)} \hat{h}_k^i(\psi^i(t_j^i), \theta) = \sum_{i=1}^{n_{\mathrm{ex}}} \sum_{k=0}^{n_{\mathrm{ms}}^i} \frac{\mathrm{d}}{\mathrm{d}\theta} \sum_{j \in \mathscr{J}(k)} \hat{h}_k^i(\psi^i(t_j^i), \theta) = \sum_{i=1}^{n_{\mathrm{ex}}} \sum_{k=0}^{n_{\mathrm{ms}}^i} D_{\theta,k}^{1,i}.$$

We get similar expressions for the derivatives with respect to s^r. With these relations, we obtain a forward recursion that is closely related to the backward condensing (4.31) in Section 4.3.4

$$\begin{aligned}
\tilde{E}_l^\theta =& D_0^l \cdot M_\theta^0 + \left(\sum_{k=1}^{n_{\mathrm{ms}}} D_k^l \prod_{j=1}^{k} \begin{pmatrix} G_{k-j} \\ H_{k-j+1} \end{pmatrix} \right) \cdot M_\theta^0 \\
&+ D_\theta^l + \sum_{k=1}^{n_{\mathrm{ms}}} D_k^l \left[\sum_{j=0}^{k-1} \left(\prod_{l=1}^{k-j-1} \begin{pmatrix} G_{k-l} \\ H_{k-l+1} \end{pmatrix} \begin{pmatrix} G_j^\theta \\ H_{j+1}^\theta \end{pmatrix} \right) \right] \\
=& D_0^l \cdot M_\theta^0 + \sum_{k=1}^{n_{\mathrm{ms}}} \left\{ D_k^l \left[\prod_{j=1}^{k} \begin{pmatrix} G_{k-j} \\ H_{k-j+1} \end{pmatrix} \cdot M_\theta^0 \right.\right. \\
&\left.\left. + \sum_{j=0}^{k-1} \left(\prod_{l=1}^{k-j-1} \begin{pmatrix} G_{k-l} \\ H_{k-l+1} \end{pmatrix} \begin{pmatrix} G_j^\theta \\ H_{j+1}^\theta \end{pmatrix} \right) \right] + D_{\theta,k}^l \right\}, \quad l = 1,2,
\end{aligned} \tag{6.17a}$$

$$\begin{aligned}
\tilde{E}_l^s =& D_0^l \cdot M_s^0 + \sum_{k=1}^{n_{\mathrm{ms}}} \left\{ D_k^l \left[\prod_{j=1}^{k} \begin{pmatrix} G_{k-j} \\ H_{k-j+1} \end{pmatrix} \cdot M_s^0 \right.\right. \\
&\left.\left. + \sum_{j=0}^{k-1} \left(\prod_{l=1}^{k-j-1} \begin{pmatrix} G_{k-l} \\ H_{k-l+1} \end{pmatrix} \begin{pmatrix} G_j^{s^r} \\ H_{j+1}^{s^r} \end{pmatrix} \right) \right] + D_{s^r,k}^l \right\}, \quad l = 1,2,
\end{aligned} \tag{6.17b}$$

$$\begin{aligned}\tilde{u}_l =& D_0^l \cdot M_r^0 + \sum_{k=1}^{n_{\text{ms}}} \left\{ D_k^l \left[\prod_{j=1}^{k} \begin{pmatrix} G_{k-j} \\ H_{k-j+1} \end{pmatrix} \cdot M_r^0 \right.\right. \\ &\left.\left. + \sum_{j=0}^{k-1} \left(\prod_{l=1}^{k-j-1} \begin{pmatrix} G_{k-l} \\ H_{k-l+1} \end{pmatrix} \begin{pmatrix} c(\tau_{j+1}) \\ g(\tau_{j+1}) \end{pmatrix} \right) \right] + R_k^l \right\}, \quad l = 1,2,\end{aligned} \tag{6.17c}$$

where we suppress the remaining dependencies of the continuity conditions c and the consistency conditions g and define

$$R_k^l := \begin{cases} \sum\limits_{j \in \mathcal{J}(k)} \hat{h}_k(\psi(t_j), \theta), & \text{if } l = 1, \\ \sum\limits_{j \in \mathcal{I}(k)} r_k(\psi(t_j), \theta), & \text{else.} \end{cases} \tag{6.18}$$

We update the seed matrices recursively

$$M_\theta^k := \begin{pmatrix} G_k \\ H_{k+1} \end{pmatrix} \cdot M_\theta^{k-1} + \begin{pmatrix} G_k^\theta \\ H_{k+1}^\theta \end{pmatrix}, \tag{6.19a}$$

$$M_s^k := \begin{pmatrix} G_k \\ H_{k+1} \end{pmatrix} \cdot M_s^{k-1} + \begin{pmatrix} G_k^{s^r} \\ H_{k+1}^{s^r} \end{pmatrix}, \quad k = 1, \ldots, n_{\text{ms}}, \tag{6.19b}$$

$$M_r^k := \begin{pmatrix} G_k \\ H_{k+1} \end{pmatrix} \cdot M_r^{k-1} + \begin{pmatrix} c(\tau_{k+1}) \\ g(\tau_{k+1}) \end{pmatrix}, \tag{6.19c}$$

and set

$$\hat{E}_l^{\theta,0} := D_0^l \cdot M_\theta^0 + D_{\theta,0}^l, \tag{6.20a}$$

$$\hat{E}_l^{s,0} := D_0^l \cdot M_s^0 + D_{s^r,0}^l, \quad l = 1,2, \tag{6.20b}$$

$$\hat{u}_l^0 := D_0^l \cdot M_r^0 + R_0^l, \tag{6.20c}$$

where we used the definition given by Equation (6.18) and compute the expressions $\tilde{u}_l$, $\tilde{E}_l^\theta$, $\tilde{E}_l^s$, $l = 1,2$ by the forward recursion

$$\hat{E}_l^{\theta,k} = \hat{E}_l^{\theta,k-1} + D_k^l \cdot M_\theta^k + D_{\theta,k}^l, \tag{6.21a}$$

$$\hat{E}_l^{s,k} = \hat{E}_l^{s,k-1} + D_k^l \cdot M_s^k + D_{s^r,k}^l, \quad k = 1, \ldots, n_{\text{ms}} \quad l = 1,2, \tag{6.21b}$$

$$\hat{u}_l^k = \hat{u}_l^{k-1} + D_k^l \cdot M_r^k + R_k^l, \tag{6.21c}$$

With

$$\tilde{E}_l^\theta = \hat{E}_l^{\theta,n_{\text{ms}}}, \tilde{E}_l^s = \hat{E}_l^{s,n_{\text{ms}}}, \tilde{u}_l = \hat{u}_i^{n_{\text{ms}}}, \quad l = 1,2$$

we obtain the reduced condensed system (6.15).

In difference to the approach presented by Bauer [7] we eliminate the consistency conditions completely from the constraints of the reduced condensed system, cf (6.20) and (6.21). We transfer all informations of the consistency conditions to the seed matrices, compare Equations (6.19). The approach illustrated by Bauer [7] results into redundant constraints, i.e., the resulting reduced condensed system has more rows than the regular condensed system, which could lead to numerical problems.

The expressions in Equations (6.20) are evaluated by directional derivatives. According to this, we compute $D_k^l \cdot M_\theta^k + D_{\theta,k}^l$, $D_k^l \cdot M_s^k + D_{s^r,k}^l$ and $D_k^l \cdot M_r^k$ in (6.21) by directional derivatives, too.

The main advantage of this method is that we do not have to compute the block matrices D, G and H of the definition of the Jacobian (6.10) explicitly. Thus, we save $n_y + n_z$ evaluations of the variational differential equations.

6.1.2 Determination of the increment

Again, we illustrate the proceeding for the first experiment and omit the superscript 1.

Here, we explain in detail the derivation of the reduced condensed system and the determination of the increment for the generalized Gauss–Newton method. Therefore, we define the residual between initial values and shooting variables at $t = \tau_0$

$$d_0(s,\theta) := y_0(\theta, s^r) - s_0^y. \tag{6.22}$$

Equation (6.22) is a constraint of the parameter estimation problem, see (6.6). For the generalized Gauss–Newton method we evaluate the derivatives of d_0 and the algebraic constraints (6.4b) at $t = \tau_0$ with respect to s_0^y, s_0^z, s^r and θ

$$\begin{aligned} D_0^v &:= \frac{d\, d_0(s,\theta)}{d\, v}, \\ H_0^v &:= \frac{d\, g(\tau_0, s_0, \theta)}{d\, v}, \end{aligned} \qquad v = s_0^y,\ s_0^z,\ s^r,\ \theta$$

and set up the linearization of (6.22) and the consistency conditions at $t = \tau_0$

$$d_0(s,\theta) + D_0^{s_0^y} \Delta s_0^y + D_0^{s_0^z} \Delta s_0^z + D_0^{s^r} \Delta s^r + D_0^\theta \Delta\theta = 0 \tag{6.23a}$$

$$g(\tau_0, s_0, \theta) + H_0^{s_0^y} \Delta s_0^y + H_0^{s_0^z} \Delta s_0^z + H_0^{s^r} \Delta s^r + H_0^\theta \Delta\theta = 0. \tag{6.23b}$$

It is easy to see that Equations (6.23) are a subsystem of (6.9b) since (6.22) is a subequation of (6.6). As mentioned before, we require that the algebraic equations have index 1. It follows that $H_0^{s_0^z}$ is regular, see Section 4.1.1. From Equation (6.22), we deduce

$$D_0^{s_0^y} = -I_{n_y}, \quad D_0^{s_0^z} = 0.$$

We obtain the matrix

$$M := \begin{pmatrix} D_0^{s_0^y} & D_0^{s_0^z} \\ H_0^{s_0^y} & H_0^{s_0^z} \end{pmatrix} = \begin{pmatrix} -I_{n_y} & 0 \\ H_0^{s_0^y} & H_0^{s_0^z} \end{pmatrix}$$

which is regular and solve (6.23) for $s_0^y,\ s_0^z$

$$\begin{aligned}\begin{pmatrix}\Delta s_0^y\\ \Delta s_0^z\end{pmatrix} &= -\begin{pmatrix}-I_{n_y} & 0\\ H_0^{s_0^y} & H_0^{s_0^z}\end{pmatrix}^{-1}\left(\begin{pmatrix}D_0^{s^r}\\ H_0^{s^r}\end{pmatrix}\Delta s^r + \begin{pmatrix}D_0^{\theta}\\ H_0^{\theta}\end{pmatrix}\Delta\theta + \begin{pmatrix}d_0\\ g_0\end{pmatrix}\right)\\ &= M_s^0\Delta s^r + M_\theta^0\Delta\theta + M_r^0\end{aligned}\tag{6.24}$$

with the definitions

$$M_r^0 := -\begin{pmatrix}-I_{n_y} & 0\\ H_0^{s_0^y} & H_0^{s_0^z}\end{pmatrix}^{-1}\begin{pmatrix}d_0\\ g_0\end{pmatrix},\tag{6.25a}$$

$$M_s^0 := -\begin{pmatrix}-I_{n_y} & 0\\ H_0^{s_0^y} & H_0^{s_0^z}\end{pmatrix}^{-1}\begin{pmatrix}D_0^{s^r}\\ H_0^{s^r}\end{pmatrix},\tag{6.25b}$$

$$M_\theta^0 := -\begin{pmatrix}-I_{n_y} & 0\\ H_0^{s_0^y} & H_0^{s_0^z}\end{pmatrix}^{-1}\begin{pmatrix}G_0^{\theta}\\ H_0^{p}\end{pmatrix},\tag{6.25c}$$

cf. Equations (6.16). The columns of $(M_s^0,\ M_\theta^0)$ form a basis of the null space of the matrix that defines the linear system of equations (6.23). We use (6.25) as initial seed matrices for the directional derivatives.

Let $t = \tau_0$ be not only a shooting node but also a measurement point, where we have to evaluate the model response $h_0(\tau_0,\ s_0,\ \theta)$. Then,

$$\frac{\partial h_0}{\partial(s_0^y, s_0^z)}(\tau_0, s_0, \theta)\begin{pmatrix}\Delta s_0^y\\ \Delta s_0^z\end{pmatrix} + \frac{\partial h_0}{\partial s^r}(\tau_0, s_0, \theta)\Delta s^r + \frac{\partial h_0}{\partial\theta}(\tau_0, s_0, \theta)\Delta\theta + h_0(\tau_0, s_0, \theta)$$

is a row of J^1 (6.9b).

Observe, h_0 does not depend on $s_i,\ i = 1, \dots, n_{ms}$. We insert (6.24) and obtain

$$\begin{aligned}&\frac{\partial h_0}{\partial s_0}(\tau_0, s_0, \theta)\cdot\left(M_s^0\Delta s^r + M_\theta^0\Delta\theta + M_r^0\right)\\ &+\frac{\partial h_0}{\partial s^r}(\tau_0, s_0, \theta)\Delta s^r + \frac{\partial h_0}{\partial\theta}(\tau_0, s_0, \theta)\Delta\theta + h_0(\tau_0, s_0, \theta).\end{aligned}\tag{6.26}$$

We factor out the increments Δs^r and $\Delta\theta$ in Equation (6.26) to get

$$\begin{aligned}&\left[\frac{\partial h_0}{\partial s_0}(\tau_0, s_0, \theta)\cdot M_s^0 + \frac{\partial h_0}{\partial s^r}(\tau_0, s_0, \theta)\right]\Delta s^r\\ +&\left[\frac{\partial h_0}{\partial s_0}(\tau_0, s_0, \theta)\cdot M_\theta^0 + \frac{\partial h_0}{\partial\theta}(\tau_0, s_0, \theta)\right]\Delta\theta\\ +&\left[\frac{\partial h_0}{\partial s_0}(\tau_0, s_0, \theta)\cdot M_r^0 + h_0(\tau_0, s_0, \theta)\right].\end{aligned}\tag{6.27}$$

Remark Here and in the following, "$\cdot$", e.g., in Equation (6.27), does not denote a matrix matrix product but a directional derivative, cf. 6.1.1. The first would require the evaluation of $n_y + n_z$ additional variational differential equations while for the second $n_\theta + n_r$ VDEs have to be evaluated, which is necessary anyway.

Following the remark, we compute

$$\left(\frac{\partial h_0}{\partial s_0}(\tau_0, s_0, \theta) \cdot M_s^0\right)_j = \frac{\mathrm{d}}{\mathrm{d}\varepsilon} h_0(\tau_0, s_0 + \varepsilon \cdot \left(M_s^0\right)_j, \theta), \tag{6.28}$$

where $B_{\cdot j}$ denotes the $j-$th column of some matrix B. As already mentioned in the chapters before, we use algorithmic differentiation to efficiently compute the required directional derivatives of the model response in (6.27). If at $t = \tau_0$ multi-experiment interior point constraints have to be evaluated, we proceed in the same manner with the derivatives of (6.1e)

$$\begin{aligned} 0 = \quad & \left[\frac{\partial r}{\partial s_0}(\tau_0, s_0, \theta, s^r) \cdot M_s^0 + \frac{\partial s}{\partial s^r}(\tau_0, s_0, \theta, s^r)\right] \Delta s^r \\ & + \left[\frac{\partial r}{\partial s_0}(\tau_0, s_0, \theta, s^r) \cdot M_\theta^0 + \frac{\partial r}{\partial \theta}(\tau_0, s_0, \theta, s^r)\right] \Delta\theta \\ & + \left[\frac{\partial r}{\partial s_0}(\tau_0, s_0, \theta, s^r) \cdot M_r^0 + r_0(\tau_0, s_0, \theta, s^r)\right]. \end{aligned} \tag{6.29}$$

With the evaluated derivatives, we define the right-hand side vectors and the matrices which we will use to assemble the reduced condensed system (6.11)

$$\begin{aligned} u_1^0 &:= \left(\frac{\partial}{\partial s_0} h_0(\tau_0, s_0, \theta) \cdot M_r^0 + h_0(\tau_0, s_0, \theta)\right), \\ u_2^0 &:= \left(\frac{\partial}{\partial s_0} r_0(\tau_0, s_0, \theta, s^r) \cdot M_r^0 + r_0(\tau_0, s_0, \theta, s^r)\right), \end{aligned} \tag{6.30a}$$

$$\begin{aligned} E_1^{s,0} &:= \left(\frac{\partial}{\partial s_0} h_0(\tau_0, s_0, \theta) \cdot M_s^0\right), \\ E_2^{s,0} &:= \left(\frac{\partial}{\partial s_0} r_0(\tau_0, s_0, \theta, s^r) \cdot M_r^0 + \frac{\partial}{\partial s^r} r_0(\tau_0, s_0, \theta, s^r)\right), \end{aligned} \tag{6.30b}$$

$$\begin{aligned} E_1^{\theta,0} &:= \left(\frac{\partial}{\partial s_0} h_0(\tau_0, s_0, \theta) \cdot M_\theta^0 \ + \frac{\partial}{\partial \theta} h_0(\tau_0, s_0, \theta)\right), \\ E_2^{\theta,0} &:= \left(\frac{\partial}{\partial s_0} r_0(\tau_0, s_0, \theta, s^r) \cdot M_\theta^0 + \frac{\partial}{\partial \theta} r_0(\tau_0, s_0, \theta, s^r)\right). \end{aligned} \tag{6.30c}$$

Here, we used that

$$\frac{\partial}{\partial s^r} h_0 = 0.$$

Now, we explain how we proceed from a shooting node τ_k to a stopping point t_s. A stopping point is a point where we have to stop the computation of the nominal trajectory to evaluate either a measurement function h, the algebraic equations g or the multi-experiment interior point constraints r.

At $t = \tau_k$, the values of the shooting variables s_k, the parameters θ, the previously computed seed matrices M_r^k, M_s^k, M_θ^k and the matrices for the reduced condensed system u_l^k, $E_l^{s,k}$, $E_l^{\theta,k}$, $l = 1,2$ are given. We evaluate the constraints, compute the corresponding directional derivatives and add them to the known matrices for the condensed system

$$\begin{aligned}
\hat{u}_1^j &= u_1^k + \left(\frac{\partial}{\partial s} h_k(\tau_k, s_k, \theta) \cdot M_r^k + h_k(\tau_k, s_k, \theta)\right), \\
\hat{u}_2^j &= u_2^k + \left(\frac{\partial}{\partial s} r_k(\tau_k, s_k, \theta, s^r) \cdot M_r^k + r_k(\tau_k, s_k, \theta, s^r)\right),
\end{aligned} \tag{6.31a}$$

$$\begin{aligned}
\hat{E}_1^{s,j} &= E_1^{s,k} + \left(\frac{\partial}{\partial s} h_k(\tau_k, s_k, \theta) \cdot M_s^k\right), \\
\hat{E}_2^{s,j} &= E_2^{s,k} + \left(\frac{\partial}{\partial s} r_k(\tau_k, s_k, \theta, s^r) \cdot M_s^k + \frac{\partial}{\partial s^r} r_k(\tau_k, s_k, \theta, s^r)\right),
\end{aligned} \tag{6.31b}$$

$$\begin{aligned}
\hat{E}_1^{\theta,j} &= E_1^{\theta,k} + \left(\frac{\partial}{\partial s} h_k(\tau_k, s_k, \theta) \cdot M_\theta^k + \frac{\partial}{\partial \theta} h_k(\tau_k, s_k, \theta)\right), \\
\hat{E}_2^{\theta,j} &= E_2^{\theta,k} + \left(\frac{\partial}{\partial s} r_k(\tau_k, s_k, \theta, s^r) \cdot M_\theta^k + \frac{\partial}{\partial \theta} r_k(\tau_k, s_k, \theta, s^r)\right),
\end{aligned} \tag{6.31c}$$

which follows directly from Equations (6.17).

Afterwards, we solve the model equations (6.4) on the interval $[\tau_k, t_s]$. As before, we denote the resulting nominal trajectories by

$$\psi(t; s_k, \theta).$$

The seed matrices are updated by the simultaneous evaluation of the variational differential equations via methods of internal numerical differentiation

$$\hat{M}_r^j = \frac{\partial}{\partial s} \psi(t_s; s_k, \theta)) \cdot M_r^k \tag{6.32a}$$

$$\hat{M}_s^j = \frac{\partial}{\partial s} \psi(t_s; s_k, \theta) \cdot M_s^k, \tag{6.32b}$$

$$\hat{M}_\theta^j = \frac{\partial}{\partial s} \psi(t_s; s_k, \theta) \cdot M_\theta^k, \tag{6.32c}$$

where we use M_r^k, M_s^k and M_θ^k as initial states and directions for the sensitivities of the states at the same time. The updated seed matrices are applied to proceed to the next stopping point and the previous steps are repeated.

If at $t = t_s$ the next shooting node is reached, we evaluate the continuity conditions (6.2) and the consistency conditions (6.3). Then, the seed matrices at $t = \tau_{k+1}$ result to

$$M_r^{k+1} = \hat{M}_r^j + \begin{pmatrix} c(\tau_{k+1}) \\ g(\tau_{k+1}) \end{pmatrix}, \tag{6.33a}$$

$$M_s^{k+1} = \hat{M}_s^j, \tag{6.33b}$$

$$M_\theta^{k+1} = \hat{M}_\theta^j. \tag{6.33c}$$

We continue to execute these steps above until we reach the last shooting node $\tau_{n_{\mathrm{ms}}+1} = t_{end}$ to obtain the reduced condensed system

$$\min_{\Delta s^r, \Delta\theta} \quad \frac{1}{2} \left\| \tilde{u}_1 + \tilde{E}_1^s \Delta s^r + \tilde{E}_1^\theta \Delta\theta \right\|_2^2 \tag{6.34a}$$

$$\text{s.t.} \quad 0 = \tilde{u}_2 + \tilde{E}_2^s \Delta s^r + \tilde{E}_2^\theta \Delta\theta, \tag{6.34b}$$

where we use the settings

$$\tilde{u}_l := u_l^{n_{ms}+1},\ \tilde{E}_l^s := E_l^{s,n_{ms}+1},\ \tilde{E}_l^\theta := E_l^{\theta,n_{ms}+1},\ l = 1,2.$$

The effort to set up the reduced condensed system (6.34) consists of the integration of the underlying model equations (6.4) and the computation of $n_r + n_\theta + 1$ directional derivatives by internal numerical differentiation. If the initial values of the differential states are given in the form

$$y(t_0) = y_0(\theta),$$

i.e., the initial conditions do not depend explicitly on s^r, the number of directional derivatives is independent of the number of states.

After we have applied this approach to all experiments, the reduced condensed system for n_{ex} experiments has the following form:

$$J^c = \begin{pmatrix} \tilde{E}_1^{s,1} & & & \tilde{E}_1^{\theta,1} \\ & \ddots & & \vdots \\ & & \tilde{E}_1^{s,n_{\mathrm{ex}}} & \tilde{E}_1^{\theta,n_{\mathrm{ex}}} \\ \tilde{E}_2^{s,1} & \dots & \tilde{E}_2^{s,n_{\mathrm{ex}}} & \sum_{i=1}^{n_{\mathrm{ex}}} \tilde{E}_2^{\theta,i} \end{pmatrix}, \quad F^c = \begin{pmatrix} \tilde{u}_1^1 \\ \vdots \\ \tilde{u}_1^{n_{\mathrm{ex}}} \\ \sum_{i=1}^{n_{\mathrm{ex}}} \tilde{u}_2^i \end{pmatrix}. \tag{6.35}$$

We solve the system, which is defined by (6.35), for Δs^r and $\Delta\theta$ and compute the remaining increments for the shooting variables by

$$\Delta s_k^i = M_r^{k,i} + M_s^{k,i} \Delta s_r^i + M_\theta^{k,i} \Delta\theta, \quad \begin{aligned} k &= 0, \dots, n_{ms}^j, \\ i &= 1, \dots, n_{ex}, \end{aligned} \tag{6.36}$$

where we use the previously computed seed matrices.

6.2 *PAREMERA*: PARameter Estimation for Multiple Experiments using a Reduced Approach

In Chapter 4, mathematical methods for solving parameter estimation problems have been established by giving convergence results of the damped generalized Gauss–Newton method. In section 6.1, we presented a reduced approach for the efficient treatment of these kind of problems. These results are the basis for the implementation of the PARameter Estimation software for Multiple Experiments using a Reduced Approach (*PAREMERA*), a new software package for solving PDE constrained parameter estimation problems with multi-experiment interior point constraints.

Survey of existing solvers

The variety of solvers that are available to solve nonlinear constrained parameter estimation problems is large. It is not our concern to give a complete overview of the available implementations for this kind of problems.

Naturally, any optimization algorithm for constrained problems could be applied to treat parameter estimation problems. However, it is possible that such solvers converge to unwanted local minima or are highly inefficient, see Bock [13]. Therefore, it is strongly advised to use appropriate solvers when faced with parameter estimation problems.

The following overview gives an insight to the problem classes which can be solved by the specific software as well as the applied numerical software.

- *PARFIT* by Bock et al. [13] can solve parameter estimation problems with ODE constraints with multiple shooting by a generalized Gauss–Newton method and is based mainly on the methods presented in Chapter 4 that can also be found in Bock et al. [11, 12, 16]. There is also an extension to DAE constrained problems presented in Körkel [58] and Bock et al. [14]. The implementation language is FORTRAN77.

- *FIXFIT* by Schlöder [85] is a generalized Gauss–Newton method for parameter estimation problems constrained by high-dimensional ODE systems. It adapts the reduced approach for this class of problems. The solver is implemented in FORTRAN77. According to Schlöder [86], no implemented version is available anymore.

- *PEST* by Doherty et al. [35] is a commercial, model independent parameter estimation software, i.e. the user has to provide the model equation in a specific form. The code relies on a Levenberg–Marquardt algorithm, see Levenberg [63] and Marquardt [70] with a Broyden update. The derivatives of the model functions and constraints are computed by finite differences. The software package is implemented in C++.

- The commercial simulation and parameter estimation tools *Presto-Kinetics* and *Predici* for reactor kinetics and macromolecular processes, respectively, by Wulkow [102]. The evaluation of the derivatives is done by finite differences following the principles of IND, see Bock [11]. Simulated annealing is used for the global search, see Kirkpatrick et al. [56]. The local search is done by a damped Gauss–Newton method following Deuflhard [32]. Furthermore, an analysis of the correlation of the parameters is used within the method of reduced directions, see Telgmann [92].

- *EASY-FIT* by Schittkowski [84] is suited for parameter estimation problems with inequality constraints. It can handle DAE constraints up to index 3 or one-dimensional PDAE constraints. It is also suited for the treatment of switching functions. The derivatives are computed by algorithmic differentiation. SQP methods are applied to solve the least-squares problem. The implementation is in Fortran.

- *COPASI* by Hoops et al. [52] is a parameter estimation software for mainly biochemical processes with a high number of parameters. It is equipped with a broad variety of optimization methods such as steepest descent, Levenberg–Marquardt and some evolutionary algorithms. The implementation language is C++.

- *ParamEDE* by Lenz [62] is a software package for parameter estimation problems constrained by delay differential equations (DDE). It is the first that applies a damped generalized Gauss–Newton method based on the restrictive monotonicity test and multiple shooting for parameter estimation with DDEs.

Features of *PAREMERA*

PAREMERA is implemented in Fortran90 and fits into the field of present solvers in the following ways.

Only some of the solvers listed above, e.g., *PARFIT*, *ParamEDE* and *Presto*, use the generalized Gauss–Newton method to compute the solution of the parameter estimation problem. Bock [13] showed that the generalized Gauss–Newton method has the advantage not to converge toward a stationary point with $\kappa > 1$. Those points are often saddle points or local minima with a large residual that are far away of the true parameters and are not stable against small perturbations of the measurement errors.

Another key feature of *PAREMERA* is the multiple shooting method to decompose the time interval and, thus, reduces the nonlinearities of stiff systems. This feature is supported else by, e.g., *PARFIT*, *FIXFIT* and *ParamEDE*.

FIXFIT was the only other solver based on multiple shooting that uses the reduced approach that allows to solve parameter estimation problems with PDE or high-dimensional ODE constraints in an acceptable time frame.

In contrast to, e.g., *Predici* and *Presto-Kinetics*, *PAREMERA* is particularly outlined to be used in combination with a software tool for algorithmic differentiation which provides, up to machine precision, the exact derivatives of the model functions.

These features justify the implementation of a new solver.

6.2.1 Decomposition of the Jacobian

In Section 6.1, we illustrated the computation of the reduced condensed system (6.34). In this section, we present the structure exploiting algorithm for the decomposition of the Jacobian that we have implemented to solve the system defined by (6.35).

First, we predefine some notations. As mentioned in Section 6.1 we consider equality constraints of the form

$$0 = \sum_{i=1}^{n_{\text{ex}}} \sum_{k=1}^{m_r^i} r_k^i \left(\psi^i(t_k^i), \theta, s^{r,i}\right) \tag{6.37}$$

These constraints may still contain interior point constraints of the form

$$0 = \sum_{k=1}^{m_r^i} r_k^i \left(\psi^i(t_k^i), \theta, s^{r,i}\right), \quad i = 1, \dots, n_{\text{ex}}, \tag{6.38}$$

defined in Section 4.1. Since (6.38) are considered as constraints for a single experiment, we denote them with the subscript l for *local* while we refer to (6.37) by the subscript m for *multiple*

$$r_m = \sum_{i=1}^{n_{\text{ex}}} \sum_{k=1}^{m_r^i} r_k^i \left(\psi^i(t_k^i), \theta, s^{r,i}\right), \quad r_l^i = \sum_{k=1}^{m_r^i} r_k^i \left(\psi^i(t_k^i), \theta, s^{r,i}\right), \quad i = 1, \dots, n_{\text{ex}},$$

Observe, that we use the same variables $s^r = s^{r,1}, \dots, s^{r,\text{ex}}$ in both formulations. Because of (CQ), the matrix

$$\begin{pmatrix} \left(\frac{\partial r_l^i}{\partial s^r}\right)_{i=1,\dots,n_{\text{ex}}} \\ \frac{\partial r_m}{\partial s^r} \end{pmatrix}$$

is regular. If necessary, we rearrange $s^r = \left(s^{r_l T}, s^{r_m T}\right)^T$ such that

$$\frac{\partial r_l^i}{\partial s^{r_l}}, \quad i = 1, \dots, n_{\text{ex}}, \quad \frac{\partial r_m}{\partial s^{r_m}}$$

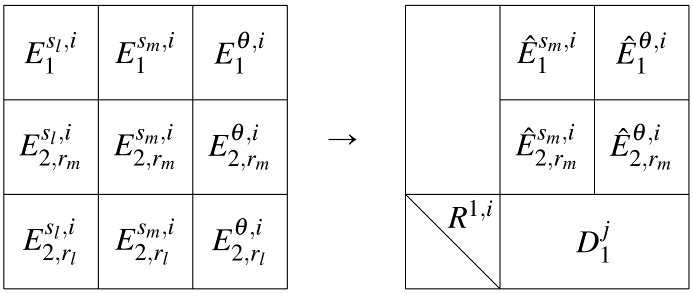
are regular matrices, too. Additionally, we rearrange the matrices (6.34b) equivalently to the following form

$$\begin{pmatrix} E_2^{s,i} & E_2^{\theta,i} \end{pmatrix} = \begin{pmatrix} E_{2,r_m}^{s_l,i} & E_{2,r_m}^{s_m,i} & E_{2,r_m}^{\theta,i} \\ E_{2,r_l}^{s_l,i} & E_{2,r_l}^{s_m,i} & E_{2,r_l}^{\theta,i} \end{pmatrix} \tag{6.39}$$

for $i = 1, \dots, n_{ex}$. Here, the superscripts s_l, s_m and θ on the right hand side of (6.39) identify the block matrices correspond to the increments Δs^{r_l}, Δs^{r_m} and $\Delta\theta$ respectively and the subscripts r_m and r_l denote if the block matrix was computed by directional derivative of the multi-experiment and experiment local interior point constraints, respectively.

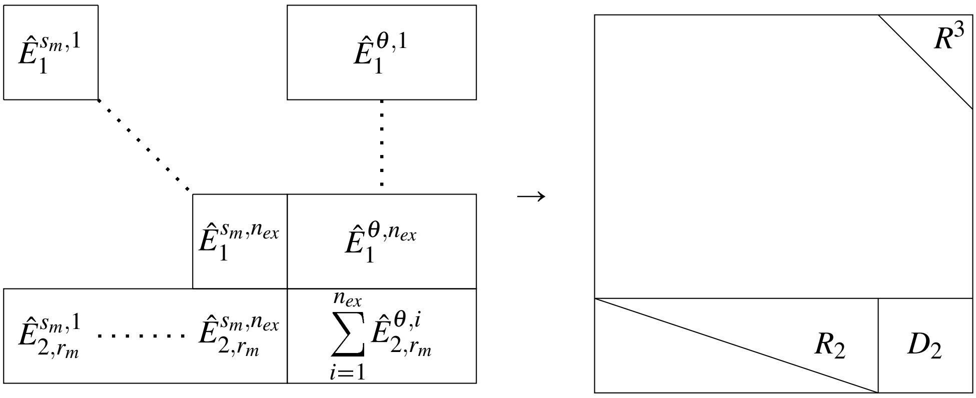
For the numerical treatment of the multi-experiment version of the condensed system (6.34), we transform the Jacobian to an upper triangular matrix. The Jacobian for n_{ex}

experiments is very sparse since there is a subvector $s^{r,i}$ for each experiment. That is why we store the blocks for each single experiment as presented on the left hand side of (6.40). By a series of equivalent matrix factorizations similar to an algorithm by Bock [13], the single blocks are transformed in the following fashion:



$i = 1,\ldots,n_{ex}.$ (6.40)

For simplicity, we suppress potential pivoting in the notation. Afterwards, we restore the remaining parts of the experiment block matrices in the upper right corner of (6.40) to a global, i.e for all n_{ex} experiments, system matrix (6.41).


$\hat{E}_1^{s_m,1}$
$\hat{E}_1^{\theta,1}$
$\hat{E}_1^{s_m,n_{ex}}$
$\hat{E}_1^{\theta,n_{ex}}$
$\hat{E}_{2,r_m}^{s_m,1}$ .... $\hat{E}_{2,r_m}^{s_m,n_{ex}}$
$\sum_{i=1}^{n_{ex}} \hat{E}_{2,r_m}^{\theta,i}$
$\rightarrow$
R^3
R_2
D_2

. (6.41)

The lower left block of the system matrix on the left hand side of Equation (6.41) that corresponds to $\Delta s^{r_m,i}$ is also transformed to an upper triangular matrix R_2 by matrix transformations, while the remaining block in the upper right that corresponds to $\Delta\theta$ is decomposed by an QR-decomposition with Householder transformations, see Golub and van Loan [45].

The right-hand sight of the system defined by (6.35) is adjusted according to the methods presented above. We obtain the increments $(\Delta\theta^T,\ \Delta s^{r^T})^T$ and compute the remaining increments by

$$\Delta s_k^i = M_r^{k,i} + M_s^{k,i}\Delta s^{r,i} + M_\theta^{k,i}\Delta\theta, \quad k = 0,\ldots,n_{ms}^i - 1, \tag{6.42}$$

$$i = 1,\ldots,n_{ex}. \tag{6.43}$$

6.2.2 Bound check

Usually, we do not allow inequality constraints for the parameter estimation problem. But it is possible that for specific values of the parameters the model can not be evaluated anymore, e.g., a negative reaction rate. It can happen that the generalized Gauss–Newton algorithm tends to such a point especially when the current iterate is not feasible. We have implemented an easy strategy to prevent this scenario from happening.

Let upper $\bar{\theta} \in I\!R^{n_\theta}$ and lower $\underline{\theta} \in I\!R^{n_\theta}$ bounds for the parameters be given

$$\theta \in \left(\underline{\theta}, \bar{\theta}\right). \tag{6.44}$$

Furthermore, let the new increment $\Delta\theta^k$ and the step length α^k have been computed, too. Then, we check if the updated iterate violates the constraints. If, this is the case we calculate the distance between the specific parameter and the bound and divide it by the increment. The result is the maximal step length. Since we do not want that

$$\theta_i^{k+1} = \bar{\theta}_i$$

or

$$\theta_i^{k+1} = \underline{\theta}_i$$

we multiply the maximal step length by a predefined factor, e.g., $\frac{1}{2}$. Thereby we achieve that the iteration remains feasible with respect to the Constraints (6.44) throughout the whole run of the optimization algorithm. The bound check algorithm then looks the following:

Algorithm 6.2.1. ***Bound check algorithm***

```
1.  Set $\alpha_1 = \alpha^k$.
2.  For $i = 1, \ldots, n_\theta$ do
3.      If $\theta_i^{k+1} < \underline{\theta}_i$ then
4.          Compute
                $$\alpha_2 = \left| \frac{\theta_i^{k+1} - \underline{\theta}_i}{\Delta\theta_i^k} \right|.$$
5.          Set    $\alpha_1 = \min(\alpha_1, \alpha_2)$.
6.      Else if $\theta_i^{k+1} > \bar{\theta}_i$ then
7.          Compute
                $$\alpha_2 = \left| \frac{\theta_i^{k+1} - \bar{\theta}_i}{\Delta\theta_i^k} \right|.$$
8.          Set    $\alpha_1 = \min(\alpha_1, \alpha_2)$.
9.      end if
10. end for
11. If $\alpha^k > \alpha_1$, set
                $$\alpha^k = \frac{1}{2}\alpha_1.$$
```

6.2.3 Parallelization

Usually, the multiple shooting formulation for initial values problems is easily parallelized because of its already separated structure. Unfortunately, this is not the case for the shooting formulation within the reduced approach. If we recall

$$M_r^{k+1} = \hat{M}_r^j + \begin{pmatrix} c(\tau_{k+1}) \\ g(t_{k+1}) \end{pmatrix}$$

from Equations (6.33), it is obvious why. We need $\hat{M}_r^j$ and $y(\tau_{k+1}; s_k, \theta)$ that are both computed in the previous shooting interval to initialize the seed matrix M_r^{k+1} at $t = \tau_{k+1}$. That is why we have to wait until the end of the previous shooting interval is reached, before we proceed with the next one.

The computation of the reduced condensed systems (6.34) for each of the n_{ex} experiments can be scheduled in parallel in a very natural way. We parallelize the loop over the n_{ex} experiments. Because of the structure how the Jacobian is saved this is a straightforward task. We use OpenMP [78] to organize the scheduling. If we have to consider multi-experiment interior point constraints, we compute the single summands in parallel and sum them up afterwards. As shown in Section 6.2.1, the vector s^r can be split up such that the single blocks affiliate with one specific experiment.

Additionally, it is possible to parallelize the matrix factorization in Equation (6.40) since the transformation of the blocks of the single experiments do not affect each other.

6.2.4 The main algorithm

This section is concerned with the main algorithm which is implemented in *PAREMERA* to solve parameter estimation problems by applying the reduced approach.

Algorithm 6.2.2. ***Main algorithm in PAREMERA***

1. *Set* $k := 0$. *Choose an initial guess* (s^0, θ^0), *an initial step length* α_0 *and predefine a termination tolerance TOL.*

2. *Compute* u_l^i, $E_l^{s,i}$ *and* $E_l^{\theta,i}$, $l = 1, 2,\ i = 1, \dots, n_{ex}$ *for all experiments as described in Section 6.1.*

3. *Solve*

$$\begin{aligned} \min_{\Delta s^k, \Delta\theta} \quad & \frac{1}{2} \left\| J_1^c \begin{pmatrix} \Delta s^k \\ \Delta\theta^k \end{pmatrix} + F_1^c \right\|_2^2 \\ s.t. \quad 0 = & J_2^c \begin{pmatrix} \Delta s^k \\ \Delta\theta^k \end{pmatrix} + F_2^c \end{aligned}$$

using the methods presented in Subsection 6.2.1.

4. *If* $k = 0$, *set* $\alpha^k = \alpha_0$ *otherwise determine the step length* $\alpha^k \in (0,1]$ *applying the RMT as presented in Section 4.3.5.*

5. *Update the iterate*
$$\begin{pmatrix} s^{k+1} \\ \theta^{k+1} \end{pmatrix} := \begin{pmatrix} s^k \\ \theta^k \end{pmatrix} + \alpha^k \begin{pmatrix} \Delta s^k \\ \Delta\theta^k \end{pmatrix}.$$

6. *If* θ^{k+1} *violates any parameter constraints, recompute the step length* $\tilde{\alpha}^k$ *applying the bound check Algorithm 6.2.1 from Section 6.2.2 and update*
$$\begin{pmatrix} s^{k+1} \\ \theta^{k+1} \end{pmatrix} := \begin{pmatrix} s^k \\ \theta^k \end{pmatrix} + \tilde{\alpha}^k \begin{pmatrix} \Delta s^k \\ \Delta\theta^k \end{pmatrix}.$$

7. *If*
$$\left\| \begin{pmatrix} \Delta s^k \\ \Delta\theta^k \end{pmatrix} \right\|_2 < TOL$$
terminate, otherwise:

8. *Set* $k := k+1$. *Go to 2.*

6.3 Test of the reduced approach by means of academic application examples

In this section, we present the results for two academic application examples that are already adequate enough to highlight the features of *PAREMERA*. The first is a 1D heat equation with homogeneous boundary equations and the second one is a home run during a baseball game where we have two model stages with a state dependent switching function.

6.3.1 Heat Equation

By means of the first application example, we highlight the advantages of the reduced approach in computational time. We examine a 1D heat equation, see Evans [38], on the domain $\Omega = [0, 2\pi]$ and the time interval $t \in [0,1]$

$$u_t = \theta_1 u_{xx}, \tag{6.45a}$$
$$u(t,0) = u(t,2\pi) = 0, \tag{6.45b}$$
$$u(0,x) = \sin x. \tag{6.45c}$$

The initial value function is illustrated in Figure 6.1. Here, u is an arbitrary function,

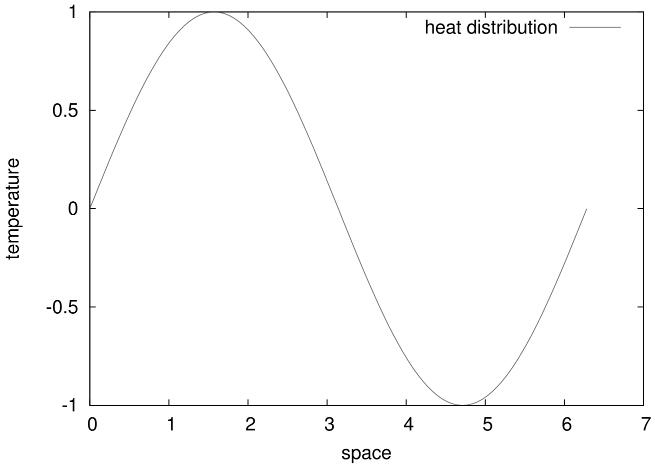


Figure 6.1: Initial heat distribution.

usually referred to as the temperature. The parameter θ_1 is referred to as thermal diffusivity. The exact solution is well known

$$u(t,x) = e^{-\theta_1 t} \sin x.$$

We discretize (6.45) with second order central finite differences with $\triangle x = \frac{\pi}{50}$ and obtain an ODE system with $n_x = 101$ states. We measure the heat at $x_1 = \frac{\pi}{2}$ and $x_2 = \frac{3\pi}{2}$ at $t \in \{0.25,\ 0.5,\ 0.75,\ 1.0\}$. The software package DAESOL by Bauer [8] is applied to generate measurement data by integrating the ODE system with the thermal diffusivity set to

$$\theta_1^* = 1.0$$

and adding Gaussian noise.

We compare the results of the parameter estimation for three different starting parameters $\theta_1^0 \in \{2,\ 10,\ 25\}$ computed using the reduced approach implemented in *PAREMERA* with the results of the parameter estimation software *PARFIT*. Here, we use the version that is capable of exploiting multiple experiment structures, see [58]. Both methods are callable by the software package *VPLAN* by Körkel et al. [58]. For both packages the integration of the ODE system and the variational differential equations are evaluated by the previously mentioned software package *DAESOL*. The computations have been performed on a 64-bit computer with 4GB memory and an Intel® Core2Duo with 2 × 2.8GHz. The results are listed in Table 6.1. For all starting parameters *PAREMERA* converges to the same solution while *PARFIT* only converges for $\theta_1^{0,1} = 2$. This can be explained by the different globalization strategies. In *PARFIT*, a first order Taylor approximation of the curvature information (4.33) is implemented. The computed step sizes for the generalized Gauss–Newton iterations are too long and $\theta_1^{k,i},\ i = 2,3$ becomes negative. No such thing as a bound check routine is applied in *PARFIT*. For negative θ_1, Equation (6.45) does not have a stable solution and the integration of the ODE system fails.

If we compare the computational time for one iteration, we see that *PAREMERA* is around five to six times faster then *PARFIT*. In *PARFIT*, we have to evaluate 101 ($\widehat{=}\, n_x$) variational

		PAREMERA	*PARFIT*
$\theta_1^1 = 2$	# of iterations	5	8
	time per iteration	0.068 sec – 0.08 sec	0.304 sec – 0.388 sec
	$\hat{\theta}_1^1$	0.996622	0.996266
	residual	2.12034	2.12748
$\theta_1^2 = 10$	# of iterations	8	3, abort
	time per iteration	0.068 sec – 0.1 sec	0.612 sec – 0.676 sec
	$\hat{\theta}_1^2$	0.99662	no convergence
	residual	2.12035	no convergence
$\theta_1^3 = 25$	# of iterations	9	2, abort
	time per iteration	0.068 sec – 0.132 sec	0.576 sec – 0.728 sec
	$\hat{\theta}_1^3$	0.996621	no convergence
	residual	2.12035	no convergence

Table 6.1: Survey of the results.

differential equations more then in *PAREMERA*. For more complex models, this factor will get even bigger. Thus, the application of the reduced approach is indispensable for an efficient treatment of PDE constrained parameter estimation problems. The reason for the differences in the computational time (0.068 up to 0.132 seconds for $\theta_1^{0,3} = 25$) between the single iterations in *PAREMERA* is based in the stiffness of the system. For large values of θ_1, (6.45) becomes very stiff and the step size for the simulation of (6.45) has to be very small. In Figure 6.2, the measurement fits for $\theta_1^{end,3} = 0.996621$ are displayed. We see that the model responses fit the data very well.

6.3.2 Home run

The second example, that we consider, is a pitch and the parabolic trajectory of a baseball after the hit. Without loss of generality, suppose we are in Chicago at Wrigley Field to watch a baseball game. The pitcher stands on the pitcher's mound and throws a fastball (a horizontal pitch). At the home plate, there is the batter who hits the ball perfectly and scores a home run. We want to model that process.

For simplicity, we assume that the ball flies straight from pitcher to batter and straight over the head of the pitcher after the hit. We end up with a 2D model. We model both curves by

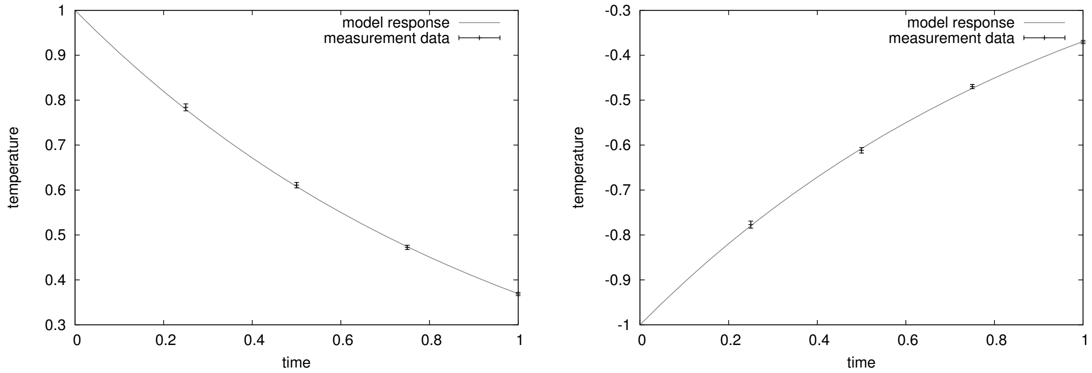


Figure 6.2: Data fit: The left figure illustrates the data and model response observed at $x = \frac{\pi}{2}$ and the right figure at $x = \frac{3\pi}{2}$.

a parabolic trajectory with friction, see Gerthsen et al. [42],

$$\dot{x}(t) = v_x(t), \qquad x(t_0) = x_0, \tag{6.46a}$$

$$\dot{v_x}(t) = -\frac{1}{2 \cdot m_b} \rho_{air} c_w A_b \sqrt{v_x^2(t) + v_y^2(t)} v_x(t), \qquad v_x(t_0) = v_x^0, \tag{6.46b}$$

$$\dot{y}(t) = v_y(t), \qquad y(t_0) = y_0, \tag{6.46c}$$

$$\dot{v_y}(t) = -\frac{1}{2 \cdot m_b} \rho_{air} c_w A_b \sqrt{v_x^2(t) + v_y^2(t)} v_y(t) - g, \quad v_y(t_0) = v_y^0, \tag{6.46d}$$

where m_b refers to the weight of the baseball, ρ_{air} denotes the density of air, c_w is the drag coefficient of the ball, A_b indicates the cross section area of the baseball and g represents the gravitational acceleration We presume that the baseball is perfectly round.

We demand that the trajectory remains continuous, i.e., the end point of the pitch is the starting point of the hit. At the moment, when the batter hits the ball, it changes its direction and its velocity instantaneously. Since we do not know the exact velocity of the ball, we do not know when the ball will reach the batter. Thus, we do not know when to switch. We end up with a state dependent switching function. To overcome this difficulty, we separate the two stages of the model into two processes (experiments) with free end time. The time becomes an additional state variable

$$t^i(\hat{t}) = \frac{d^i}{v_x^i(\hat{t})} \hat{t}, \; i = 1,2, \quad \hat{t} \in [0,1], \tag{6.47}$$

where d^i denotes the distance between the pitcher and the batter and the batter and the outfield wall, respectively. We use the real distances of Wrigley Field, see Wikipedia [101]. The distance from the center of the pitcher's mound to the center of the home plate is around $d_1 = 18,29m$ $(\approx 60ft)$. The outfield wall at the center field is approximately $d^2 = 121.92m$ $(\approx 400ft)$ away from the home plate. In addition, we have to transform Equations (6.46) by

$$\frac{d}{d\hat{t}} x\left(t(\hat{t})\right) = \dot{x}(t) \frac{dt}{d\hat{t}}.$$

The continuity of the trajectory is assured by formulation (6.5) of the multi-experiment interior point constraints (6.1e). We define

$$r_1 := \begin{pmatrix} t^1(1) \\ x^1(1) \\ y^1(1) \end{pmatrix}, \quad r_2(0) := \begin{pmatrix} -s_1^r \\ -s_2^r \\ -s_3^r \end{pmatrix}$$

and add

$$\sum_{i=1,2} r_i = 0 \tag{6.48}$$

as an additional constraint to our problem. The variables s_i^r are the initial states for $t^2(0)$, $x^2(0)$ and $y^2(0)$ respectively.

We want to estimate three parameters. The velocity of the ball when it leafs the hand of the pitcher at $\hat{t}^1 = 0$ is referred to as θ_1 while θ_2 and θ_3 denote the initial angle between the trajectory and the ground and the norm of the velocity of the ball, respectively, when the batter hits it at $\hat{t}^2 = 0$.

To solve the model, given by Equations (6.46), we have to define initial values for the states. Since we only know the norm of the initial velocity, we have to define a further constraint which transforms the norm to the values of the velocity in both directions. This can be done by the Pythagorean theorem

$$r^3 := \begin{pmatrix} \theta(3) \cdot \cos(\theta(2)) - s_4^r \\ \theta(3) \cdot \sin(\theta(2)) - s_5^r \end{pmatrix} \doteq 0. \tag{6.49}$$

Similar to (6.48) the variables s_4^r and s_5^r are the initial values for v_x^2 and v_y^2 respectively. We summarize the initial states

$$\begin{gathered} t^1(0) = 0,\ x^1(0) = 60,\ v_x^1(0) = \theta_1,\ y^1(0) = 1.8,\ v_y^1(0) = 0, \\ t^2(0) = s_1^r,\ x^2(0) = s_2^r,\ v_x^2(0) = s_4^r,\ y^2(0) = s_3^r,\ v_y^2(0) = s_5^r. \end{gathered}$$

We execute four measurements. We measure the velocity of the ball in x-direction at $\hat{t}^1 = 1$, when it reaches the home plate, the angle of the trajectory, namely parameter θ_2, at $\hat{t}^2 = 0$ and the position of the ball at the end of the time horizon at $\hat{t}^2 = 1$. As in the first application example in Subsection 6.3.1, we use simulated data so that the "*true*" parameter values are known

$$\theta_1^* = -44{,}7\frac{m}{s},\ \theta_2^* = 30^\circ,\ \theta_3^* = 40.27\frac{m}{s}. \tag{6.50}$$

Since this example should serve as a proof of concepts for the treatment of multi-experiment interior point constraints, we show results for one starting set of parameters

$$\theta_1^0 = -60\frac{m}{s},\ \theta_2^0 = 90^\circ,\ \theta_3^0 = 60\frac{m}{s}. \tag{6.51}$$

Additionally, we have to state initial values for the vector s^r.

$$s_1^{r,0} = 0.0,\ s_2^{r,0} = 0.0,\ s_3^{r,0} = 0.0,\ s_4^{r,0} = 10.0,\ s_5^{r,0} = 10.0. \tag{6.52}$$

I	EQ	LS	INCREM	RELAX	RK
1	$2.60188 \cdot 10^{3}$	$6.28593 \cdot 10^{7}$	$1.23680 \cdot 10^{7}$	$8.52 \cdot 10^{-5}$	3
2	$2.60143 \cdot 10^{3}$	$6.28486 \cdot 10^{7}$	$1.22168 \cdot 10^{7}$	$1.59 \cdot 10^{-2}$	3
3	$2.52520 \cdot 10^{3}$	$6.15967 \cdot 10^{7}$	$1.66937 \cdot 10^{6}$	$4.70 \cdot 10^{-2}$	3
4	$2.31467 \cdot 10^{3}$	$5.75224 \cdot 10^{7}$	$3.75492 \cdot 10^{4}$	$7.33 \cdot 10^{-2}$	3
5	$2.00022 \cdot 10^{3}$	$5.03770 \cdot 10^{7}$	$2.18587 \cdot 10^{4}$	$1.76 \cdot 10^{-1}$	3
6	$1.36901 \cdot 10^{3}$	$3.65040 \cdot 10^{7}$	$4.34467 \cdot 10^{4}$	1.0	3
7	$3.71270 \cdot 10^{2}$	$5.01891 \cdot 10^{6}$	$1.85598 \cdot 10^{2}$	1.0	3
8	$4.96001 \cdot 10^{2}$	$6.09286 \cdot 10^{5}$	$6.01948 \cdot 10^{2}$	1.0	3
9	$1.95987 \cdot 10^{2}$	$5.10551 \cdot 10^{4}$	$2.20908 \cdot 10^{2}$	1.0	3
10	$1.69509 \cdot 10^{1}$	$2.55719 \cdot 10^{3}$	$4.53835 \cdot 10^{1}$	1.0	3
11	$2.17587 \cdot 10^{-1}$	$2.87520 \cdot 10^{1}$	$8.74512 \cdot 10^{-1}$	1.0	3
12	$5.16637 \cdot 10^{-5}$	$7.48120 \cdot 10^{-1}$	$2.17827 \cdot 10^{-4}$	1.0	3
13	$3.04598 \cdot 10^{-12}$	$7.41757 \cdot 10^{-1}$	$1.18196 \cdot 10^{-11}$	1.0	3

Table 6.2: Progress of the parameter estimation. ”I” denotes the current number of iterations, ”EQ” is the squared violation of the multi-experiment interior point constraints and the continuity conditions and ”LS” refers to the squared weighted sum of the difference between measurement data and model response. ”INCREM” stands for the squared norm of the increment $\left((\Delta s^k)^T,\ (\Delta\theta^k)^T\right)^T$ and ”RELAX” identifies the step length α^k. ”RK” represents the rank of the block E^{θ} of the Jacobian.

Table 6.2 illustrates the output of *PAREMERA*. For the starting parameters (6.51) and the initial guesses for s^r (6.52), the residual given by LS is very large. The violation of Equations (6.48) and (6.49) illustrated by EQ is substantial. As the iteration proceeds, the residual and the violation of the constraints get smaller until we reach the minimum at

$$\hat{\theta}_1 = -45.0162 \pm 0.450089 \quad (1.00)\%, \tag{6.53a}$$

$$\hat{\theta}_2 = 30.5034 \pm 0.305034 \quad (1.00)\%, \tag{6.53b}$$

$$\hat{\theta}_3 = 40.4874 \pm 0.133590 \quad (0.33)\%, \tag{6.53c}$$

with given confidence intervals. Starting at iteration 6 on, the RMT suggests full steps for the step length of the generalized Gauss–Newton method. Because we used simulated data, we obtain the expected, very good result for the parameter estimation problem. For completeness we state the final components of the vector s^r

$$s^r_1 = 0.412403,\ s^r_2 = 6.6876 \cdot 10^{-14},\ s^r_3 = 0.973991,\ s^r_4 = 34.8839,\ s^r_5 = 20.551. \tag{6.54}$$

It is easy to see that (6.54) fulfill the constraints (6.48) and (6.49). In Figure 6.3, we display the resulting parabolic trajectory of the baseball with the estimated parameters (6.53). We observe that the switch from the pitch (stage 1) to the hit and the subsequent flight (stage 2) is continuous and that we have solved the parameter estimation problem with state dependent switching function.

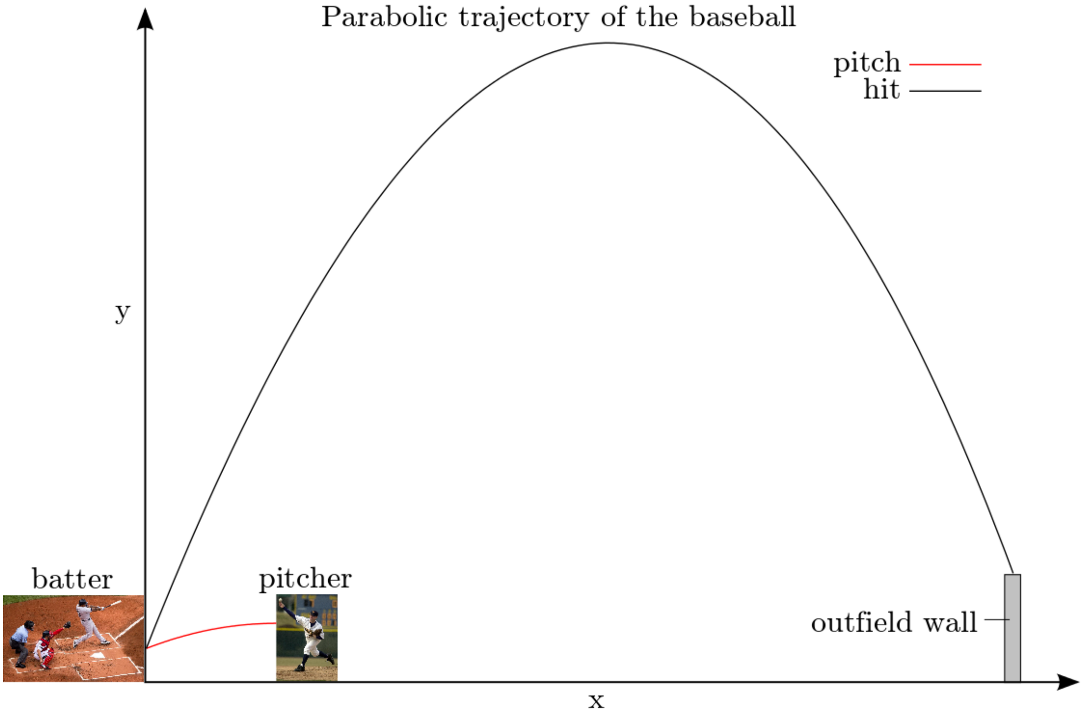


Figure 6.3: Fitted curve of the pitch and after the following hit. The picture of the pitcher is taken form `http://commons.wikimedia.org/wiki/File:Baseball_pitch_release.jpg` and is public domain, while the picture of the batter can be found on `http://commons.wikimedia.org/wiki/File:Red_Sox_Yankees_Game_Boston_July_2012.jpg` and was taken by Victor Gigras and added to the Creative Commons Attribution.

7 Numerical results for microbial enhanced oil recovery

In this chapter, we apply the methods of optimum experimental design and parameter estimation presented in Chapters 5 and 6, respectively, to our model for microbial enhanced oil recovery. We illustrate the procedure of sequential optimum experimental design and explain the setup of the considered experiments. The procedure is applied to the model of microbial enhanced oil recovery and numerical results are presented to conclude the chapter.

7.1 Model equations for microbial enhanced oil recovery

In Chapter 2, we introduced a coupled model that is a combination of the black oil model for two phase flow through porous media

$$\nabla\cdot(u_\mathrm{a}+u_\mathrm{c}) \equiv -\nabla\cdot\lambda_\mathrm{t}K\nabla\Phi_\mathrm{w} - \nabla\cdot\lambda_\mathrm{o}K\nabla\Phi_\mathrm{c} = 0, \tag{7.1a}$$

$$\phi\frac{\partial S_\mathrm{w}}{\partial t} - q_\mathrm{w} = -\nabla\cdot(f_\mathrm{w}u_\mathrm{a}) \equiv \nabla\cdot\lambda_\mathrm{w}K\nabla\Phi_\mathrm{w}, \tag{7.1b}$$

and the transport equations for bacteria, nutrients, gas and metabolites

$$\frac{\partial(\phi S_\mathrm{w}C)}{\partial t} = -\nabla\cdot(u_\mathrm{w}C) + \phi S_\mathrm{w}(r_\mathrm{g}-r_\mathrm{d})C - k_\mathrm{c}\phi S_\mathrm{w}C, \tag{7.2a}$$

$$q_1 = \sqrt{S_\mathrm{w}D_\mathrm{f}}\cdot\nabla C_\mathrm{f}, \tag{7.2b}$$

$$\begin{aligned}\frac{\partial(\phi S_\mathrm{w}C_\mathrm{f})}{\partial t} &= \phi\nabla\cdot[C_\mathrm{f}D_\mathrm{f}\cdot\nabla S_\mathrm{w}] + \phi\nabla\cdot\left[\sqrt{S_\mathrm{w}D_\mathrm{f}}\cdot q_1\right]\\ &\quad -\nabla\cdot(u_\mathrm{w}C_\mathrm{f}) - \phi S_\mathrm{w}u_\mathrm{f}r_\mathrm{g}C,\end{aligned} \tag{7.2c}$$

$$\frac{\partial(\phi S_\mathrm{o}C_\mathrm{g})}{\partial t} = -\nabla\cdot(u_\mathrm{o}C_\mathrm{g}) + \phi S_\mathrm{w}u_\mathrm{g}\frac{r_\mathrm{g}C}{Y_\mathrm{g}}, \tag{7.2d}$$

$$q_2 = \sqrt{S_\mathrm{w}D_\mathrm{m}}\cdot\nabla C_\mathrm{m}, \tag{7.2e}$$

$$\begin{aligned}\frac{\partial(\phi S_\mathrm{w}C_\mathrm{m})}{\partial t} &= \phi\nabla\cdot[C_\mathrm{m}D_\mathrm{m}\cdot\nabla S_\mathrm{w}] + \phi\nabla\cdot\left[\sqrt{S_\mathrm{w}D_\mathrm{m}}\cdot q_2\right]\\ &\quad -\nabla\cdot(u_\mathrm{w}C_\mathrm{m}) - \phi S_\mathrm{w}u_\mathrm{m}\frac{r_\mathrm{g}C}{Y_\mathrm{m}}.\end{aligned} \tag{7.2f}$$

Here, we apply the first order formulation presented in Section 3.1. We use the mixed discontinuous Galerkin finite element method derived in Chapter 3 to transform the model equation into the weak formulation. Additionally, we recall the modified Brooks–Corey relations for the relative permeabilities

$$k_{\mathrm{ro}} = \hat{k}_{\mathrm{ro}} \left(\frac{S_{\mathrm{o}} - S_{\mathrm{or}}}{1 - S_{\mathrm{or}} - S_{\mathrm{wc}}} \right)^{c_{\mathrm{o}}}, \tag{7.3a}$$

$$k_{\mathrm{rw}} = \hat{k}_{\mathrm{rw}} \left(\frac{S_{\mathrm{w}} - S_{\mathrm{wc}}}{1 - S_{\mathrm{or}} - S_{\mathrm{wc}}} \right)^{c_{\mathrm{w}}} \tag{7.3b}$$

and the growth term with the Monod equation

$$r_{\mathrm{g}} = \left(1 - \left(\frac{C}{C_{\max}} \right)^2 \right) \cdot \mu_{\max} \left(\frac{C_{\mathrm{f}}}{K_{\mathrm{S}} + C_{\mathrm{f}}} \right) \tag{7.4}$$

since these expressions contain the four parameters we want to estimate. The parameters are listed below

$$\theta_1 = c_o,\ \theta_2 = c_w,\ \theta_3 = \mu_{\max},\ \theta_4 = K_{\mathrm{S}}.$$

As mentioned in Section 2.3, we consider one control variable, the inflow velocity q^{in}, that defines the Neumann-type boundary condition at the inflow boundary for the pressure equation (7.1a) and two control variables, namely the injected bacteria concentration C^{in} and the injected nutrient concentration $C_{\mathrm{f}}^{\mathrm{in}}$ that specify the Dirichlet-type boundary conditions at the inflow boundary for the transport equation of bacteria (7.2a) and nutrients (7.2c), respectively.

7.2 Sequential optimum experimental design and parameter estimation

Experimental design problems depend on the current estimated parameter values. Due to high nonlinearities, even small perturbations in the model may lead to inadequate results of the optimum experimental design.

Körkel et al. [59] presented a procedure called sequential optimum experimental design and parameter estimation that we want to apply to our model (7.1)–(7.2). Thereby we design experiments, execute them and estimate the parameters in a repeating order. By optimum experimental design, we maximize the gain of information to successively improve the quality of identified parameter values.

We proceed in the following manner, see also Figure 7.1 for a conceptional illustration.

Algorithm 7.2.1. ***Sequential optimum experimental design***

1. *Start with an initial guess for the parameters θ^0 and set $k := 1$.*

2. *Plan n_{ex}^k new experiments in consideration of all already executed experiments and the current set of parameters θ^{k-1}.*

3. *The n_{ex}^k are executed either in the laboratory or at the computer to produce measurement data.*

4. *Estimate the parameters for all $\sum_{i=1}^{k} n_{\text{ex}}^i$ experiments. Use θ^{k-1} as starting value.*

5. *If a predefined convergence criterion on the variance–covariance matrix is not fulfilled, set $k := k+1$ and go to step 2.*

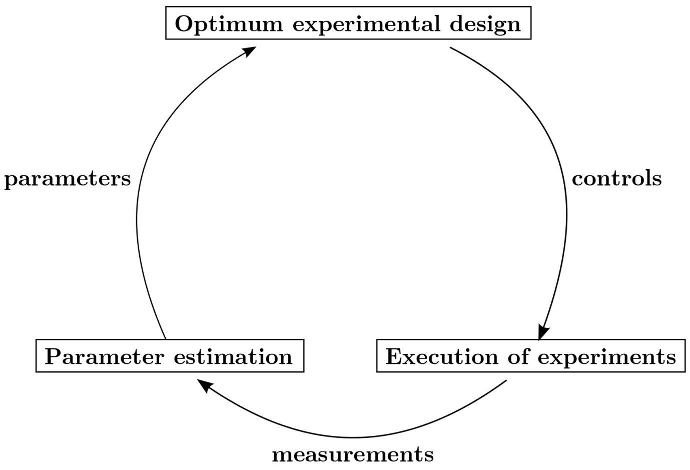


Figure 7.1: Scheme for sequential optimum experimental design.

7.3 Experimental setup

In Figure 7.2, we illustrate the experimental setup. In the middle, we have a core sample that is fixed in a core holder. The core holder controls the temperature around the core. A dosing system is connected to the bottom of the core. It controls the inflow velocity and the injected concentrations of the bacteria and the nutrients. A collecting vessel is attached to the top of the core. It collects the outflow which can be measured. By ports at the side of the core we can execute some additional measurements from inside the core. We do not use a core sample from an oil reservoir but a tube filled with glass beads; a so called *sandpack*. Sandpacks have the advantage that we know the properties, e.g., the shape of the pores, the porosity ϕ and the permeability K, very well since the dimensions of the glass beads of known exactly.

One sandpack experiment contains the following steps, see Kögler [57]. In the beginning, the sandpack is soaked with oil up to the connate water saturation. We inject water into the sandpack over a predefined time interval. This phase is called initial water flood.

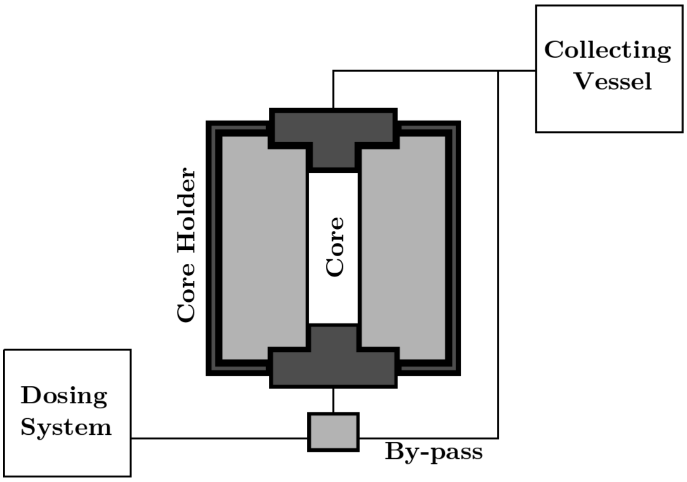


Figure 7.2: Setup of a sandpack experiment.

Afterwards, the bacteria inoculation starts. A water mixed with bacteria and nutrients is injected followed by an incubation phase which lasts a few days. Subsequently, the bacteria are stimulated again by a first nutrient injection. Then, after an additional incubation phase, a second nutrient injection follows. The sequence is completed by a final water flood. During the whole process, the difference of the pressure between the top and the bottom of the sandpack is measured at specified time points as well as the recovered oil. As mentioned above, the core holder has ports at its side that allow the sampling of the water phase from inside the sandpack to measure bacteria, nutrients or metabolite concentration. The three quantities may also be measured at the outflow boundary of the sandpack. Figure 7.3 shows the chronological progress of a sandpack experiment for the initial water flood.

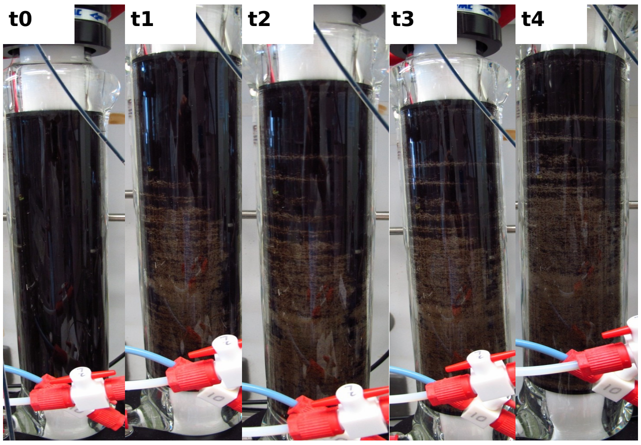


Figure 7.3: Sandpack experiment during the initial water flood at different, unspecified time points. Copyright: Felix Kögler, Wintershall Holding GmbH, EOT/R.

Definition 7.3.1. *A sequence consisting of the above described parts is called experiment.*

7.4 Results for the MEOR model

In this section, we exemplarily execute two cycles of the sequential optimum experimental design and parameter estimation presented in Section 7.2. Again, we assume that there is no capillary pressure. All computations are done using simulated data, i.e., we know the ”*true*” parameters from the beginning.

The ”*true*” parameters are set to

$$c_{\mathrm{o}} = 2.0,\ c_{\mathrm{w}} = 1.6,\ \mu_{\max} = 8.3,\ K_{\mathrm{S}} = 0.06. \tag{7.5}$$

7.4.1 First parameter estimation

Assume that the experimenters have executed one experiment which provides a set of measurement data. We consider a sandpack experiment as described in Section 7.3 without incubation. We examine a one-dimensional bounded domain $\Omega = [0,31]$. The time interval is set to $t \in [0,24]$.

The inflow rate q^{in} is constant over the whole time interval

$$q^{\mathrm{in}} = -0.212758715, \quad t \in [0,24],$$

and the inflow concentrations of the bacteria and the nutrients are piecewise constant. The temporal profile of both control functions is depicted in Equation (7.6)

$$C^{\mathrm{in}} = \begin{cases} 0.0, & t \in [0,13.5], \\ 0.01, & t \in (13.5,16.5], \\ 0.0, & t \in (16.5,24], \end{cases} \qquad C_{\mathrm{f}}^{\mathrm{in}} = \begin{cases} 0.0, & t \in [0,13.5], \\ 0.05, & t \in [13.5,16.5], \\ 0.08, & t \in (16.5,19], \\ 0.05, & t \in (19,21.5], \\ 0.0, & t \in (21.5,24.0]. \end{cases} \tag{7.6}$$

Similar to the 2D example from Section 3.5, the initial water saturation is set to $S^0 = 0.079$ and the initial concentration of the bacteria, nutrients, gas and metabolites are set to zero. The injected saturation of the water phase is set to $S_{\mathrm{w}}^{\mathrm{in}} = 0.95$ and the pressure at the outflow boundary is equal standard pressure.

In Section 7.3, we described different methods to observe the process. We consider four types of measurements. We measure:

1. h_1 : the difference between the bottom and the top of the sandpack, also referred to as the pressure gradient at time points

   $$t \in \{1,2,3,4,5,6,7,8,9,10,11,12,13,15,16,17,18,19,20,21,22,23,24\},$$

2. h_2 : the volume of recovered oil at time points

   $$t \in \{1,2,3,4,5,6,7,8,9,10,11,12,13,15,16,17,18,19,20,21,22,23,24\},$$

3. h_3 : the concentrations of bacteria, nutrients and metabolites inside the sandpack through a port at $x = 5$ at time points

$$t \in \{17, 18, 19, 20, 21, 22, 23, 24\},$$

4. h_4 : and the concentrations of bacteria, nutrients and metabolites through another port at $x = 25$ at time points

$$t \in \{20, 21, 22, 23, 24\}.$$

This sums up to 85 measurements. Note, the concentrations of bacteria, nutrients and metabolites, respectively can not be measured separately.

The fixed model parameters are listed in Table 7.1. For the choice of the starting values θ^0 of the parameter estimation problem, we pretend that we have only a rough idea of the region where the true parameters are located. The two exponents c_o and c_w of the modified Brooks–Corey relation for the relative permeabilities (7.3) have to be greater than zero. The same holds for the maximum growth rate μ_{max} of the Monod kinetics (7.4) while the half rate constant K_{S} is to be found in the interval $[0, 1]$.

$\hat{k}_{\text{ro}}$	$\hat{k}_{\text{rw}}$	K	$\hat{\mu}_{\text{o}}$	$\hat{\mu}_{\text{w}}$	μ_{g}	μ_{m}	S_{ro}	S_{wc}	ϕ
1.18	0.39	10.47	42.17	1.18	0.02	20.10	0.05	0.037	0.399

C_{max}	r_{d}	k_{c}	u_f	u_g	u_m	Y_g	Y_m
0.05	0.25	0.0	0.5	0.45	0.05	3.0	1.0

Table 7.1: Parameter settings

Following these argumentations, a reasonable guess for the starting values of the parameters that we want to estimate is

$$c_{\text{o}}^0 = 1.0,\ c_{\text{w}}^0 = 1.0,\ \mu_{\text{max}}^0 = 1.0,\ K_{\text{S}}^0 = 0.01.$$

The progress of the parameter estimation algorithm is presented in Table 7.2. We see that the Jacobian has full rank (*RK*) for all iterations, i.e., the four parameters are identifiable by the measurements. The least-squares functional (*LS*) is decreasing until we reach a final residual 88.5235 for the 85 measurements. The step length per iteration is shown in the column *RELAX*. We start relatively far from the solution so in the beginning the step length is small. While the algorithm advances the step length increases up to full step close to solution.

We obtain the following values for the vector of estimated parameters

$$\hat{c}_{\text{o}} \quad = 2.16745 \quad \pm 0.20019 \qquad 9.24\%, \tag{7.7a}$$

I	LS	INCREM	RELAX	RK
1	$1.23201 \cdot 10^6$	$1.00367 \cdot 10^3$	$3.16 \cdot 10^{-3}$	4
2	$1.22721 \cdot 10^6$	$1.00435 \cdot 10^3$	$1.00 \cdot 10^{-1}$	4
3	$9.36348 \cdot 10^5$	$6.62681 \cdot 10^2$	$1.00 \cdot 10^{-1}$	4
4	$7.69562 \cdot 10^5$	$7.06049 \cdot 10^1$	$2.87 \cdot 10^{-1}$	4
5	$4.29877 \cdot 10^5$	$7.47752 \cdot 10^0$	$3.16 \cdot 10^{-1}$	4
6	$2.60316 \cdot 10^5$	$4.71281 \cdot 10^0$	1.0	4
7	$5.82880 \cdot 10^4$	$1.63041 \cdot 10^1$	1.0	4
8	$2.54705 \cdot 10^4$	$8.92830 \cdot 10^1$	1.0	4
9	$4.71851 \cdot 10^3$	$8.68825 \cdot 10^0$	$9.64 \cdot 10^{-1}$	4
10	$3.42004 \cdot 10^3$	$4.27387 \cdot 10^0$	1.0	4
11	$3.50423 \cdot 10^2$	$2.53653 \cdot 10^0$	1.0	4
12	$1.15375 \cdot 10^2$	$1.05841 \cdot 10^0$	1.0	4
13	$8.91391 \cdot 10^1$	$1.85997 \cdot 10^{-2}$	1.0	4
14	$8.85252 \cdot 10^1$	$1.17253 \cdot 10^{-4}$	1.0	4
15	$8.85235 \cdot 10^1$	$2.65125 \cdot 10^{-7}$	1.0	4

Table 7.2: Progress of the first parameter estimation run. The headers of the table are the same as in Table 6.2.

$$\hat{c}_\mathrm{w} = 1.59567 \pm 0.00930518 \quad 0.58\%, \tag{7.7b}$$

$$\hat{\mu}_\mathrm{max} = 8.29163 \pm 0.455538 \quad 5.49\%, \tag{7.7c}$$

$$\hat{K}_\mathrm{S} = 0.0598582 \pm 0.000120482 \quad 0.20\%. \tag{7.7d}$$

All four estimated parameters are good fits for the true parameters (7.5). The standard deviation for c_o is comparatively large with around nine percent. The standard deviation of c_w and K_S is very small (lower than one percent). Nevertheless, K_S^* does not lie in the computed confidence interval. This can be explained by the fact that we compute a first order approximation of the confidence interval and the model responses depend on the parameter K_S in a nonlinear way.

Figure 7.4 illustrates the fits of the different model responses to the data. There are eight subplots one for each model response. The gray lines refer to the different model responses and the black bars depict the executed measurements with corresponding error bars. Line by line from the top left to the bottom right we have the model responses and measurement values of the pressure gradient, the recovered oil, the bacteria concentration at $x = 5$ and at $x = 25$, the nutrient concentration at $x = 5$ and at $x = 25$ and the metabolite concentration at $x = 5$ and at $x = 25$.

We deduce that the model describes the process quiet well but that the model responses do not hit all error bars. Especially for the subfigure in the top left corner we see that not all measurements are fitted satisfactorily in a desired manner which is probably caused by the measurement value of the pressure gradient at $t = 15$ that obviously possesses a large measurement error.

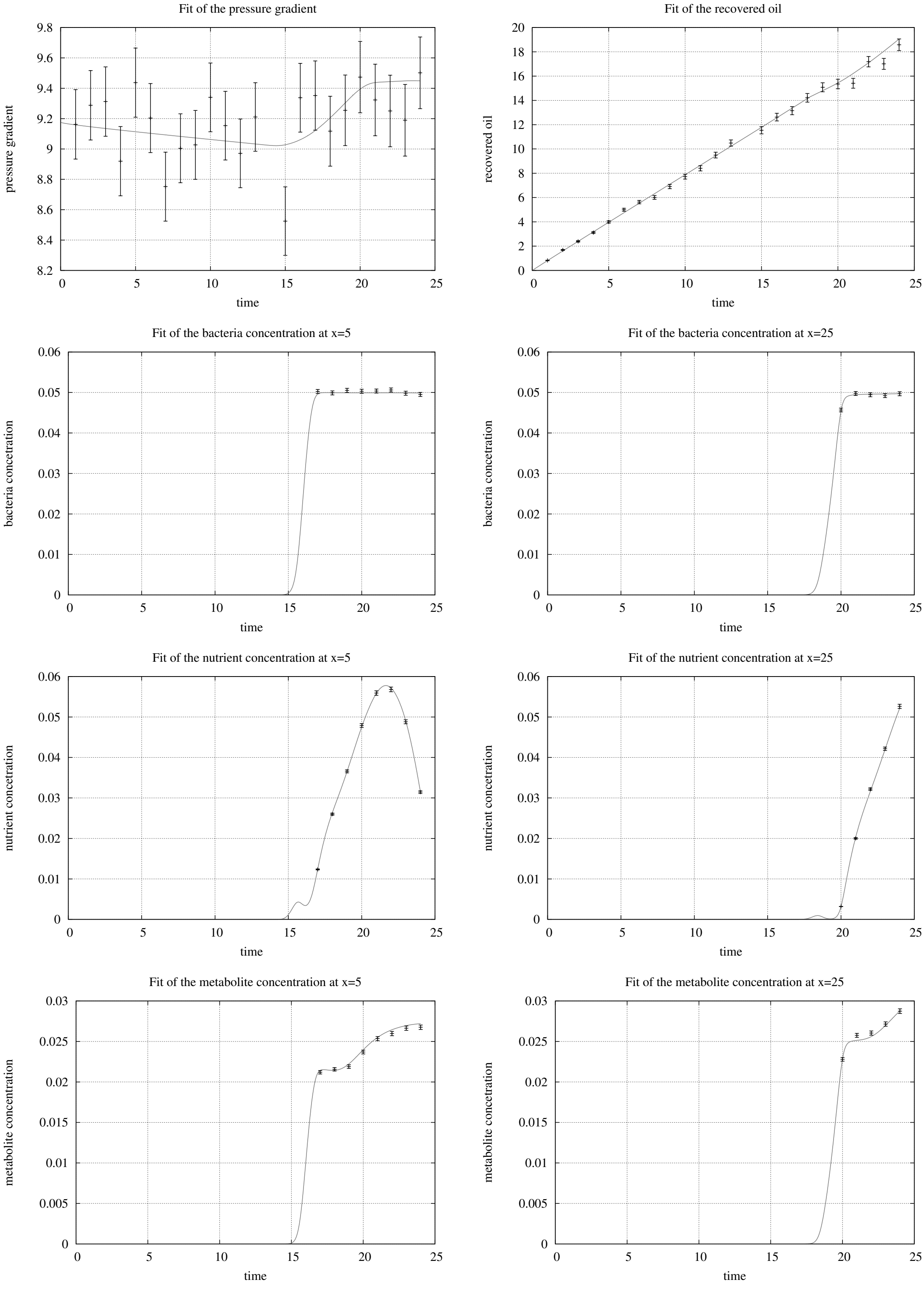


Figure 7.4: Data fit for the first experiment after the first parameter estimation.

7.4.2 Design of additional experiments

We plan a second experiment to shorten the length of the linearized confidence intervals of the estimated parameters. In words, we try to find experimental settings such that the information gain is maximized with respect to a criterion on the variance–covariance matrix as presented in Section 5.1.

Therefore, we define bounds for the controls according to physical, biological and practical reasons, see (5.3). The inflow velocity has to be chosen out of the interval

$$q^{\text{in}} \in [-0.5, 0.0].$$

We assume the control functions, i.e, the injected bacteria and nutrient concentrations, to be piecewise constant and define four switching points at

$$t \in \{13.5, 16.5, 19.5, 21\}.$$

For the control functions, we define interval-wise constraints

$$C^{\text{in}} = \begin{cases} [0.0, 0.0], & t \in [0, 13.5], \\ [0.0, 0.05], & \textit{else}, \end{cases} \qquad C_{\text{f}}^{\text{in}} = \begin{cases} [0.0, 0.0], & t \in [0, 13.5], \\ [0.0, 0.1], & \textit{else}. \end{cases}$$

Additionally, we define bounds on the measurements weights, see (5.2), since we do not want to measure as often as in the first experiment. The possible time points for each measurement stay the same as in the first experiment. We denote the weights corresponding to the measurement functions defined in section 7.4.1 by $w_j,\ j = 1, \ldots, 4$ and state the bounds for the sums of each measurement weight

$$\sum_{i=1}^{24} w_1^i = 12, \quad \sum_{i=1}^{24} w_2^i = 12, \quad \sum_{i=1}^{8} w_3^i = 4, \quad \sum_{i=1}^{5} w_4^i = 3, \tag{7.8}$$

where $w_j^i \in \{0, 1\},\ j = 1, \ldots, 4$ for all time points.

For the computations, we have to relax the integral constraints for the measurement weights

$$w_j^i \in [0, 1],\ j = 1, \ldots, 4.$$

Afterwards, we use rounding strategies to retransform the measurement weights to integrity. For an overview of different strategies, we refer to Körkel [58].

We use the A-criterion as information function on the predicted variance-covariance matrix and include the first (fixed) experiment as suggested in step 2 of algorithm 7.2.1. In *VPLAN*, the computation of multiple experiments can be done in parallel such that the evaluation of the Jacobians for each experiment requires almost the same computational time as for a single experiment.

The results for the controls of the optimum experimental design problem are listed in Equation (7.9)

$$q^{\text{in},*} = -0.126884, \quad t \in [0, 24], \tag{7.9a}$$

$$C^{\mathrm{in},*} = \begin{cases} 0.0, & t \in [0,16.5], \\ 7.08635 \cdot 10^{-5}, & t \in (16.5,19], \\ 0.000687064, & t \in [19,21.5) \\ 0.000155375, & t \in (21.5,24]. \end{cases} \tag{7.9b}$$

$$C_{\mathrm{f}}^{\mathrm{in},*} = \begin{cases} 0.0, & t \in [0,13.5], \\ 0.0104577, & t \in [13.5,16.5], \\ 0.0311397, & t \in (16.5,19], \\ 0.0160987, & t \in (19,21.5], \\ 0.0649456, & t \in (21.5,24]. \end{cases} \tag{7.9c}$$

The first thing to notice is that the inflow velocity is reduced drastically. In comparison to the first experiment, q^{in} is almost halved. This leads to the conclusion that a slower inflow velocity is more suited to estimate the exponents of the modified Corey–Brooks model (7.3). Furthermore, the sequences of bacteria and nutrient injections are rearranged. We start the injection of the nutrients at $t = 13.5$ before any bacteria have been injected. Thus, the sandpack is filled with nutrients and the bacteria are able to start growing immediately. Bacteria are injected from $t = 16.5$ until the end of the time interval with three different relatively low concentrations. There is no final water flood for the designed experiment. A reason for this might be to increase the growth inside the sandpack. Nutrients are injected also until the end of the experiment with varying concentrations.

As constrained in Equation (7.8), we end up with 12 measurements for the pressure gradient and the recovered oil both at the same time points

$$t \in \{7,8,11,12,13,15,16,18,21,22,23,24\}.$$

The concentrations of the bacteria, nutrients and metabolites at $x = 5$ are measured 4 times at

$$t \in \{21,22,23,24\},$$

and for the concentrations at $x = 25$ we measure at

$$t \in \{22,23,24\}.$$

The measurement weights at the given time points already fulfill the integrity condition. Thus, no round strategy has to be applied.

In Table 7.3, we show a comparison of the A-criterion and the relative confidence intervals for the first experiment and for both experiments. The third column shows the difference between both values. The A-criterion has been improved by 99 percent. The relative radius for the first parameter, that has been the largest after the first parameter estimation with 9 percent, is reduced significantly to around 2 percent. All other relative radii are smaller than 0.5 percent which implies a significantly enhanced estimation for the parameters.

In Figure 7.5, we illustrate the two dimensional projections of the four dimensional confidence ellipsoid, i.e., the linearized 95%-confidence region, see Theorem 4.3.5. The projections are moved to the origin by

$$\delta\theta_i = \theta_i - \hat{\theta}_i$$

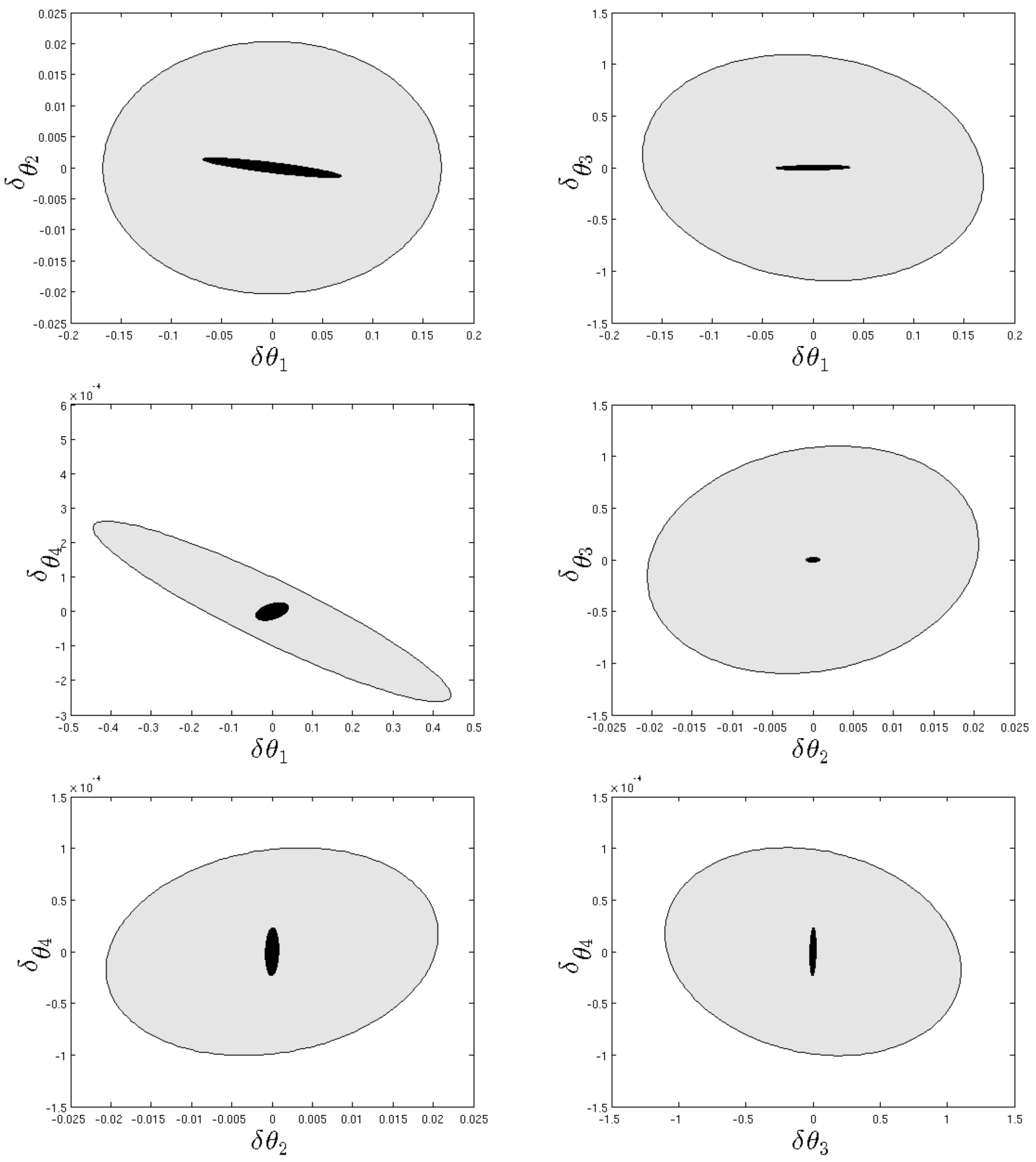


Figure 7.5: Projections of the four dimensional ellipsoid of the linearized 95%-confidence regions for the first (gray) and for the first and the second optimized (black) experiment. Computed at the solution of the first parameter estimation run (7.7)

	first experiment	plus second experiment	error
A-criterion	0.0619034	0.000557845	0.061345555
Θ_1/θ_1	0.0924	0.0210	0.0724
Θ_2/θ_2	0.0058	0.0006	0.0052
Θ_3/θ_3	0.0549	0.0013	0.0536
Θ_4/θ_4	0.0020	0.0003	0.0017

Table 7.3: Comparison of relative radii of the confidence intervals for parameters (7.7).

In the next section, we use the designed experiment to compute a new estimate for the true parameters (7.5).

7.4.3 Second parameter fit

Assume that the experiment designed in Section 7.4.2 has been executed by an experimenter and that corresponding measurement data have been produced. We start the parameter estimation with the results from Section 7.4.1

$$c_{\mathrm{o}}^0 = 2.16745 \quad c_{\mathrm{w}}^0 = 1.59567 \quad \mu_{\mathrm{max}}^0 = 8.29163, \quad K_{\mathrm{S}}^0 = 0.0598582.$$

Table 7.4 shows the progress of the algorithm. We start with the minimal step size and switch to full step afterwards. The residual goes down to 147.534 for 130 measurements. Since we start close to the true parameters, the increment is small from the beginning. After three iteration the algorithm stops in the solution

I	EQ	LS	INCREM	RELAX	RK
1	0.0	$1.53846 \cdot 10^2$	$5.26800 \cdot 10^{-3}$	10^{-1}	4
2	0.0	$1.52648 \cdot 10^2$	$4.25822 \cdot 10^{-3}$	1.0	4
3	0.0	$1.47534 \cdot 10^2$	$2.48207 \cdot 10^{-7}$	1.0	4

Table 7.4: Progress of the second parameter estimation run for both experiments

$$\hat{c}_{\mathrm{o}} = 2.09652 \pm 0.0583926 \quad 2.79\%, \tag{7.10a}$$
$$\hat{c}_{\mathrm{w}} = 1.59911 \pm 0.00679002 \quad 0.42\%, \tag{7.10b}$$
$$\hat{\mu}_{\mathrm{max}} = 8.30356 \pm 0.0114448 \quad 0.14\%, \tag{7.10c}$$
$$\hat{K}_{\mathrm{S}} = 0.0599189 \pm 6.10991 \cdot 10^{-5} \quad 0.10\%. \tag{7.10d}$$

In Figures 7.6 and 7.7, we show the model responses fitted to the measurement data for the first and the second experiment respectively. The order of the subplots is the same as in Figure 7.4. We obtain good fits of the measurement data for both experiments and the model is validated.

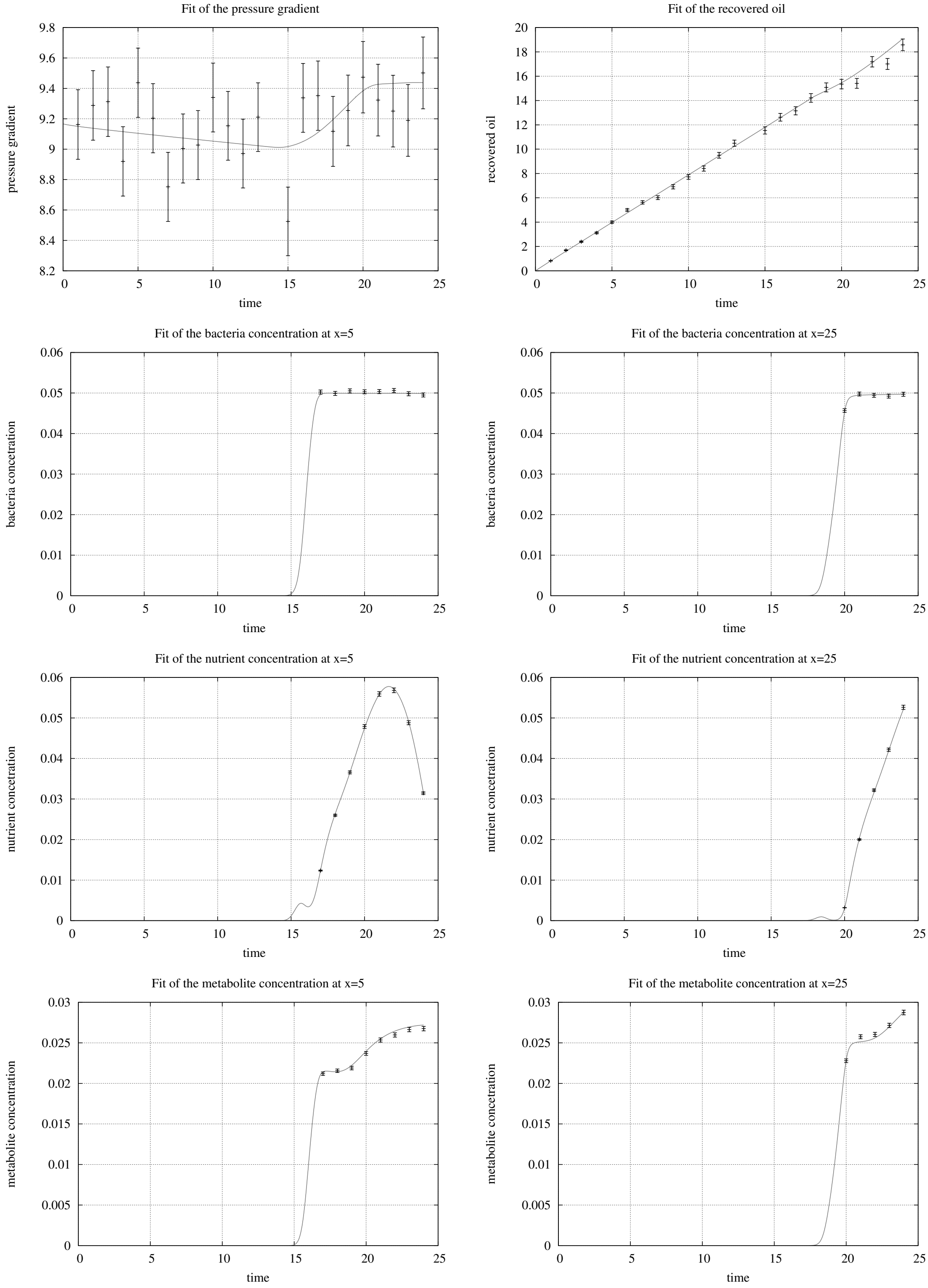


Figure 7.6: Data fit for the first experiment after the second parameter estimation.

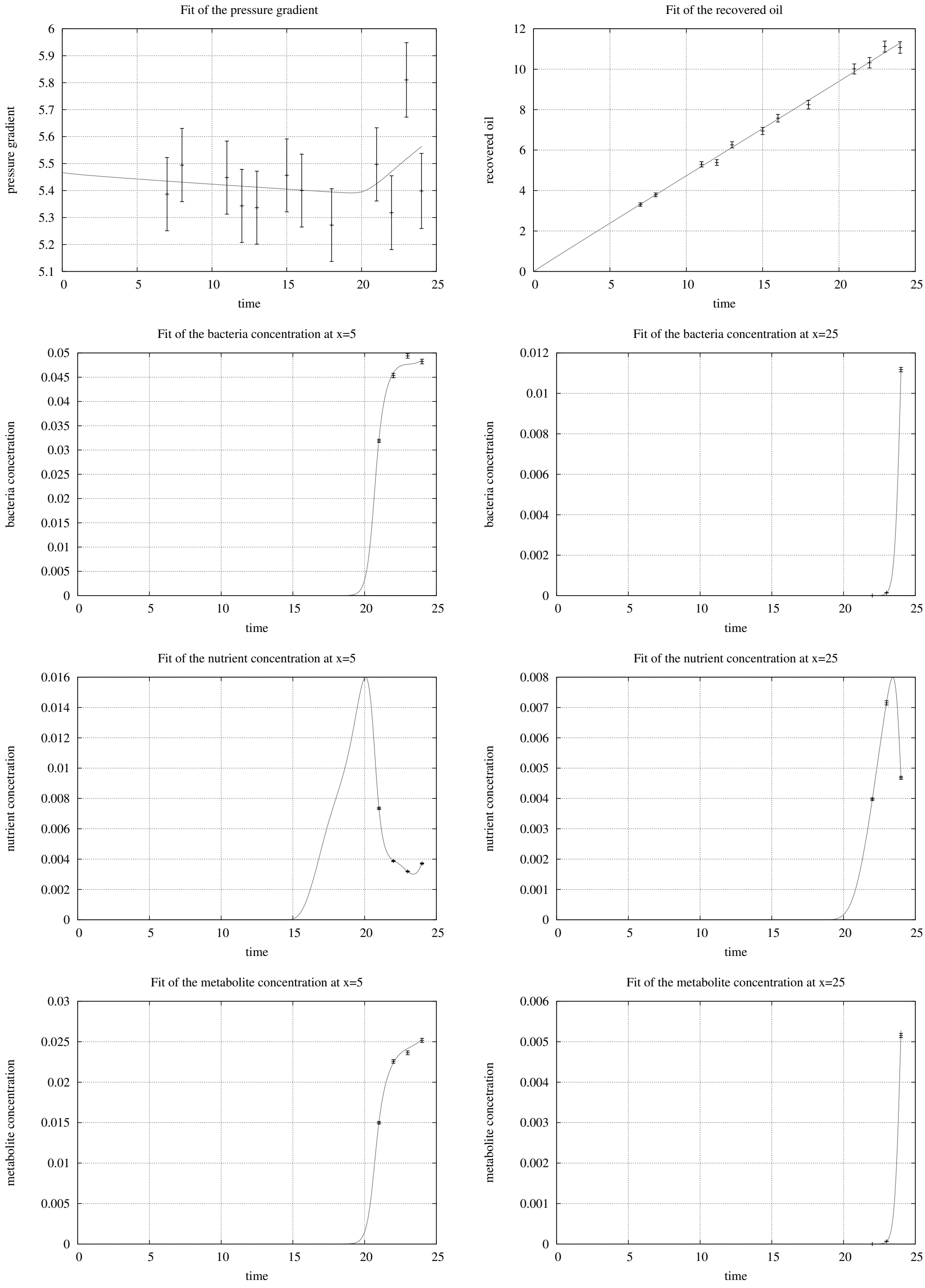


Figure 7.7: Data fit for the second experiment after the second parameter estimation.

8 Conclusions and Outlook

In this thesis, we have developed an efficient and structure exploiting numerical method based on multiple shooting to solve PDE constrained parameter estimation problems. By coupling the evaluation of the Jacobian and the subsequent condensing, we reduce the number of derivatives to the minimal number and, thus, the numerical effort is reduced drastically. We have implemented this approach in the new software package *PAREMERA* and have illustrated the applicability on parameter estimation problems constrained by partial differential equations as well as the efficiency and the reliability of the presented methods by means of two academic application examples.

We have specified a model for microbial enhanced oil recovery (MEOR) and the extended IMPES method, an efficient and robust numerical simulation method for MEOR processes. Using this method, we are able to produce simulation results for numerous combinations of parameters, control variables, control functions and boundary values.

To our knowledge, this is the first time that optimum experimental design (OED) and parameter estimation have been successfully applied to MEOR problems. This has been done by the evaluation of the derivatives with algorithmic differentiation and solving the variational differential equations which has never been done before for this kind of problems.

In line with this work, we implemented a simulation software using the finite element library *deal.ii* where we solve the nominal model and evaluate the first order derivatives with respect to the model parameters, the control variables and the mixed second order derivatives.

Our research induces several tasks for future studies. We conclude this thesis with a list of the major issues:

Extension of the model. We have developed a model that covers the main effects of the microbial enhanced oil recovery process, but it still may be extended by other components. One point would be to neglect the assumption that the gas is solved in the oil phase and to consider a third phase and in this context to assume that the phase densities may vary over time. It would be worthwhile to include further effects of the bacteria such as the reduction of interfacial tension or plugging effects. Additionally, we could drop the homogeneity assumption for the sandpacks. Approaches may be found in Maudgalya [71] and Nielsen [76].

Numerical simulation methods. The IMPES method is widely used in reservoir engineering but because of its restrictive constraints for the time steps it is comparatively slow.

Especially for the evaluation of second order derivatives for OED the computational effort is comparatively large. Thus, an improved parallelization strategy or the development of suitable fully implicit methods is needed. Another possible enhancement is the application of methods for error estimation and mesh adaptivity.

Model validation. To rate the quality of the model, it is necessary to fit the model equation to measurement data obtained for real experiments executed in the laboratory. This should be tested as a next step.

Parameter estimation with inequality constraints. So far it is not possible to consider inequality constraints for the parameter estimation with *PAREMERA*. Therefore, the adaptation of a trust-region method for the determination of the step length for the generalized Gauss–Newton algorithm is needed.

List of Figures

List of Tables

Bibliography

[1] J.H. Abou-Kassem, S.M. Farouq-Ali, and M.R. Islam. *Petroleum Reservoir Simulations*. Elsevier, 2013.

[2] J. Albersmeyer. *Adjoint based Algorithms and numerical Methods for Sensitivity Generation and Optimization of large scale dynamic Systems*. PhD thesis, Ruprecht–Karls–Universität Heidelberg, 2010.

[3] H. Amann. *Ordinary differential equations: an introduction to nonlinear analysis*, volume 13. Walter de Gruyter, 1990.

[4] U. Ascher and M.R. Osborne. A note on solving nonlinear equations and the natural criterion function. *Journal of Optimization Theory and Applications*, 55(1):147–152, 1987.

[5] W. Bangerth, R. Hartmann, and G. Kanschat. deal.ii — a general-purpose object-oriented finite element library. *ACM Transactions on Mathematical Software (TOMS)*, 33(4):24, 2007.

[6] Y. Bard. *Nonlinear Parameter Estimation*. Academic Press, 1974.

[7] I. Bauer. *Numerische Verfahren zur Lösung von Anfangswertaufgaben und zur Generierung von ersten und zweiten Ableitungen mit Anwendungen bei Optimierungsaufgaben in Chemie und Verfahrenstechnik*. PhD thesis, Ruprecht–Karls–Universität Heidelberg, 1999.

[8] I. Bauer, H.G. Bock, and J.P. Schlöder. DAESOL — a BDF-code for the numerical solution of differential algebraic equations. Internal report, IWR, SFB 359, Universität Heidelberg, 1999.

[9] R.A. Berenblyum. *Streamline simulation with capillary effects applied to petroleum engineering problems*. PhD thesis, Technical University of Denmark, Denmark, 2004.

[10] C.H. Bischof, A. Carle, P.M. Khademi, and A. Mauer. The ADIFOR 2.0 System for the Automatic Differentiation of Fortran 77 Programs. Preprint MCS–P481–1194, Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, Ill., 1994. To appear in IEEE Computational Science & Engineering.

[11] H.G. Bock. Numerical treatment of inverse problems in chemical reaction kinetics. In K.H. Ebert, P. Deuflhard, and W. Jäger, editors, *Modelling of Chemical Reaction Systems*, volume 18 of *Springer Series in Chemical Physics*, pages 102–125. Springer, Heidelberg, 1981.

[12] H.G. Bock. Recent advances in parameter identification techniques for ODE. In P. Deuflhard and E. Hairer, editors, *Numerical Treatment of Inverse Problems in Differential and Integral Equations*, pages 95–121. Birkhäuser, Boston, 1983.

[13] H.G. Bock. *Randwertproblemmethoden zur Parameteridentifizierung in Systemen nichtlinearer Differentialgleichungen*, volume 183 of *Bonner Mathematische Schriften*. Universität Bonn, Bonn, 1987.

[14] H.G. Bock, S. Körkel, and J.P. Schlöder. Parameter estimation and optimum experimental design for differential equation models. In H.G. Bock, T. Carraro, W. Jäger, S. Körkel, R. Rannacher, and J.P. Schlöder, editors, *Model Based Parameter Estimation: Theory and Applications*, volume 4 of *Contributions in Mathematical and Computational Sciences*, pages 1–30. Springer Verlag, 2013.

[15] H.G. Bock, E.A. Kostina, and O.I. Kostyukova. Conjugate gradient methods for computing covariance matrices for constrained parameter estimation problems. *SIAM Journal on Matrix Analysis and Application*, 29:626, 2007.

[16] H.G. Bock, E.A. Kostina, and J.P. Schlöder. On the role of natural level functions to achieve global convergence for damped Newton methods. In M.J.D. Powell and S. Scholtes, editors, *System Modelling and Optimization. Methods, Theory and Applications*, pages 51–74. Kluwer, 2000.

[17] H.G. Bock and K.J. Plitt. A multiple shooting algorithm for direct solution of optimal control problems. In *Proceedings of the 9th IFAC World Congress*, pages 242–247, Budapest, 1984. Pergamon Press. Available at http://www.iwr.uni-heidelberg.de/groups/agbock/FILES/Bock1984.pdf.

[18] H.G. Bock, J.P. Schlöder, and V.H. Schulz. Numerik großer Differentiell-Algebraischer Gleichungen — Simulation und Optimierung. In H. Schuler, editor, *Prozeß-Simulation*, chapter 2, pages 35–80. VCH, 1995.

[19] D. Braess. *Finite Elemente: Theorie, schnelle Loser und Anwendungen in der Elastizitätstheorie*. Springer, 2007.

[20] R.H. Brooks and A.T. Corey. Hydraulic properties of porous media. *Hydrology Papers 3*, 1964.

[21] Bundesanstalt für Geowissenschaften und Rohstoffe (2013). Energiestudie 2009. Reserven, Ressourcen und Verfügbarkeit von Energierohstoffen (17). *Hannover*, page 112, 2013.

[22] M.-M. Chang, F. T-H. Chung, R.S. Bryant, H.W. Gao, and T.E. Burchfield. Modeling and laboratory investigation of microbial transport phenomena in porous media. In

SPE Annual Technical Conference and Exhibition. Society of Petroleum Engineers, 1991.

[23] G. Chavent and G. Salzano. A finite-element method for the 1-d water flooding problem with gravity. *Journal of Computational Physics*, 45(3):307–344, 1982.

[24] Z. Chen. *Reservoir Simulation: Mathematical Techniques in Oil Recovery*. CBMS-NSF Regional Conference Series in Applied Mathematics. SIAM, 2007.

[25] K.H. Coats. IMPES stability: the CFL limit. presented at the SPE Reservoir Simulation Symposium, Houston. *Houston, TX.*, 2001.

[26] K.H. Coats et al. IMPES stability: selection of stable timesteps. *SPE Journal*, 8(02):181–187, 2003.

[27] M.Y. Corapcioglu and A. Haridas. Transport and fate of microorganisms in porous media: A theoretical investigation. *Journal of Hydrology*, 72(1):149–169, 1984.

[28] A.T. Corey. The interrelation between gas and oil relative permeabilities. *Producers monthly*, 19(1):38–41, 1954.

[29] R. Courant, K. Friedrichs, and H. Lewy. Über die partiellen Differenzengleichungen der mathematischen Physik. *Mathematische Annalen*, 100(1):32–74, 1928.

[30] J.E. Dennis and J.J. Moré. A characterization of superlinear convergence and its application to quasi-Newton methods. *Mathematics of Computation*, 28(126):549–560, 1974.

[31] S.M. Desouky, M.M. Abdel-Daim, M.H. Sayyouh, and A.S. Dahab. Modelling and laboratory investigation of microbial enhanced oil recovery. *Journal of Petroleum Science and Engineering*, 15(2):309–320, 1996.

[32] P. Deuflhard. *Newton methods for nonlinear problems: affine invariance and adaptive algorithms*, volume 35. Springer, 2011.

[33] P. Deuflhard and M. Weiser. *Adaptive numerical solution of PDEs*. Walter de Gruyter, 2012.

[34] A. Dieses. *Numerical Methods for Optimization Problems in Water Flow and Reactive Solute Transport Processes of Xenobiotics in soils*. PhD thesis, Universität Heidelberg, 2000.

[35] J. Doherty, L. Brebber, and P. Whyte. PEST: Model-independent parameter estimation. *Watermark Computing, Corinda, Australia*, 122, 1994.

[36] J. Douglas Jr., D.W. Peaceman, H.H. Rachford Jr., et al. A method for calculating multi-dimensional immiscible displacement. *Trans. Aime*, 216:297–308, 1959.

[37] F.A.L. Dullien. *Porous Media: Fluid Transport and Pore Structure*. Academic press, 1991.

[38] L.C. Evans. *Partial Differential Equations*, volume 19 of *Graduate Studies in Mathematics*. American Mathematical Society, second edition, 2010.

[39] R. Fletcher. *Practical Methods of Optimization*. John Wiley & Sons, New York, 2nd edition, 1987. ISBN 0-471-49463-1 (paperback).

[40] C.H. Gao and A. Zekri. Applications of microbial-enhanced oil recovery technology in the past decade. *Energy Sources, Part A: Recovery, Utilization, and Environmental Effects*, 33(10):972–989, 2011.

[41] C. Geiger and C. Kanzow. *Theorie und Numerik restringierter Optimierungsaufgaben*. Springer–Lehrbuch Masterclass. Springer Verlag, Berlin Heidelberg New York, 2002.

[42] C. Gerthsen, H.O. Kneser, and H. Vogel. *Physik*. Springer Verlag, 1989.

[43] P.E. Gill, W. Murray, and M.A. Saunders. SNOPT: An SQP algorithm for large-scale constrained optimization. *SIAM Review*, 47(1):99–131, 2005.

[44] P.E. Gill, W. Murray, and M.A. Saunders. SNOPT 7 user's guide: software for large-scale nonlinear programming. 2007.

[45] G.H. Golub and C.F. van Loan. *Matrix computations*, volume 3. JHU Press, 2012.

[46] A. Griewank. *Evaluating Derivatives, Principles and Techniques of Algorithmic Differentiation*. Number 19 in Frontiers in Applied Mathematics. SIAM, Philadelphia, 2000.

[47] C. Grossmann and H.-G. Roos. *Numerical Treatment of Partial Differential Equations*. Springer, 2007.

[48] E. Hairer and G. Wanner. *Solving Ordinary Differential Equations II. Stiff and Differential-Algebraic Problems*. Springer Series in Computational Mathematics. Springer, 2nd edition, 1996.

[49] S.B. Hazra and V. Schulz. Numerical parameter identification in multiphase flow through porous media. *Computing and Visualization in Science*, 5(2):107–113, 2002.

[50] J.S. Hesthaven and T. Warburton. *Nodal Discontinuous Galerkin Methods: algorithms, analysis, and applications*, volume 54. Springer, 2007.

[51] M. Hinze, R. Pinnau, M. Ulbrich, and S. Ulbrich. *Optimization with PDE Constraints*. Springer, New York, 2009.

[52] S. Hoops, S. Sahle, R. Gauges, C. Lee, J. Pahle, N. Simus, M. Singhal, L. Xu, P. Mendes, and U. Kummer. COPASI — a complex pathway simulator. *Bioinformatics*, 22(24):3067–3074, 2006.

[53] H. Hoteit and A. Firoozabadi. Numerical modeling of two-phase flow in heterogeneous permeable media with different capillarity pressures. *Advances in Water Resources*, 31(1):56–73, 2008.

[54] P. Hron, D. Jost, P. Bastian, C. Gallert, J. Winter, and O. Ippischa. Application of reactive transport modelling to growth and transport of microorganisms in the capillary fringe. *Vadose Zone Journal*, submitted 2014.

[55] R. Kircheis and S. Körkel. Parameter estimation for high-dimensional pde models using a reduced approach. In *Multiple Shooting and Time Domain Decomposition Methods*, Contributions in Mathematical and Computational Sciences. Springer, submitted 2014.

[56] S. Kirkpatrick, C.D. Gelatt, M.P. Vecchi, et al. Optimization by simmulated annealing. *science*, 220(4598):671–680, 1983.

[57] F. Kögler. The Impact of in-situ Microorganisms on Oil Recovery Measured and Evaluated in Sandpack Experiments. Master's thesis, Friedrich–Schiller–Universität Jena, Jena, 2013.

[58] S. Körkel. *Numerische Methoden für Optimale Versuchsplanungsprobleme bei nichtlinearen DAE-Modellen*. PhD thesis, Ruprecht–Karls–Universität Heidelberg, Heidelberg, 2002.

[59] S. Körkel, I. Bauer, H.G. Bock, and J.P. Schlöder. A sequential approach for nonlinear optimum experimental design in DAE systems. In F. Keil, W. Mackens, H. Voss, and J. Werther, editors, *Scientific Computing in Chemical Engineering II*, volume 2, pages 338–345, Berlin, Heidelberg, 1999. Springer-Verlag.

[60] J. Kou and S. Sun. On iterative IMPES formulation for two phase flow with capillarity in heterogeneous porous media. *International Journal of Numerical Analysis and Modeling. Series B*, 1(1):20–40, 2010.

[61] D.B. Leineweber. *Efficient reduced SQP methods for the optimization of chemical processes described by large sparse DAE models*, volume 613 of *Fortschritt-Berichte VDI Reihe 3, Verfahrenstechnik*. VDI Verlag, Düsseldorf, 1999.

[62] S.M. Lenz. *Impulsive Hybrid Discrete-Continuous Delay Differential Equations*. PhD thesis, Ruprecht–Karls–Universität Heidelberg, Heidelberg, 2014.

[63] K. Levenberg. A method for the solution of certain problems in least squares. *Quarterly of applied mathematics*, 2:164–168, 1944.

[64] R.J. LeVeque. *Finite volume methods for hyperbolic problems*, volume 31. Cambridge university press, 2002.

[65] Y. Li and W. Bangerth. The step-21 tutorial program. Website, 2014. available online `http://www.dealii.org/7.3.0/doxygen/deal.II/step_21.html`; accessed 17-May-2014.

[66] K.-A. Lie, S. Krogstad, I.S. Ligaarden, J.R. Natvig, H.M. Nilsen, and B. Skaflestad. Open-source MATLAB implementation of consistent discretisations on complex grids. *Computational Geosciences*, 16(2):297–322, 2012.

[67] T. Lohmann, H.G. Bock, and J.P. Schlöder. Numerical methods for parameter estimation and optimal experimental design in chemical reaction systems. *Industrial and Engineering Chemistry Research*, 31:54–57, 1992.

[68] A. Lohne. User's manual for BUGSIM — an MEOR simulator (Version 1.2). Report 2008/084, International Research Institute of Stavanger, Stavanger, Norway, 2008.

[69] R.C. MacDonald and K.H. Coats. Methods for numerical simulation of water and gas coning. *Trans. SPE AIME*, 249:425–436, 1970.

[70] D.W. Marquardt. An algorithm for least-squares estimation of nonlinear parameters. *Journal of the Society for Industrial & Applied Mathematics*, 11(2):431–441, 1963.

[71] S. Maudgalya. *Experimental and numerical simulation study of microbial enhanced oil recovery using bio-surfactants*. PhD thesis, University of Oklahoma, Norman, Oklahoma, 2005.

[72] S. Maudgalya, R.M. Knapp, M. McInerney, et al. Microbially enhanced oil recovery technologies a review of the past present and future. In *Production and Operations Symposium*. Society of Petroleum Engineers, 2007.

[73] G.P. McCormick. *Nonlinear Programming*. John Wiley & Sons, New York, 1983.

[74] J. Monod. The wrowth of bacterial cultures. *Annual Reviews in Microbiology*, 3(1):371–394, 1949.

[75] J. Naar, J.H. Henderson, et al. An imbibition model — Its application to flow behavior and the prediction of oil recovery. *Society of Petroleum Engineers Journal*, 1(02):61–70, 1961.

[76] S.M. Nielsen. *Microbial Enhanced Oil Recovery — Advanced Reservoir Simulation*. PhD thesis, Technical University of Denmark, Denmark, 2010.

[77] J. Nocedal and S.J. Wright. *Numerical Optimization*. Springer Verlag, Berlin Heidelberg New York, 2nd edition, 2006. ISBN 0-387-30303-0 (hardcover).

[78] OpenMP Architecture Review Board. OpenMP application program interface version 3.0, May 2008.

[79] H. Pan and H. Cao. User Manual for General Purpose Research Simulator (GPRS), 2002.

[80] F. Pukelsheim. *Optimal Design of Experiments*. Classics in Applied Mathematics 50. SIAM, 2006. ISBN 978-0-898716-04-7.

[81] H.J. Quadbeck-Seeger. *Der Wechsel allein ist das Beständige*. Wiley VCH, Weinheim, 2007.

[82] W.H. Reed and T.R. Hill. Triangular mesh methods for the neutron transport equation. *Los Alamos Report LA-UR-73-479*, 1973.

[83] W.E. Schiesser. *The numerical method of lines*. Academic Press, 1991.

[84] K. Schittkowski. Solving constrained nonlinear least squares problems by a general purpose sqp-method. In *Trends in Mathematical Optimization*, pages 295–309. Springer, 1988.

[85] J.P. Schlöder. *Numerische Methoden zur Behandlung hochdimensionaler Aufgaben der Parameteridentifizierung*. Dissertation, Hohe Mathematisch-Naturwissenschaftliche Fakultät der Rheinischen Friedrich-Wilhelms-Universität zu Bonn, 1987.

[86] J.P. Schlöder. personal communication, 2011.

[87] G.A.F. Seber and C.J. Wild. *Nonlinear Regression*. John Wiley & Sons, New York, 1989.

[88] J.W. Sheldon, B. Zondek, and W.T. Cardwell. One-dimensional, incompressible, non-capillary, two-phase fluid flow in a porous medium. *Trans. SPE AIME*, 216:290–296, 1959.

[89] J. Stoer and R. Bulirsch. *Numerische Mathematik 2*. Springer Verlag, Berlin Heidelberg New York, 4th edition, 2000. ISBN 3-540-67644-9.

[90] H.L. Stone et al. Estimation of three-phase relative permeability and residual oil data. *Journal of Canadian Petroleum Technology*, 12(04), 1973.

[91] H.L. Stone and A.O. Garder Jr. Analysis of gas-cap or dissolved-gas reservoirs. *Trans. SPE AIME*, 222:92–104, 1961.

[92] R. Telgmann. *Computerunterstütztes Modellieren*. PhD thesis, FU Berlin, 2008.

[93] G.W. Thomas and D.H. Thurnau. Reservoir simulation using an adaptive implicit method. *Society of Petroleum Engineers Journal*, 23(5):759–768, 1983.

[94] F. Tröltzsch. *Optimale Steuerung partieller Differentialgleichungen: Theorie, Verfahren und Anwendungen*. Vieweg+Teubner Verlag, Wiesbaden, 2nd edition, 2009.

[95] M. Ulbrich and S. Ulbrich. *Nichtlineare Optimierung*. Birkhüser, Basel, 2012. ISBN 978-3-03-460654-7 (hardcover).

[96] P.H. Valvatne and M.J. Blunt. Predictive pore-scale modeling of two-phase flow in mixed wet media. *Water Resources Research*, 40(7), 2004.

[97] S.F. Walter. *Structured Higher-Order Algorithmic Differentiation in the Forward and Reverse Mode with Application in Optimum Experimental Design*. PhD thesis, Humboldt–Universität Berlin, Berlin, 2011.

[98] C.K.F. Weiler. *Optimum Experimental Design for the Identification of Gaussian Disorder Mobility Parameters in Charge Transport Models of Organic Semiconductors*. PhD thesis, Ruprecht–Karls–Universität Heidelberg, Heidelberg, 2014.

[99] J.A. Wheeler and M. Peszynska. IPARS: Integrated Parallel Reservoir Simulator. *Center for Subsurface Modeling, University of Texas at Austin*.

[100] M.F. Wheeler, S. Sun, and S.G. Thomas. Modeling of flow and reactive transport in IPARS. *Groundwater Reactive Transport Models*, page 42, 2012.

[101] ”Wikipedia”. Wrigley Field — Wikipedia, the free encyclopedia, 2004. Online; accessed 06-October-2014.

[102] M. Wulkow. Computer Aided Modeling of Polymer Reaction Engineering—The Status of Predici, I-Simulation. *Macromolecular Reaction Engineering*, 2(6):461–494, 2008.