TU Delft

Systems theory for reservoir management

Lecture notes for course AES1490

Jan-Dirk Jansen

Version 6g, July 2013

These notes are still under development and the text is therefore incomplete. If you find errors, please let me know, JDJ.

Title:	Systems theory for reservoir management
Version:	6g (incomplete)
Date:	July 2013
Type of report:	Lecture notes
Course code:	AES1490 (System analysis)
Author:	Jan-Dirk Jansen
Postal address:	Section Petroleum Engineering Department of Geoscience & Engineering Faculty of Civil Engineering and Geosciences Delft University of Technology P.O. Box 5048 2600 GA Delft The Netherlands
E-mail:	j.d.jansen@tudelft.nl

Chapters 2, 3and 4 have been published as:

Jansen, J.D., 2013: *A systems description of flow through porous media*. SpringerBriefs in Earth Sciences, Springer. ISBN: 978-3-319-00259-0 (print), 978-3-319-00260-6 (online). DOI: 10.1007/978-3-319-00260-6.

Contents

Preface

The elective course AES1490 *System analysis* aims to

- Provide an introduction to the use of system analysis techniques for reservoir management.
- Provide the background knowledge required to perform MSc thesis work in the area of *closed-loop reservoir management*, also known as *smart fields*.

In this course we describe concepts from measurement and control theory as applied to the flow of fluids in oil or gas reservoirs. The course will be given for the eight time this year (2011), and the format will to a large extent be developed during the course period depending on the needs, wishes and capacities of the students enrolled. The course material exists of journal and conference papers and these lecture notes, which are still under development. To successfully complete this course you are supposed to start with at least a basic knowledge of mathematics and reservoir engineering. The mathematical prerequisites concern differential equations, linear algebra, numerical analysis and some statistics. A part of this material will be recapitulated in these lecture notes, but if you are unfamiliar with the basics it will be necessary to refer to a good textbook. Some textbooks are mentioned in these notes, but their choice is based on my own, somewhat arbitrary experience, and many other good texts are available. For the reservoir engineering background it is expected that you have successfully completed the courses AES3110/1320 (Rock-fluid interaction) and AES1340/1350 (Reservoir engineering and simulation).

1 Introduction

1.1 Systems theory

System(s) theory, also known as *system(s) analysis*, is a branch of applied mathematics focused on the modeling and understanding of *dynamical systems*, and in particular the measurement and control aspects. For the purpose of this course we define a dynamical system as a set of physical *components* that can interact with each other through the exchange of *mass, momentum* and *energy*. In particular we will consider numerical reservoir models where the components are discrete *grid blocks* that result from discretizing the governing *partial differential equations* in the spatial dimensions. The resulting set of *ordinary differential equations* can be interpreted as a *continuous-time* dynamical system. If we also apply a discretization in time, the resulting set of *ordinary difference equations* can be interpreted as a *discrete-time* dynamical system. For an introduction to dynamic systems we refer to e.g. Luenberger (1979)[†]. In particular the theory for linear dynamical systems has been developed to great depth and a large number of text books is available of which we mention Olsder and van der Woude (2004) and Antsaklis and Michel (2007) for introductions, and Kailath (1980), DeCarlo (1989) or Antsaklis and Michel (2006) for more in-depth treatments. For a further study of the measurement and control aspects there is also a vast amount of text books available, many of which, however, heavily rely on frequency domain techniques which are of less relevance to our application. A good text book at an elementary level which emphasizes state-space techniques is Friedland (1986). More advanced treatments with a focus on engineering applications are given by e.g. Glad and Ljung (2000) or Skogestad and Postlewaite (2005).

1.2 Reservoir management

1.2.1 Spatial and time domains

Within the Exploration and Production (E&P) life cycle various processes can be distinguished, with their own spatial and time domains. Figure 1.1 displays a rough subdivision in three major processes:

1. *Production management*, which can be further subdivided in e.g. *real-time operations* and *production optimization.* Real-time operations are focussed on short-term control, e.g. automatic gas-coning control using wellhead chokes on a timescale from seconds to hours. Production optimization addresses aspects on a somewhat longer timescale, say days or weeks. A classic example is the optimal distribution of lift gas[‡] over a group of wells or an entire field.
2. *Reservoir management*, which also can be subdivided, e.g. in *reservoir surveillance* and *field (re-) development planning*. Reservoir surveillance has a typical time horizon of weeks to months and is concerned with verifying actual production performance against predictions, and subsequent adjustments of the field management through e.g. production logging, pattern management, recompletion or side-tracking. Field (re-)development planning is a design activity focused on optimizing the

[†] The format for references to books or papers is: Author name (year), e.g. Luenberger (1979). Further details are given in the alphabetical list of references on page 10.

[‡] *Lift gas* is gas pumped from surface and injected at the bottom of a well to reduce the weight of the fluid column in the well with the aim to increase the inflow from the reservoir.

depletion strategy over the entire reservoir life, i.e. a time scale of many years to decades.

3. *Portfolio management*, which typically involves strategic decision making and other business activities to ensure a balanced development of many oil and gas fields over decades. The focus for commercial oil companies is primarily on sustained profitability, while for national oil companies and governments the prime focus is on responsible long-term development of national resources.

The portfolio management process provides *objectives* and *constraints* for the various elements of the reservoir management process in the domain below it, just as a reservoir management process provides objectives and constraints for production optimization and production operations. Oppositely, the production management process provides the *historic data* and *forecasts* that are required in the domain above it to perform proper reservoir management, just as the data and forecasts from reservoir management are essential for good portfolio management. In this course we will mainly focus on techniques that are of use in the reservoir management domain, in particular those that are based on the use of numerical models, see Figure 1.2.

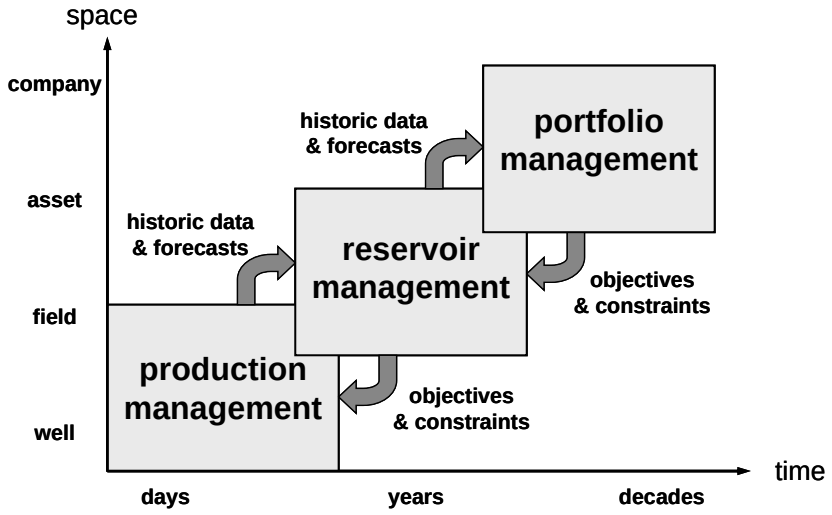


Figure 1.1: E&P process domains.

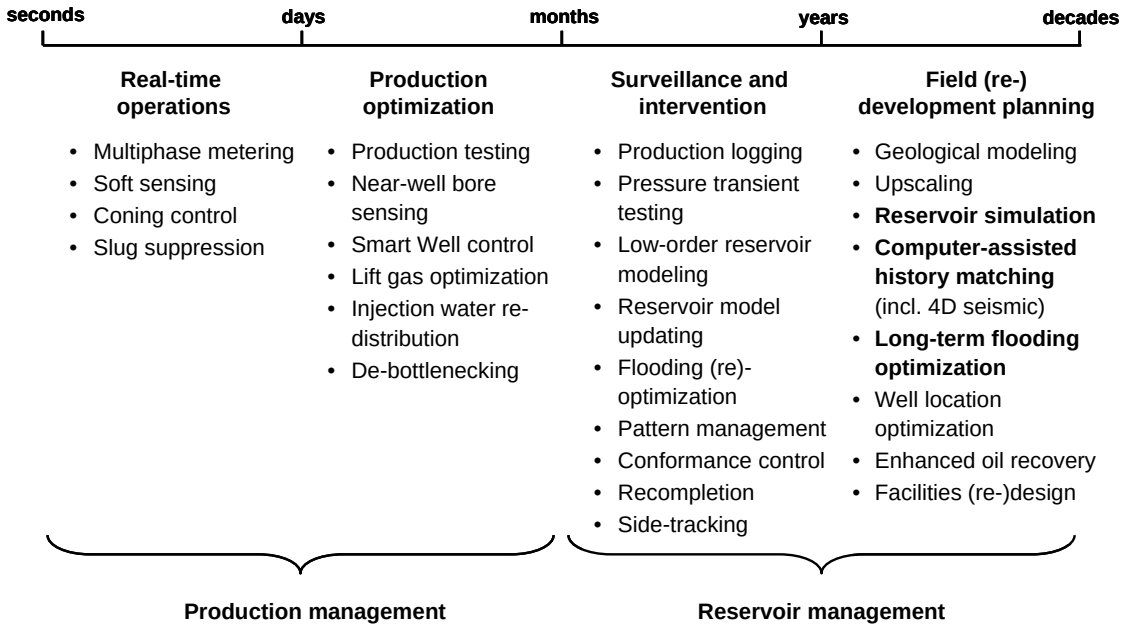


Figure 1.2: Activities and associated time scales in production and reservoir management. Activities indicated in bold are addressed in the current version of these lecture notes.

1.2.2 Open-loop and closed-loop control

Figure 1.3 depicts reservoir management as a model-based controlled process (Jansen et al., 2005, 2008). The *system*, at the top of the figure, comprises of one or more reservoirs, wells and facilities. Generally, the *system boundaries* can be specified accurately for the wells and the surface facilities, but are much more uncertain for the reservoir of which the geometry is usually deduced from seismics with a limited resolution. Also the *parameters* of the system are known to varying degrees: the fluid properties can usually be determined quite well, but the permeabilities and porosities of the reservoir are only really known at the wells. Moreover, the dynamic variables, also known as the *state variables* of the system, (i.e. the pressures and saturations in the reservoirs, the pressures and phase rates in the wells, etc.) are only known to a limited extent from the measured *output* of various *sensors* at surface or down hole, and from more indirect measurements such as time-lapse seismics. Not only are measurements scarce, but as indicated in the upper-right corner of the figure, they also contain *noise*. Also the *input* to the system is only known to a limited extent (i.e. water injection rates or gas lift rates may be roughly known, but aquifer support may be a major unknown). The unknown inputs can also be interpreted as noise, as indicated at the top-left corner of the figure. At the field development stage of a reservoir, i.e. before production starts, no input or output data are available and the proposed process *control*, i.e. the field development plan (FDP), must therefore be based on static and dynamic reservoir models built on data from outcrop studies, seismics, well tests, etc. The process control is thus performed without feedback, or in other words in *open-loop*. During the producing life of the reservoir production data become available which are then used to manage day-to-day operations. E.g. production test data and wellhead pressure readings are used to control oil and gas production to meet daily targets, although usually without any form of formal system model. In addition, reservoir surveillance is performed in the form of monitoring trends in production and injection rates and well head pressures, and deployment of production logs on wire line to estimate the influx from different reservoir zones. The surveillance data are used

to guide reservoir management decisions on e.g. well interventions or even infill drilling. However, usually these data are not used in a systematic fashion to update the reservoir model, and typically a reservoir model is gradually getting out of date. After a period of several years it is then decided to perform a field redevelopment. At that moment the model is updated or, quite often, rebuilt from scratch using all data available. At best we can therefore classify this as *closed-loop* reservoir management in a *batch* mode.

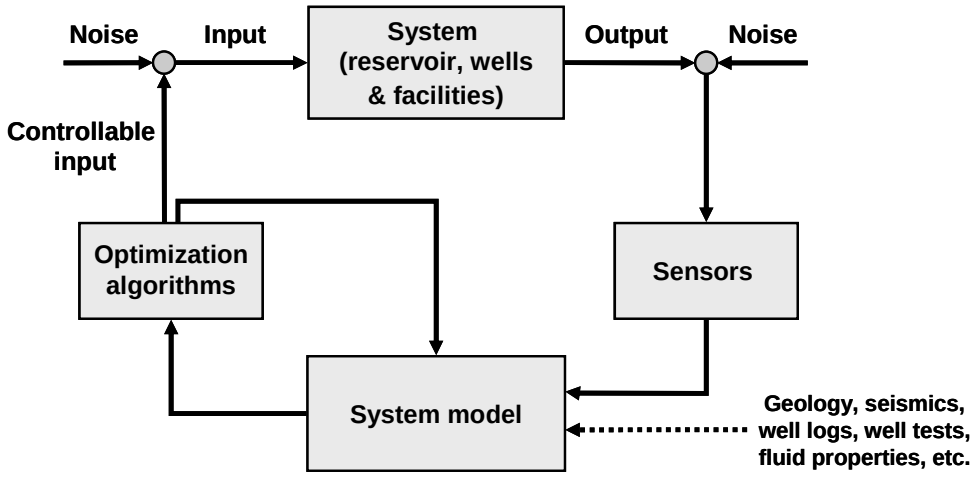


Figure 1.3: Reservoir management represented as a model-based closed-loop controlled process.

If we would like to make use of measured data to change reservoir management to a *near-continuous closed-loop* controlled process, it is useful to first address the different ways that data are used in classical measurement and control theory. In many industries where process control plays a role, the properties of the system are relatively well known, but the state variables, i.e. the dynamic process variables, cannot be measured directly. E.g. the stress in a robot arm may be difficult to measure directly. In that case the use of a system model together with measured inputs and outputs can be used to reconstruct the state. E.g. measured accelerations at the tip of the robot arm together with a dynamic model of the arm can be used to estimate the internal stresses. The motions of the arm can then be controlled such that the stresses do not exceed a certain level. In that case the system model, with known parameters, is used as an *observer* of the states that are inaccessible to direct measurements, to allow closed-loop control of system. A different situation occurs if the system parameters are not well known. This prompts another use of measured inputs and outputs, namely to *estimate* the value of the unknown system parameters, or to *update* their values over time. In it's most extreme form the system model is nothing more than a mathematical relationship between the inputs and the outputs. Such *identified* system models are often called *data-driven* or *black-box* models, as opposed to *white-box* models which are primarily based on known relationships such as conservation laws. Also white-box models, however, often contain a number of parameters that need to be 'tuned', i.e. estimated, using measured input and output data, and are therefore sometimes referred to as *gray-box* models.

1.2.3 Conventional reservoir management

An example of closed-loop reservoir management using a black-box model is the use of decline curves to predict future well and reservoir performance. Also material balance models could be classified as black-box, although the presence of some physics in the form of a mass

balance could arguably make them grayish. Finite difference or finite-element reservoir models, based on physics such as conservation of mass, Darcy's law and vapor-liquid equilibrium, certainly classify as white-box models. Because of geological uncertainties they are usually only a very crude approximation of reality, and the model parameters, such as permeabilities and porosities, are only known with a large degree of uncertainty. Therefore the predictive value of such reservoir models is limited and tends to deteriorate over time. This sometimes leads to attempts to *history match* the model by adapting model parameters such that simulated results approach measured production data. Traditional history matching, however, suffers from a number of drawbacks:

1. It is usually only performed on a campaign basis, typically after periods of years.
2. The matching techniques are often ad-hoc and involve manual adjustment of model parameters, instead of systematic parameter updating.
3. Uncertainties in the state variables, model parameters and measured data are usually not explicitly taken into account.
4. The resulting history-matched models often violate essential geological constraints.
5. Worst of all, the updated model may reproduce the production data almost perfectly but have no predictive capacity because it has been over-fitted by adjusting a large number of unknown parameters using a much smaller number of measurements.

Manual history matching is of course a form of parameter updating meant to improve the predictive capacity of the model, and management of a reservoir based on manually history-matched models is indeed a form of closed-loop reservoir management. True closed-loop reservoir management would require a shift from campaign-based ad-hoc history matching to a much more frequent systematic updating of system models, based on data from different sources, while honoring geological constraints and the various sources of uncertainty. A considerable amount of research has been performed over the past decades, especially in academia, to develop 'computer-assisted' history matching methods that address the shortcomings of traditional history matching. Also, several research groups have been working on advanced optimization methods for reservoir drainage. Much of this work has been published, but often using completely different terminology and notation.

1.3 Closed-loop reservoir management

1.3.1 Drivers

The concept of 'closed-loop' or 'real-time' reservoir management and production optimization has been described in different forms before see e.g. Chierici (1992), Nyhavn et al. (2000), Rossi *et al.* (2000), Nygård et al. (2001). Moreover, during the past decade the concept has been receiving considerable attention as part of initiatives with names such as *e-fields* (Litvak et al., 2002), *smart fields* (Kapteijn and Muessig, 2003; Potters and Kapteijn, 2005), *self-learning reservoir management* (Saputelli et al., 2005), *real-time asset management* (Unneland and Hauser, 2005), or *integrated operations*. Most of these recent initiatives primarily address short term production optimization issues, but usually foresee the need for some form of closed-loop reservoir management as a next step. As opposed, earlier references (e.g. Chierici et al., 1992) are often more focussed on field (re-)development planning, i.e. on improving reservoir management from a geosciences perspective. We will consider closed-loop reservoir management also from a reservoir-surveillance point of view, in addition to the geosciences perspective. The underlying hypothesis is that

"It will be possible to significantly increase life-cycle value by changing reservoir management from a periodic to a near-continuous model-based controlled activity."

We stress that, in our view, "closed-loop" does not imply removal of human judgment from the loop. The use of model-based optimization and data assimilation techniques should result in a reduction of time spent on repetitive and tedious human activities and thus in more time to be spent on judging results and taking decisions.

1.3.2 External focus

The justification for our research is partly economic necessity and partly technological progress. Advances in sensor technology have resulted in the possibility to generate vast amounts of data at low costs, and computing power is still steadily increasing. At the same time new reserves are increasingly more difficult to find and increasingly smaller than in the past, making new, more profitable, reservoir management techniques a necessity. Smart fields technology may increase ultimate recovery of a field, accelerate production or reduce the production of unwanted by-products, in particular water. Moreover smart fields technology may reduce the amount of development wells, thus reducing land take, cuttings and waste drilling fluids. At present, the use of smart field technology to increase ultimate recovery is mainly focused on secondary recovery, in particular water flooding, but the concept of closed-loop reservoir management may also be applicable to improve tertiary recovery e.g. water-alternating-gas (WAG) flooding, polymer flooding or thermal recovery. In the long term, the ability to measure and control subsurface flow may be one of the enablers for controlled subsurface hydrocarbon conversion. In the utopian form such a *down-hole factory* would convert hydrocarbons in-situ into hydrogen while leaving all unwanted by-products such as CO_2 or H_2S in the ground. Naturally, less spectacular but much more realistic conversion processes have to be addressed first and are indeed subject of ongoing exploratory research in various research groups around the world.Improvement of the state of the art in the E&P industry will require techniques from other industries. A key source of inspiration is systems theory as used in the process industry. Also known as measurement and control theory, it offers a wealth of mathematical techniques to address multivariate optimization and control problems of both linear and nonlinear systems, including a systematic treatment of uncertainties. The only drawback is that most of the theory is focused on relatively simple, low-order systems, i.e. systems with a small number of state variables as compared to the number of variables (grid block pressures and saturations) used in numerical reservoir simulation. A second important source of inspiration from outside the E&P industry are *data assimilation* techniques as used in meteorology or oceanography. These involve the rapid updating of large scale numerical models as used for weather and climate forecasting based on data from various sources such as satellite imaging, weather balloons, buoys or ground stations. Designed for very high-order systems (containing up to millions of state variables) these techniques appear to be very relevant to reservoir simulation, and indeed they have already successfully been applied in the area of groundwater flow. The main challenge is therefore to adapt them to the particular aspects of multi-phase reservoir flow, such as the occurrence of near shock-like behavior, and the large uncertainty in model parameter values.

1.3.3 Process elements

Figure 1.4 is a more detailed version of Figure 1.3. Instead of a single system model, the figure displays an *ensemble* (group) of *high-order* (detailed) models and another ensemble of *low-order* (simplified) models. Typically the high-order ensemble consists of equiprobable

realizations of *static* (geological) reservoir models with different parameters and possibly different geometries. The use of an ensemble rather than a single model is meant to capture the uncertainty about the true system. The ensemble of low-order models typically consists of *dynamic* (flow simulation) reservoir models, consisting of *upscaled* versions of the static models. However, it is also possible to interpret the figure such that the high-order models are dynamic reservoir simulation models, whereas the low-order models are *proxies*, i.e. simplified representations, that can be either white-box models with limited physics or even black-box models that just describe the input-output relations. These could be of use for e.g. fast simulations during reservoir surveillance.

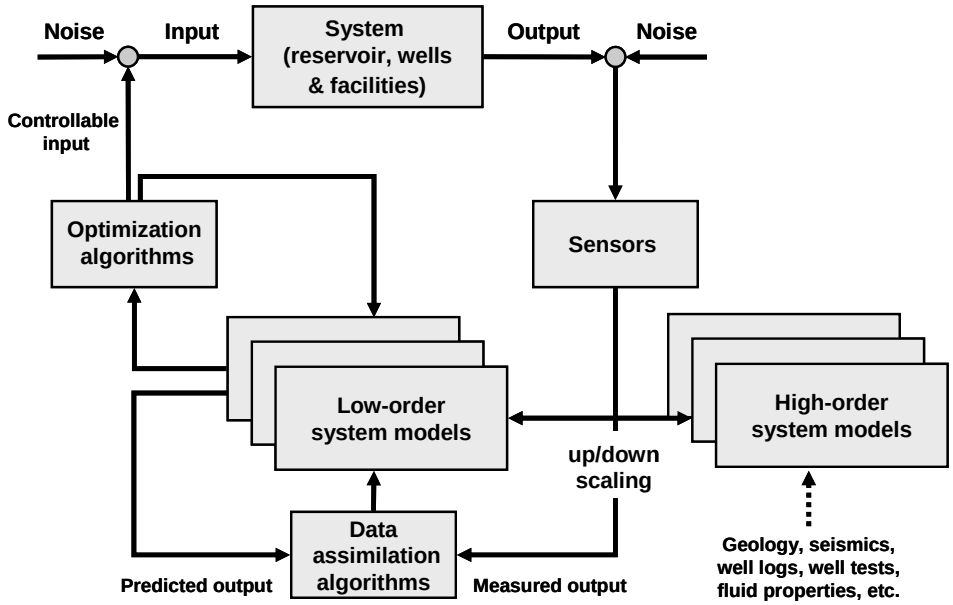


Figure 1.4: Elements of the closed-loop reservoir management process addressed in this course.

Figure 1.4 also displays the various elements of closed-loop reservoir management:

1. *Optimization.* This involves optimization of an *objective function* while honoring *constraints* that could results from e.g. physical, environmental or economical requirements. Often the objective is to maximize the *net present value* (NPV), i.e. the cumulative discounted cash flow, over the producing life of a reservoir. The optimization may concern a given configuration, e.g. optimizing the injection and production rates in smart well segments, or a free configuration, e.g. determining the optimal position of sidetracks or infill wells. Optimization is a widely developed topic in applied mathematics, and many techniques have been introduced in the reservoir engineering literature, often to disappear after a few years and then to reappear again somewhat later. However, in practice little use is made of mathematical optimization techniques. Most reservoir drainage optimization is based on a trial-and-error design process at the field (re-)development stage and a pragmatic infill drilling program during the reservoir's producing life. The major reasons are the difficulty to quantify geological concepts, and the large uncertainty in reservoir properties which strongly reduce the value of optimization based on a single given model. However, in combination with improved model updating techniques, as will be discussed below, the scope for numerical optimization increases. Promising techniques are *adjoint-*

based methods, and moving-horizon *model-predictive control* techniques. Furthermore, a lot of work is done in the area of 'non-classical' optimization methods such as genetic algorithms or simulated annealing.

2. *Data assimilation*. The bottom of the figure reflects the *updating* of system model parameters through assimilation of data from different sources (e.g. production sensors, time-lapse seismics, passive seismics, remote sensing). The most widely used technique for data assimilation ('computer-assisted history matching') considers the evolution of states over a certain time period. The aim is then to minimize an objective function defined in terms of the differences between the simulated and the measured states over the period. This transforms the updating problem to an optimization problem for which, as discussed above, many mathematical techniques are available. For systems with a large number of state variables the most popular optimization methods are those using an *adjoint* set of equations. An alternative data-assimilation method, *ensemble Kalman filtering,* has been developed over the past decade in oceanography, and can potentially be applied to updating of reservoir models. Ordinary Kalman filtering was developed in the 1960s for tracking of flying objects with radar, but is only applicable to linear systems. However, a recent development using ensembles of models allows application to large-scale nonlinear systems. The first applications of the EnKF were in oceanography and meterology, but the method has rapidly become a very popular also for reservoir history matching.
3. *Up/down scaling.* As discussed above, Figure 1.4 displays ensembles of system models with different levels of complexity. An important reason for the use of low-order models is computational efficiency: the simulation time of detailed models is usually too high to use them for optimization or data assimilation purposes which typically require tens to hundreds of system response simulations. However, from a systems-theoretical point of view there is a much more important reason to use low-order models: the level of detail in system models should be adapted to the available information on the one hand (*observability*; to what extent can a state variable be reconstructed from the outputs?) and the extent of control on the other hand (*controllability*; to what extent can a state variable be influenced by the inputs?). Various model reduction techniques have been developed over the past decades for low-order process control applications. To what extent these techniques can be applied to high-order nonlinear reservoir models is a matter of ongoing research. A promising technique appears to be *Proper Orthogonal Decomposition* (POD) which makes use of *snapshots* obtained during simulation of a high-order model and which is then applicable to nonlinear models. A completely different approach to arrive at a low-order model is to try to identify a black-box model directly from input-output measurements, without first building a high-order model based on physics and geology.

1.3.4 Hardware aspects

An important reason to start the development of a system-theoretical approach to reservoir management was the advent of so-called *smart wells* during the late 1990s. Smart wells are equipped with measurement devices and *inflow control valves* (ICVs) to control inflow from different parts of the reservoir; see Figure 1.5. In first instance these wells were used to merely replace rig intervention by a remotely-controlled operation; the classic example is the

replacement of closing perforations in a watered-out zone with cement and reperforating in an oil-bearing zone, by closing and opening of an ICV respectively.

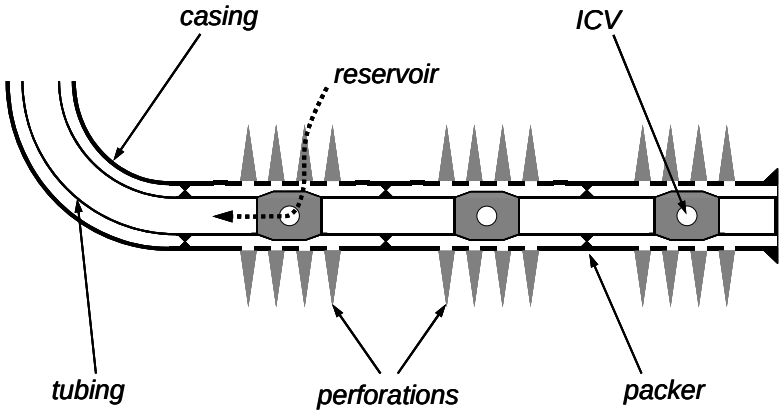


Figure 1.5: Schematic representation of a smart horizontal well equipped with three ICVs. These allow for individual control of the inflow from the reservoir, through the perforations and the ICVs, into the tubing.

However, soon it was realized that smart wells might allow for a more active control of flow in the near well bore area, see e.g. Sinha et al. (2001) and Jansen et al. (2002), or even to some extent of flow in the reservoir further away from the wells, in particular between injectors and producers. The latter is feasible in particular if the ICVs are located in individual reservoirs or reservoir zones, separated by more or less impermeable barriers. If the ICVs are located in a single reservoir, the scope to control the flow far away from the well is limited, due to the diffusive nature of the pressure propagation process. Especially if the reservoir is homogeneous, the 'radius of influence' around the well is restricted to about the distance between the ICVs (Ramakrishnan, 2007). Although in a strongly heterogeneous reservoir there is more scope to control the flow away from a single well equipped with multiple ICVs, in general flow control at a reservoir scale requires wells that are spatially distributed over the reservoir. This could be achieved with multiple horizontal or multi-lateral wells, or with conventional vertical wells, as long as they are equipped with remotely controllable valves (surface chokes and/or ICVs), and some form of pressure and flow rate measurement. Surface-mounted pressure gauges are sufficiently accurate for water injection wells, but for production wells, in which nearly always irregular gas-liquid flow occurs, the pressure measurements should preferably be performed with permanent downhole gauges (PDGs). Flow measurements may be performed using traditional periodic production tests, during which a well is rerouted to a test separator to allow for the measurement of the individual phase rates (i.e. oil, gas and water rates). Preferably, this should be combined with some form of a *soft sensing* system that produces continuous estimates of the phase rates from pressure drop measurements at various points in the production system, periodically calibrated against production tests. Alternatively, multiphase flow meters could be used to obtain more accurate continuous phase rate measurements. If the wells are completed in different reservoir layers that are separated by impermeable barriers, spatial control of the flow in the individual layers with multiple wells will require ICVs and, ideally, some form of pressure and flow measurement at each layer in each well. In addition to direct or indirect measurement of production variables (pressures and flow rates), there are several developments to obtain reservoir information from other sources during the producing life of

a field. Most notably is the use of '*4-dimensional' (4D) seismics*, also known as *time-lapse seismics*, to achieve an estimate of fluid front movements in the reservoir through observation of the differences in seismic images over time. Other developments, although much more in their infancy, are reservoir drainage imaging with the aid of *continuous resistivity measurements* in a well bore or between well bores, or through listening to *micro-seismicity* (cracking) around the well bore with down hole geophones. The combined infrastructure of wells, control valves, and sensors, and the associated data transmission and storage facilities, is variably referred to as *digital oil field*, *e-field*, *intelligent field* or *smart field*. In the remainder of this text we will no further discuss the hardware aspects of smart fields, but concentrate on the system theoretical aspects that are required to benefit from this hardware through improved reservoir management.

1.4 References for Chapter 1

Antsaklis, P.J. and Michel, A.N., 2006: *Linear systems*, Birkhäuser, Boston.

Antsaklis, P.J. and Michel, A.N., 2007: *A linear systems primer*, Birkhäuser, Boston.

Chierici, G.L., 1992: Economically improving oil recovery by advanced reservoir management. *Journal of Petroleum Science and Engineering* **8** (3) 205-219. DOI: 10.1016/0920-4105(92)90034-X.

DeCarlo, R.A., 1989: *Linear systems – a state variable approach with numerical implementation*, Prentice-Hall, Englewood Cliffs.

Friedland, B., 1986: *Control system design – An introduction to state-space methods*, McGraw-Hill. Reprinted in 2005 by Dover, New York.

Glad, T. and Ljung, L., 2000: *Control theory*, Taylor and Francis, London.

Jansen, J.D., Wagenvoort, A.M., Droppert, V.S., Daling, R., and Glandt, C.A., 2002: Smart well solutions for thin oil rims: Inflow switching and the smart stinger completion. Paper SPE 77942 presented at the *Asia Pacific Oil and Gas Conference and Exhibition*, Melbourne, Australia, 8-10 October. DOI: 10.2118/77942-MS.

Jansen, J.D., Brouwer, D.R., Nævdal, G. and van Kruijsdijk, C.P.J.W., 2005: Closed-loop reservoir management. *First Break*, January, **23**, 43-48.

Jansen, J.D., Bosgra, O.H. and van den Hof, P.M.J., 2008: Model-based control of multiphase flow in subsurface oil reservoirs. *Journal of Process Control* **18**, 846-855. DOI: 10.1016/j.jprocont.2008.06.011.

Kailath, T., 1980: *Linear systems*, Prentice-Hall, Englewood Cliffs.

Kapteijn, P.K.A. and Muessig, S., 2003: Smart fields: How to generate more value from hydrocarbon resources. *Oil Gas European Magazine* (3) OG1-OG6.

Litvak, M.L., Hutchins, L.A., Skinner, R.C., Darlow, B.L., Wood, R.C. and Kuest, L.J., 2002: Prudhoe bay e-field production optimization system based on integrated reservoir and facility simulation. Paper SPE 77643 presented at the *SPE Annual Technical Conference and Exhibition*, San Antonio, Texas, USA, 29 September - 2 October. DOI: 10.2118/77643-MS.

Luenberger, D.G., 1979: *Introduction to dynamic systems*, Wiley, New York.

Nygård, O., Cramer, C., Kulkarni, R. and Nordtvedt, J.E., 2001: Development of a marginal gas-condensate field using a novel integrated reservoir and production management approach. Paper SPE 68734 presented at the *SPE Asia Pacific Oil and Gas Conference and Exhibition*, Jakarta, Indonesia, 17 -19 April, DOI 10.2118/68734-MS.

Nyhavn, F., Vassenden, F. and Singstad, P., 2000: Reservoir drainage with down hole permanent monitoring and control systems. Paper SPE 62937 presented at the *SPE Annual*

Technical Conference and Exhibition, Dallas, Texas, USA, 1-4 October. DOI: 10.2118/62937-MS.

Olsder, G.J. and Van der Woude J.W., 2004: *Mathematical systems theory*, *3rd ed.*, VSSD, Delft.

Potters, H. and Kapteijn, P., 2005: Reservoir surveillance and smart fields. Paper IPTC 11039 presented at the *International Petroleum Technology Conference*, Doha, Qatar, 21-23 November. DOI: 10.2523/11039-MS.

Ramakrishnan, T.S., 2007: On reservoir fluid-flow control with smart completions. *SPE Production and Operations* **22** (1) 4-12. DOI: 10.2118/84219-PA.

Rossi, D.J., Gurpinar, O., Nelson, R. and Jacobsen, S., 2000: Discussion on integrating monitoring data into the reservoir management process. Paper SPE 65150 presented at the *SPE European Petroleum Conference*, Paris, France, 24-25 October. DOI: 10.2118/65150-MS.

Saputelli, L., Nikolaou, M. and Economides, M.J., 2005: Self-learning reservoir management. *SPE Reservoir Evaluation and Engineering* **8** (6): 534-547. DOI: 10.2118/84064-PA.

Sinha, S., Kumar, R., Vega, L. and Yalali, Y., 2001: Flow equilibration towards horizontal wells using downhole valves. Paper 68635 presented at the *SPE Asian Pacific Oil and Gas Conference and Exhibition*, Jakarta, Indonesia, April 17-19. DOI: 10.2118/68635-MS.

Skogestad, S. and Postlewaite, I., 2005: *Multivariable feedback control 2nd ed.*, Wiley, Chichester.

Unneland, T. and Hauser, M., 2005: Real-time asset management: From vision to engagement – An operator's experience. Paper SPE 96390 presented at the *SPE Annual Technical Conference and Exhibition*, Dallas, Texas, USA, 9-12 October. DOI: 10.2118/96390-MS.

2 Porous media flow

2.1 Introduction

This chapter gives a brief review of the basic equations to simulate flow through porous media. We will restrict the theory to the relatively simple cases of isothermal, slightly compressible single-phase and two-phase (oil-water) flow, which, however, are sufficient to illustrate most of the typical aspects involved. Moreover, we will only consider spatial discretizations using finite differences, which, however, is no limitation for the development of the theory in subsequent chapters. For more complex flows, involving multiple chemical components with multiple phases and possibly thermal effects and chemical interactions, we refer to standard textbooks on reservoir simulation such as Aziz and Settari (1979), Lake (1989), Mattax and Dalton (1990), or Chen et al. (2006). The latter also treats alternative spatial discretization methods such as finite elements.

2.2 Notation

Scalars will be indicated with Latin or Greek, lower or upper case letters, and vectors with Latin or Greek lower case letters in bold face or in index notation. Occasionally we will use bold italics to indicate vectors with a special meaning. Matrices will be indicated with Latin or Greek bold capitals. The superscript T is used to indicate the transpose, and dots above variables to indicate differentiation with respect to time. Unless indicated otherwise, vectors are always considered to be column vectors. E.g. a vector $\mathbf{x} \in \mathbb{R}^n$ is defined as

$$\mathbf{x} \triangleq \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}. \tag{2.1}$$

This expression also illustrates the use of the 'embellished' equality sign $\triangleq$ to introduce definitions.

2.3 Single-phase flow

2.3.1 Governing equations

General case

This section gives an overview of the derivation of the governing equations for single-phase flow. For details see e.g. Bear (1972), Peaceman (1977), Aziz and Settari (1979) or Helmig (1997). We consider one-dimensional, horizontal, iso-thermal flow of a compressible single-phase liquid through a compressible porous medium with constant cross-sectional area; see Figure 2.1.

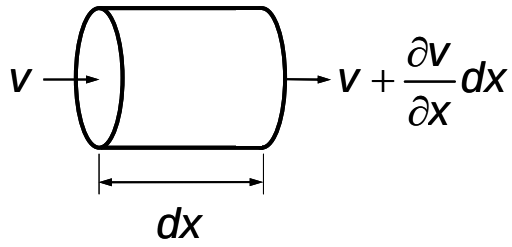


Figure 2.1: Control volume.

We can write the mass balance per unit time for a control volume with length *dx* as:

$$\underbrace{A\rho v}_{\text{mass rate in}} - \underbrace{A\left(\rho+\frac{\partial\rho}{\partial x}dx\right)\left(v+\frac{\partial v}{\partial x}dx\right)}_{\text{mass rate out}} - \underbrace{A\frac{\partial\rho\phi}{\partial t}dx}_{\frac{\text{mass accumulated}}{\text{unit time}}} + \underbrace{\rho q}_{\text{source term}} = 0 \ , \qquad (2.\ 2)$$

where A is the cross-sectional area, $\rho(t,x)$ is the fluid density, $v(t,x)$ is the *Darcy velocity* averaged over the cross-section[†], $\phi(t,x)$ is the porosity, x is the spatial coordinate, t is time and q is a source term. Positive values of q correspond to injection, negative values to production. If we expand equation (2. 2), drop all terms higher than first order in the differentials, and simplify the results, we obtain

$$\frac{\partial(\rho v)}{\partial x}+\frac{\partial\rho\phi}{\partial t}-\rho q'''=0 \ , \qquad (2.3)$$

where $q'''(t,x)$ is a now a source term expressed as flow rate per unit volume. We can generalize this result to a situation with non-constant cross-section or to two or three dimensions by writing

$$\nabla\cdot\left(\alpha\rho\vec{\mathbf{v}}\right)+\alpha\frac{\partial(\rho\phi)}{\partial t}-\alpha\rho q'''=0 \ , \qquad (2.4)$$

where $\nabla\cdot$ is the divergence operator, $\alpha(\vec{\mathbf{x}})$ is a geometric factor that will be defined below, $\vec{\mathbf{x}}$ is the spatial coordinate vector with components x, y and z, and $\vec{\mathbf{v}}(t,\vec{\mathbf{x}})$ is the Darcy velocity vector[†]. Depending on whether we consider one-, two- or three-dimensional flow, the factor α, the vectors $\vec{\mathbf{x}}$ and $\vec{\mathbf{v}}$, and the divergence operator are given by:

[†] The *Darcy velocity* or the *filtration velocity*, is the superficial velocity that would occur if the entire cross section, and not just the pores, would be open to flow. This is as opposed to the *interstitial velocity* $\tilde{v}$, which is defined as $\tilde{v}=v/\phi$, and which is the true fluid velocity in the pore space. The Darcy velocity can also be interpreted as a *volumetric flux*, i.e. the amount of volume flowing through a unit of surface area per unit time.

[†] We use an arrow above a vector or matrix to indicate that it its components are representing quantities in physical space. E.g. $\vec{\mathbf{v}}$ is a velocity vector with one, two or three components, depending on whether we use a one-, two-, or three-dimensional reservoir description. Note that the spatial coordinate vector $\vec{\mathbf{x}}$ as used in this appendix is unrelated to the state vector $\mathbf{x}$ as used in the body of the text. The use of the same symbol for two different quantities is somewhat unfortunate, but results from conventions in different disciplines.

$$
\begin{aligned}
&\text{1-D:} \quad \alpha = A(x)\,, \quad \vec{\mathbf{x}} = x\,, \quad \vec{\mathbf{v}} = v\,, \quad \nabla\cdot\bullet = \frac{\partial\bullet}{\partial x} \\
&\text{2-D:} \quad \alpha = h(x,y)\,, \quad \vec{\mathbf{x}} = (x,y)\,, \quad \vec{\mathbf{v}} = (v_x, v_y)\,, \quad \nabla\cdot\bullet = \frac{\partial\bullet}{\partial x} + \frac{\partial\bullet}{\partial y} \\
&\text{3-D:} \quad \alpha = 1\,, \quad \vec{\mathbf{x}} = (x,y,z)\,, \quad \vec{\mathbf{v}} = (v_x, v_y, v_z)\,, \quad \nabla\cdot\bullet = \frac{\partial\bullet}{\partial x} + \frac{\partial\bullet}{\partial y} + \frac{\partial\bullet}{\partial z}
\end{aligned}
\tag{2.5}
$$

where h is reservoir height. Conservation of momentum in flow through porous media is usually expressed with Darcy's law, an experimental relationship that contains only resistance and gravity terms. Disregarding the inertia forces is justified because of the very low flow velocities that occur in porous media flow; see Bear (1972). For the one-dimensional, horizontal case with constant cross-sectional area Darcy's law can be expressed as

$$
v = -\frac{k}{\mu}\frac{\partial p}{\partial x}\,, \tag{2.6}
$$

where $k(x)$ is the rock *permeability*[‡], and μ is the fluid viscosity. For the more general, non-horizontal, one-, two- or three-dimensional case we can write the same equation in vector form as

$$
\vec{\mathbf{v}} = -\frac{1}{\mu}\vec{\mathbf{K}}\left(\nabla p - \rho g \nabla d\right)\,, \tag{2.7}
$$

where

$$
\nabla\bullet \triangleq \frac{\partial\bullet}{\partial x}\,, \quad \nabla\bullet \triangleq \begin{bmatrix} \frac{\partial\bullet}{\partial x} & \frac{\partial\bullet}{\partial y} \end{bmatrix}^T \text{ or } \nabla\bullet \triangleq \begin{bmatrix} \frac{\partial\bullet}{\partial x} & \frac{\partial\bullet}{\partial y} & \frac{\partial\bullet}{\partial z} \end{bmatrix}^T\,, \tag{2.8}
$$

is the gradient operator for one, two or three dimensions respectively, and where $\vec{\mathbf{K}}(\vec{\mathbf{x}})$ is the permeability tensor, g is the acceleration of gravity and $d(\vec{\mathbf{x}})$ is depth. Usually the orientation of the coordinate system can be aligned with the geological layering in the reservoir such that $\vec{\mathbf{K}}$ is a diagonal matrix:

$$
\text{1-D:} \quad \vec{\mathbf{K}} = k\,, \quad \text{2-D:} \quad \vec{\mathbf{K}} = \begin{bmatrix} k_x & 0 \\ 0 & k_y \end{bmatrix}\,, \quad \text{3-D:} \quad \vec{\mathbf{K}} = \begin{bmatrix} k_x & 0 & 0 \\ 0 & k_y & 0 \\ 0 & 0 & k_z \end{bmatrix}\,. \tag{2.9}
$$

Combining equations (2.4) and (2.7) results in

$$
-\nabla\cdot\left[\frac{\alpha\rho}{\mu}\vec{\mathbf{K}}\left(\nabla p - \rho g \nabla d\right)\right] + \alpha\frac{\partial(\rho\phi)}{\partial t} - \alpha\rho q''' = 0\,. \tag{2.10}
$$

The variables ρ, ϕ, μ and $\vec{\mathbf{K}}$ in equation (2.10) may all be functions of the pressure p. However, the dependency of μ and $\vec{\mathbf{K}}$ on p is usually very small and to simplify our formulation we will therefore assume from now on that these parameters are pressure-

[‡] Permeability has a dimension of length squared and is therefore expressed in SI units in m^2. In reservoir engineering use is often made of Darcy units, which are defined as: 1 D = 9.869 233 × 10^{-13} ≈ 10^{-12} m^2.

independent. The relationship between ρ and p follows from the equation of state for a liquid, which can be written in differential form as

$$c_l(p) \triangleq \frac{1}{\rho}\frac{\partial \rho}{\partial p}\bigg|_{T_0} , \tag{2.11}$$

where $c_l(p)$ is the iso-thermal liquid compressibility and T_0 is a constant reference temperature, for which the reservoir temperature T_R is a logical choice. Similarly, the relationship between ϕ and p is given by

$$c_r(p) \triangleq \frac{1}{\phi}\frac{\partial \phi}{\partial p}\bigg|_{T_0} , \tag{2.12}$$

where $c_r(p)$ is the rock compressibility. Equations (2.11) and (2.12) are nonlinear ordinary differential equations for the dependent variables ρ and ϕ respectively as a function of the independent variable p. They are of first-order and therefore require a boundary condition each, which can be specified as:

$$\rho\big|_{p=p_0} = \rho_0 \text{ and } \phi\big|_{p=p_0} = \phi_0 . \tag{2.13, 2.14}$$

With the aid of equations (2.11) and (2.12) we can rewrite the accumulation term $\partial(\rho\phi)/\partial t$ in equation (2.10) as

$$\frac{\partial(\rho\phi)}{\partial t} = \rho\frac{\partial \phi}{\partial t} + \phi\frac{\partial \rho}{\partial t} = \left(\rho\frac{\partial \phi}{\partial p} + \phi\frac{\partial \rho}{\partial p}\right)\frac{\partial p}{\partial t} = \rho\phi(c_l + c_r)\frac{\partial p}{\partial t} = \rho\phi c_t\frac{\partial p}{\partial t} , \tag{2.15}$$

where $c_t = (c_l + c_r)$ is known as the total compressibility. Combining equations (2.10) and (2.15) results in

$$\underbrace{-\nabla\cdot\left[\frac{\alpha\rho}{\mu}\vec{\mathbf{K}}(\nabla p - \rho g\nabla d)\right]}_{\text{flux term}} + \underbrace{\alpha\rho\phi c_t\frac{\partial p}{\partial t}}_{\text{accumulation term}} - \underbrace{\alpha\rho q'''}_{\text{source term}} = 0 . \tag{2.16}$$

Equation (2.16) is a nonlinear partial differential equation for the dependent variable p as a function of the independent variables $\vec{\mathbf{x}}$ and t. It is of first order in t and of second order in $\mathbf{x}$, and therefore requires an initial condition and two boundary conditions for each coordinate direction. The initial condition for p can be written as

$$p(t,\vec{\mathbf{x}})\big|_{t=t_0} = \breve{p}(\vec{\mathbf{x}}) . \tag{2.17}$$

Boundary conditions are usually specified in terms of p (Dirichlet conditions) or $\partial p/\partial\vec{\mathbf{n}}$ (Neumann conditions where $\vec{\mathbf{n}}$ is the outward pointing unit normal vector on the boundary[†]. With the aid of Darcy's law (equation (2.7)) the Neumann conditions can be expressed in terms of the velocity, i.e. the flow rate per unit area at the boundary. Therefore we can express the two types of boundary conditions as:

$$p(t,\vec{\mathbf{x}})\big|_{\Gamma_p} = \breve{\breve{p}}(t) \text{ and } q''(t,\vec{\mathbf{x}})\big|_{\Gamma_q} = \breve{\breve{q}}''(t) , \tag{2.18, 2.19}$$

[†] More complicated boundary conditions are possible, e.g. by specifying a relationship between p and $\partial p/\partial\mathbf{n}$, a so-called mixed boundary condition. Furthermore, different boundary conditions may be specified along different parts of the boundary.

where Γ is the boundary of the domain Ω to which equation (2.16) applies, q'' is the outward flow rate per unit area normal to the boundary.

Linearized case

In case of a weakly compressible liquid and relatively small pressure differences we may assume that ρ and ϕ can be linearized at a reference pressure p_0, while c_l and c_r remain constant. E.g. to linearize the density, we integrate equation (2.11) for constant c_l to obtain

$$\rho = C\exp\left(c_l p\right), \tag{2.20}$$

then determine the integration constant C from condition (2.13), use a Taylor expansion to approximate the exponential function, and maintain terms up to first order leading to

$$\rho = \rho_0 \exp\left[c_l\left(p - p_0\right)\right] \approx \rho_0\left[1 + c_l\left(p - p_0\right)\right]. \tag{2.21}$$

Similarly, we find for the porosity

$$\phi \approx \phi_0\left[1 + c_r\left(p - p_0\right)\right]. \tag{2.22}$$

Next, we also assume that the permeability is isotropic, i.e. that the tensor $\vec{\mathbf{K}}$ can be replaced by a scalar k, and that α, k and μ are constant over the spatial domain. Substitution of expressions (2.21) and (2.22) in equation (2.10), disregarding small terms containing the products $c_l c_r$, $(\nabla p)^2 c_l$ and $\nabla p(p - p_0)c_l$, and dividing out $\alpha\rho_0$ results in the linear equation

$$-\frac{k}{\mu}\nabla^2\left(p - \rho_0 g d\right) + \phi_0 c_t \frac{\partial p}{\partial t} - q''' = 0\ . \tag{2.23}$$

If we furthermore define the potential $\Phi = p - \rho_0 g d$, equation (2.23) can be expressed as a linear diffusion equation

$$\frac{\partial \Phi}{\partial t} = \zeta \nabla^2 \Phi + Q\ , \tag{2.24}$$

where $\zeta = k/(\mu\phi_0 c_t)$ is the diffusion constant, and $Q = q'''/(\phi_0 c_t)$ is a scaled source term.

2.3.2 Finite difference discretization

Formulation

This sub-section presents a straightforward approach to the semi-discretization of equation (2.16) with the aid of finite differences for 2-D flow. The next sub-section will present an alternative approach that should be used if it is required that the numerical scheme exactly satisfies the mass conservation equations. For further information on finite difference discretizations, see Peaceman (1977), Aziz and Settari (1979) and Mattax and Dalton (1990). For alternative discretization methods such as finite volume or finite element methods, see e.g. Patankar (1980), Helmig (1997), or Chen *et al*. (2006). First we rewrite equation (2.16) in scalar 2-D form, assuming isotropic permeability, small rock and fluid compressibilities, uniform reservoir thickness and absence of gravity forces:

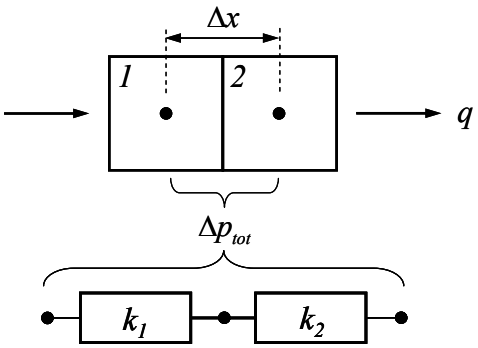
$$-\frac{h}{\mu}\frac{\partial}{\partial x}\left(k\frac{\partial p}{\partial x}\right) - \frac{h}{\mu}\frac{\partial}{\partial y}\left(k\frac{\partial p}{\partial y}\right) + h\phi_0 c_t \frac{\partial p}{\partial t} - hq''' = 0\,. \tag{2.25}$$

Just like equation (2.23), equation (2.25) is linear in p. However, unlike equation (2.23), equation (2.25) does not contain gravity effects, while it still does have the option of a spatial

variability of k. Moreover, in equation (2.25), we have not divided out the geometric factor h, to stay in line with usual textbook derivation of the discretized equations. Note that because of dividing out the density ρ_0 equation (2.25) is now expressed in m^3s^{-1}. We apply a block-centered central difference scheme with uniform grid to approximate the spatial differentials. The first term in equation (2.25) can then be rewritten as

$$\frac{h}{\mu}\frac{\partial}{\partial x}\left(k\frac{\partial p}{\partial x}\right)\approx\frac{h}{\mu}\frac{\Delta}{\Delta x}\left(k\frac{\Delta p}{\Delta x}\right)=\frac{h}{\mu}\frac{k_{i+\frac{1}{2},j}\left(p_{i+1,j}-p_{i,j}\right)-k_{i-\frac{1}{2},j}\left(p_{i,j}-p_{i-1,j}\right)}{\left(\Delta x\right)^2}\,, \qquad (2.26)$$

where i and j are grid block counters in x and y direction, and where the subscripts $i+½, j$ and $i-½, j$ indicate averaged values at the boundaries between grid blocks (i, j) and $(i+1, j)$, and grid blocks $(i-1, j)$ and (i, j) respectively†. In analogy to electrical resistances in series we can work out the series resistance against flow between two grid block centers; see Figure 2.2 for a 1-D example.



$$\Delta p_{tot}=\frac{\mu}{k_{av}}\frac{\Delta x}{A}q=\Delta p_1+\Delta p_2=\frac{\mu}{k_1}\frac{\frac{1}{2}\Delta x}{A}q+\frac{\mu}{k_2}\frac{\frac{1}{2}\Delta x}{A}q$$

$$\frac{1}{k_{av}}=\frac{\frac{1}{2}}{k_1}+\frac{\frac{1}{2}}{k_2}\quad\Rightarrow\quad k_{av}=\frac{2}{\frac{1}{k_1}+\frac{1}{k_2}}$$

Figure 2.2: One-dimensional example of harmonic averaging of grid block permeabilities.

Similarly, considering flow in the x-direction for the 2-D case we can write

$$\frac{\mu}{h\Delta y}\frac{\Delta x}{k_{i-\frac{1}{2},j}}=\frac{\mu}{h\Delta y}\frac{\frac{1}{2}\Delta x}{k_{i-1,j}}+\frac{\mu}{h\Delta y}\frac{\frac{1}{2}\Delta x}{k_{i,j}}\,, \qquad (2.27)$$

from which we obtain the *harmonic average* for the permeability:

† This two-dimensional grid block numbering is introduced to obtain a systematic description of the transmissibilities in a two-dimensional reservoir model. In the MATLAB implementation, however, we used a one-dimensional grid block numbering as displayed in Figure 2.3, and a *connectivity table* to list the pairs of adjacent grid blocks. See also Table 2-2 which illustrates the two different numbering systems as applied to Example 1.

$$k_{i-\frac{1}{2},j} \triangleq \frac{2}{\dfrac{1}{k_{i-1,j}} + \dfrac{1}{k_{i,j}}} \ . \tag{2.28}$$

A similar expression can be obtained for $k_{i+½j}$. After rewriting the second term in equation (2.25) in the same fashion, and reorganizing terms we can write

$$\begin{aligned} &Vc_t(\phi_0)_{i,j}\left(\frac{\partial p}{\partial t}\right)_{i,j} - T_{i-\frac{1}{2},j}p_{i-1,j} - T_{i,j-\frac{1}{2}}p_{i,j-1} + \\ &\left(T_{i-\frac{1}{2},j} + T_{i,j-\frac{1}{2}} + T_{i,j+\frac{1}{2}} + T_{i+\frac{1}{2},j}\right)p_{i,j} - T_{i,j+\frac{1}{2}}p_{i,j+1} - T_{i+\frac{1}{2},j}p_{i+1,j} = Vq'''_{i,j} \ , \end{aligned} \tag{2.29}$$

where $V = h\,\Delta x\,\Delta y$ is the grid block volume (taken identical for all grid blocks) and where

$$T_{i-\frac{1}{2},j} \triangleq \frac{\Delta y}{\Delta x}\frac{h}{\mu}k_{i-\frac{1}{2},j} \ . \tag{2.30}$$

is the transmissibility between grid blocks (i-1, j) and (i, j), with similar expressions for the other transmissibilities. Equation (2.29) can be rewritten in vector form as

$$\begin{bmatrix} 0 & \cdots & 0 & Vc_t(\phi_0)_{i,j} & 0 & \cdots & 0 \end{bmatrix} \begin{bmatrix} \dot{p}_{i,j-1} \\ \vdots \\ \dot{p}_{i-1,j} \\ \dot{p}_{i,j} \\ \dot{p}_{i+1,j} \\ \vdots \\ \dot{p}_{i,j+1} \end{bmatrix} +$$

$$\begin{bmatrix} -T_{i,j-\frac{1}{2}} & \cdots & -T_{i-\frac{1}{2},j} & \left(T_{i,j-\frac{1}{2}} + T_{i-\frac{1}{2},j} + T_{i+\frac{1}{2},j} + T_{i,j+\frac{1}{2}}\right) & -T_{i+\frac{1}{2},j} & \cdots & -T_{i,j+\frac{1}{2}} \end{bmatrix} \begin{bmatrix} p_{i,j-1} \\ \vdots \\ p_{i-1,j} \\ p_{i,j} \\ p_{i+1,j} \\ \vdots \\ p_{i,j+1} \end{bmatrix} = q_{i,j} \ , \tag{2.31}$$

where we have used dots above variables to indicate differentiation with respect to time, and where we have changed from flow rates per unit volume q''' to flow rates q expressed in m^3/s. The row vectors in the first and second term of equation (2.31) form building blocks for matrices that represent the flow behavior of a collection of grid blocks. The second term of the equation illustrates that, for the chosen 2-D discretization, the change of a grid block pressure at a certain moment in time is a function of its own value and of the pressure values in the four neighboring grid blocks. The vector with transmissibility matrix elements has therefore typically five non-zero elements. Only the rows that correspond to grid blocks at the edges of the domain require a special treatment to incorporate the boundary conditions. For no-flow boundary conditions this simply means that they only have four non-zero elements,

because the fifth one, which represents the boundary transmissibility, is equal to zero. For rows corresponding to grid blocks at a corner of the domain the number of non-zero elements reduces to three. For a system with n grid blocks we can specify n equations of the form (2.31), which, when combined, can be written as

$$\mathbf{V}\dot{\mathbf{p}} + \mathbf{T}\mathbf{p} = \mathbf{q}\,. \tag{2.32}$$

The $n \times n$ matrices **T** and **V** are generally known as the *transmissibility matrix* and the *accumulation matrix* respectively.

2.3.3 Example 1 – Single-phase flow in a simple reservoir

We illustrate the structure of the matrices with a (very) simple example. It consists of a finite difference model of a two-dimensional horizontal reservoir with two vertical wells: an injector in block 1 and a producer in block 6. Figure 2.3 displays the block-centered finite difference model with six grid blocks. The reservoir and fluid properties have been listed in Table 2.1.

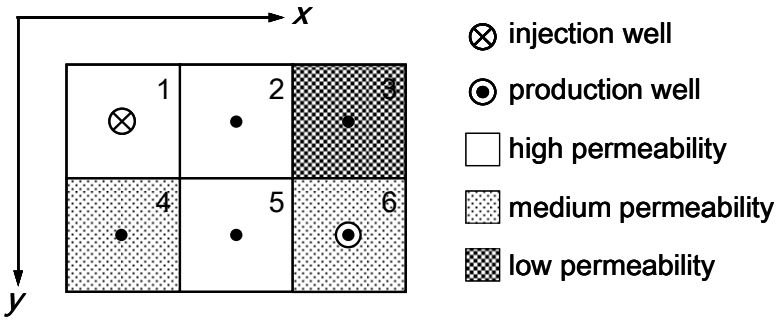


Figure 2.3: Top view of a six-block finite difference model of a reservoir with two wells.

Table 2.1: Reservoir and fluid properties for Example 1.

Symbol	*Variable*	*Value*	*SI units*	*Value*	*Field units*
h	Grid block height	20	m	66	ft
Δx, Δy	Grid block length/width	500	m	1640	ft
μ	Oil dynamic viscosity	5.0×10^{-4}	Pa s	0.5	cP
k_{low}	Permeability, low	1.0×10^{-14}	m^2	10	mD
k_{med}	Permeability, medium	1.0×10^{-13}	m^2	101	mD
k_{high}	Permeability, high	1.0×10^{-12}	m^2	1013	mD
ϕ	Porosity	0.3	-	0.3	-
c_t	Total compressibility[†]	2.0×10^{-8}	Pa^{-1}	1.4×10^{-4}	psi^{-1}
$\breve{p}_R$	Initial reservoir pressure	30×10^6	Pa	4351.1	psi
r_{well}	Well bore radius	0.114	m	4.50	in

[†] The value of the compressibility has been chosen about a factor 10 higher than would be expected for an oil-reservoir above bubble point pressure. This results in additional energy storage in the reservoir, an effect that in reality would occur in the presence of an aquifer and/or a gas cap.

The transmissibility matrix **T** for the six grid blocks of Example 1 can be composed as follows:

$$\begin{bmatrix} \left(T_{1,1\frac{1}{2}}+T_{1\frac{1}{2},1}\right) & -T_{1\frac{1}{2},1} & 0 & -T_{1,1\frac{1}{2}} & 0 & 0 \\ -T_{1\frac{1}{2},1} & \left(T_{1\frac{1}{2},1}+T_{2,1\frac{1}{2}}+T_{2\frac{1}{2},1}\right) & -T_{2\frac{1}{2},1} & 0 & -T_{2,1\frac{1}{2}} & 0 \\ 0 & -T_{2\frac{1}{2},1} & \left(T_{2\frac{1}{2},1}+T_{3,1\frac{1}{2}}\right) & 0 & 0 & -T_{3,1\frac{1}{2}} \\ -T_{1,1\frac{1}{2}} & 0 & 0 & \left(T_{1,1\frac{1}{2}}+T_{1\frac{1}{2},2}\right) & -T_{1\frac{1}{2},2} & 0 \\ 0 & -T_{2,1\frac{1}{2}} & 0 & -T_{1\frac{1}{2},2} & \left(T_{1\frac{1}{2},2}+T_{2,1\frac{1}{2}}+T_{2\frac{1}{2},2}\right) & -T_{2\frac{1}{2},2} \\ 0 & 0 & -T_{3,1\frac{1}{2}} & 0 & -T_{2\frac{1}{2},2} & \left(T_{2\frac{1}{2},2}+T_{3,1\frac{1}{2}}\right) \end{bmatrix} \tag{2.33}$$

Using the data of Table 2.1 we can work out the numerical values of the transmissibilities. The results have been displayed in Table 2-2.

Table 2-2: Transmissibilities for Example 1.[†]

Connectivity	*Grid block pair*	*i-j Transmissibility numbering*	*Transmissibility,* $m^3/(Pa\ s)$
1	1-2	(1½, 1)	4.000×10^{-8}
2	1-4	(1, 1½)	0.727×10^{-8}
3	2-3	(2½, 1)	0.079×10^{-8}
4	2-5	(2, 1½)	4.000×10^{-8}
5	3-6	(3, 1½)	0.073×10^{-8}
6	4-5	(1½, 2)	0.727×10^{-8}
7	5-6	(2½, 2)	0.727×10^{-8}

The accumulation matrix for the six grid blocks of Example 1 becomes simply

$$\begin{bmatrix} Vc_t(\phi_0)_1 & 0 & 0 & 0 & 0 & 0 \\ 0 & Vc_t(\phi_0)_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & Vc_t(\phi_0)_3 & 0 & 0 & 0 \\ 0 & 0 & 0 & Vc_t(\phi_0)_4 & 0 & 0 \\ 0 & 0 & 0 & 0 & Vc_t(\phi_0)_5 & 0 \\ 0 & 0 & 0 & 0 & 0 & Vc_t(\phi_0)_6 \end{bmatrix}. \tag{2.34}$$

Equation (2.32) can now be worked out as

[†] The two-dimensional (*i-j*) transmissibility numbering in the third column is shown as illustration only. In the MATLAB implementation we used the connectivities displayed in the first column.

$$10^{-1}\begin{bmatrix} 0.300 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.300 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.300 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.300 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.300 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.300 \end{bmatrix}\begin{bmatrix} \dot{p}_1 \\ \dot{p}_2 \\ \dot{p}_3 \\ \dot{p}_4 \\ \dot{p}_5 \\ \dot{p}_6 \end{bmatrix} + 10^{-8}\begin{bmatrix} 4.727 & -4.000 & 0 & -0.727 & 0 & 0 \\ -4.000 & 8.079 & -0.079 & 0 & -4.000 & 0 \\ 0 & -0.079 & 0.151 & 0 & 0 & -0.073 \\ -0.727 & 0 & 0 & 1.454 & -0.727 & 0 \\ 0 & -4.000 & 0 & -0.727 & 5.454 & -0.727 \\ 0 & 0 & -0.073 & 0 & -0.727 & 0.800 \end{bmatrix}\begin{bmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \\ p_5 \\ p_6 \end{bmatrix} = \begin{bmatrix} q_1 \\ 0 \\ 0 \\ 0 \\ 0 \\ q_6 \end{bmatrix}, \tag{2.35}$$

2.3.4 Incompressible flow

In the special case that the fluid and rock compressibilities are so small that they may be neglected, it follows from equation (2.31) that $\mathbf{V} = \mathbf{0}$. In that case we can rewrite equation (2.32) as

$$\mathbf{T}\mathbf{p} = \mathbf{q}. \tag{2.36}$$

At first sight it appears if equation (2.36) can simply be solved for the constant pressure **p**. However, as discussed in Section 4.1.4, the transmissibility matrix **T** is singular which implies that we cannot directly solve the equation. The singularity can be removed through prescribing the pressure in at least one of the grid blocks, or through the use of a well model, in which case we may indeed solve for **p** as discussed in more detail in Section 4.2.3.

2.3.5 Mass-conservative formulation*

The numerical simulation of a physical process using a discretized form of the governing PDEs generally results in an approximate solution of those PDEs. In case of the simulation of reservoir flow this implies that both the mass conservation equation and Darcy's law may not be represented accurately. In reservoir engineering there is often a desire to adhere to the mass conservation equation as much as possible, because most simulations are made to predict recoverable hydrocarbon volumes in some sense. Depending on the discretization used, mass conservation may be more or less compromised. In discussing this issue we will to a large extent follow the approach of Aziz and Settari (1979), pages 93-97. The effect of the discretization on the mass balance error can be understood by considering equation (2.31) which can be interpreted as the mass balance equation for a single grid block. It basically states that the mass accumulation rate of a grid block plus the sum of the mass fluxes to or from the four neighboring grid blocks equals the flow rate of the source term. Note that the matrix coefficients in each row of the transmissibility matrix exactly add up, a property that is also apparent from inspecting the transmissibility matrices in equations (2.33) and (2.35). Because of the symmetry of the transmissibility matrix, the same property holds for each column. Adding the rows of equation (2.31) we therefore obtain

$$\sum_{i=1}^{n_x}\sum_{j=1}^{n_y}\left[Vc_t\left(\phi_0\right)_{i,j}\dot{p}_{i,j}+q_{i,j}\right], \tag{2.37}$$

where the double sums indicate summation over all grid blocks in the x and y directions, and where the transmissibility terms do no longer appear. Equation (2.37) can be interpreted as the mass balance equation for the entire system, and it states that the sum of the mass accumulation rates in all grid blocks equals the sum of the source terms. The summation of the source terms does not involve any approximation, and therefore any mass balance error in the numerical solution results from errors in the accumulation terms $Vc_t\phi_0\dot{p}$. In our derivation of equation (2.25), the starting point for the spatial discretization, we used the assumption of small and constant compressibilities c_l and c_r, and in substituting expressions (2.21) and (2.22) in equation (2.10) we therefore disregarded small terms containing the products $c_l c_r$. A straightforward time discretization of the accumulation terms in equation (2.31) in the form of

$$Vc_t\phi_0\frac{\partial p}{\partial t}\approx Vc_t\phi_0\frac{p_{k+1}-p_k}{\Delta t} \tag{2.38}$$

where k is the time step indicator, is therefore, in general, not mass conservative. A mass-conservative time discretization can be obtained by starting from the original form of the accumulation term, $\partial(\rho\phi)/\partial t$, as present in equation (2.10). We can now write

$$\begin{aligned}\frac{\partial\left(\rho\phi\right)}{\partial t}&\approx\frac{1}{\Delta t}\left[\rho_{k+1}\left(\phi_{k+1}-\phi_k\right)+\left(\rho_{k+1}-\rho_k\right)\phi_k\right]\\&\approx\frac{1}{\Delta t}\left[\rho_{k+1}\phi_0c_r\left(p_{k+1}-p_k\right)+\rho_0c_l\left(p_{k+1}-p_k\right)\phi_k\right]\\&\left(\rho_{k+1}\phi_0c_r+\rho_0\phi_kc_l\right)\frac{p_{k+1}-p_k}{\Delta t}.\end{aligned} \tag{2.39}$$

In the process of spatially discretizing equation (2.25) to arrive at equation (2.31) we multiplied with V and divided by ρ_0. Starting from equation (2.39), the mass-conservative discretization for the accumulation term in equations (2.31) can therefore be written as

$$Vc_t\phi_0\frac{\partial p}{\partial t}\approx V\left(\frac{\rho_{k+1}}{\rho_0}\phi_0c_r+\phi_kc_l\right)\frac{p_{k+1}-p_k}{\Delta t}\ . \tag{2.40}$$

Comparison with equation (2.38) shows that the constant coefficient $Vc_t\phi_0$ has been replaced by a state-dependent coefficient, which, moreover, contains an element ρ_{k+1} that should be computed a the new time step k+1. A mass-conservative implementation therefore always requires some form of implicit time integration. For liquid flow, and as long as the pressure changes in the reservoir remain small compared to the total pressure, the effect of mass balance errors is small, and therefore we do not make use of the strict mass-conserving formulation in our numerical examples. However if compressibility plays a role, e.g. when free gas is present, the use of a mass conservative formulation is essential.

2.3.6 Well models

Formulation

The flow between two grid blocks is linearly proportional to the product of pressure drop Δp and transmissibility T_{gb}, as was discussed in Section 2.3.2 above. However, the pressure close

to a well displays very strong, nonlinear, gradients in the radial direction, and to capture this effect accurately a very fine grid around the well is required. Alternatively, one may attempt to model an additional pressure drop based on some analytical or semi-analytical solution for the converging flow around a well. Many authors have treated this topic, see e.g. Aziz and Settari (1979) and the classic paper of Peaceman (1978). For an overview of methods for wells with complex geometries see Ding *et al*. (2000). Here we will follow Peaceman, who developed an expression for the additional pressure drop due to steady-state cylindrical radial flow towards a well in the centre of a grid block. In general, the pressure p as a function of radial distance r from a production well operating at bottom hole pressure p_{well}, in a homogeneous reservoir with permeability k, and producing fluid with viscosity μ, is given by the logarithmic relationship

$$p = p_{well} - \frac{\mu q}{2\pi kh} \ln\left(\frac{r}{r_{well}}\right) , \tag{2.41}$$

a classic result that follows from solving the steady-state differential equation for radial flow through a porous medium, see e.g. Bear (1972). Note that a negative value of the flow rate q indicates production, and a positive value injection. According to equation (2.41) the pressure in an injection well would decrease without limit for increasing r. Similarly, the pressure in a production well would increase without limit. The expression has therefore only physical relevance for a finite domain, bounded by e.g. a circular constant-pressure boundary. Peaceman demonstrated that for the particular case of a repeated five-spot injection-production configuration, the analytical solution for the pressure drop and the numerical solution using a fine grid produce the same value for p at an *equivalent* radial distance from the well in the order of

$$r_{eq} \approx 0.2\,\Delta x , \tag{2.42}$$

where Δx is the length of the (square) grid blocks. Although this result is only valid for a rather restricted set of assumptions, it has proved to be a very useful basis to model the near-well pressure drop for simple, vertical, wells in regular grids. In a follow-up paper, Peaceman argued that for rectangular grid blocks with length Δx and width Δy, expression (2.42) should be modified to

$$r_{eq} = 0.14\sqrt{\Delta x^2 + \Delta y^2} \ ; \tag{2.43}$$

see Peaceman (1983). Combining equations (2.41) and (2.43) we find for the additional pressure drop between the grid block pressure and the well bottom hole pressure

$$p_{gb} - p_{well} = -\frac{q}{J_{well}} = -\frac{\mu q}{2\pi kh} \ln\left(\frac{0.14\sqrt{\Delta x^2 + \Delta y^2}}{r_{well}}\right) , \tag{2.44}$$

where J_{well} is known as the *well index* or *productivity index*, and where negative flow rates indicate production.

Example 1 – well model

If we consider the wells in our Example 1 and fill in the numerical values of Table 2.1 we find that $J_{well,11} = 3.72\times10^{-8}$ m^3/(Pa s) and $J_{well,66} = 3.72\times10^{-9}$ m^3/(Pa s).

2.4 Two-phase flow

2.4.1 Governing equations

This section gives a brief overview of the derivation of the governing PDEs for two-phase (oil-water) flow, using the *simultaneous solution method* formulated in p_o and S_w as described in Aziz and Settari (1979), p.133. We consider iso-thermal conditions only and we will formulate the equations in terms of in-situ volumes. The often applied formulation in terms of surface volumes, using formation volume factors, is not necessary for our purpose. The mass balance equations can be expressed for each of the phases as

$$\begin{aligned} \nabla\cdot\left(\alpha\rho_w\vec{\mathbf{v}}_w\right)+\alpha\frac{\partial\left(\rho_w\phi S_w\right)}{\partial t}-\alpha\rho_w q_w'''=0\,,\\ \nabla\cdot\left(\alpha\rho_o\vec{\mathbf{v}}_o\right)+\alpha\frac{\partial\left(\rho_o\phi S_o\right)}{\partial t}-\alpha\rho_o q_o'''=0\,, \end{aligned} \qquad (2.45, 2.46)$$

where subscripts w an o are used to identify water and oil, and where S_w and S_o are the saturations, defined as the proportion of the pore space occupied by the respective phase. Darcy's law can now be expressed as

$$\begin{aligned} \vec{\mathbf{v}}_w=-\frac{k_{rw}}{\mu_w}\vec{\mathbf{K}}\left(\nabla p_w-\rho_w g\nabla d\right),\\ \vec{\mathbf{v}}_o=-\frac{k_{ro}}{\mu_o}\vec{\mathbf{K}}\left(\nabla p_o-\rho_o g\nabla d\right), \end{aligned} \qquad (2.47, 2.48)$$

where k_{rw} and k_{ro} are the relative permeabilities, which represent the additional resistance to flow of a phase caused by the presence of the other phase. For an explanation of the underlying physical concepts, see e.g. Lake (1989). Combining equations (2.45) to (2.48) we obtain:

$$\begin{aligned} -\nabla\cdot\left[\frac{\alpha\rho_w k_{rw}}{\mu_w}\vec{\mathbf{K}}\left(\nabla p_w-\rho_w g\nabla d\right)\right]+\alpha\frac{\partial\left(\rho_w S_w\phi\right)}{\partial t}-\alpha\rho_w q_w'''=0\,,\\ -\nabla\cdot\left[\frac{\alpha\rho_o k_{ro}}{\mu_o}\vec{\mathbf{K}}\left(\nabla p_o-\rho_o g\nabla d\right)\right]+\alpha\frac{\partial\left(\rho_o S_o\phi\right)}{\partial t}-\alpha\rho_o q_o'''=0\,. \end{aligned} \qquad (2.49, 2.50)$$

Equations (2.49) and (2.50) together contain four unknowns, p_w, p_o S_w and S_o, two of which can be eliminated with aid of the relationships

$$S_w+S_o=1\,,\quad p_o-p_w=p_c\left(S_w\right) \qquad (2.51, 2.52)$$

where $p_c(S_w)$ is the oil-water capillary pressure. Substituting equations (2.51) and (2.52) in equations (2.49) and (2.50), expanding the right-hand sides, applying chain-rule differentiation, and substituting the isothermal equations of state

$$c_o\triangleq\frac{1}{\rho_o}\frac{\partial\rho_o}{\partial p_o}\bigg|_{T_R}\,,\quad c_w\triangleq\frac{1}{\rho_w}\frac{\partial\rho_w}{\partial p_w}\bigg|_{T_R}\approx\frac{1}{\rho_w}\frac{\partial\rho_w}{\partial p_o}\bigg|_{T_R}\,, \qquad (2.53, 2.54)$$

and the definition of rock compressibility

$$c_r\triangleq\frac{1}{\phi}\frac{\partial\phi}{\partial p_o}\,, \qquad (2.55)$$

allows us to express equations (2.49) in terms of p_o and S_w as follows:

$$
\begin{aligned}
&-\nabla\cdot\left\{\frac{\alpha\rho_w k_{rw}}{\mu_w}\vec{\mathbf{K}}\left[\left(\nabla p_o-\frac{\partial p_c}{\partial S_w}\nabla S_w\right)-\rho_w g\nabla d\right]\right\}+\\
&\alpha\rho_w\phi\left[S_w\left(c_w+c_r\right)\frac{\partial p_o}{\partial t}+\frac{\partial S_w}{\partial t}\right]-\alpha\rho_w q_w'''=0\,,\\
&-\nabla\cdot\left[\frac{\alpha\rho_o k_{ro}}{\mu_o}\vec{\mathbf{K}}\left(\nabla p_o-\rho_o g\nabla d\right)\right]+\\
&\alpha\rho_o\phi\left[\left(1-S_w\right)\left(c_o+c_r\right)\frac{\partial p_o}{\partial t}-\frac{\partial S_w}{\partial t}\right]-\alpha\rho_o q_o'''=0\,.
\end{aligned}
\qquad (2.56, 2.57)
$$

The diffusive effect of capillary pressure plays a role during displacement processes on a relatively small length scale (as e.g. in core flooding experiments). During water flooding on reservoir scale the dispersive effect of geological heterogeneities is usually much larger than the diffusive effect of capillary pressures. The correct way to take this dispersion into account is through the use of a velocity-dependent dispersion tensor; see Russell and Wheeler (1983). In addition to diffusion and dispersion caused by physical phenomena, artificial diffusion will occur as a result of the numerical solution of the discretized form of the equations. In many cases this numerical diffusion is of the same order of magnitude as or even larger than the physical diffusion and dispersion. At this point we will simply neglect capillary forces and dispersion all together. Equations (2.56) and (2.57) can then be simplified to:

$$
\begin{aligned}
&-\nabla\left[\frac{\alpha\rho_w k_{rw}}{\mu_w}\vec{\mathbf{K}}\left(\nabla p-\rho_w g\nabla d\right)\right]+\alpha\rho_w\phi\left[S_w\left(c_w+c_r\right)\frac{\partial p}{\partial t}+\frac{\partial S_w}{\partial t}\right]-\alpha\rho_w q_w'''=0\,,\\
&-\nabla\left[\frac{\alpha\rho_o k_{ro}}{\mu_o}\vec{\mathbf{K}}\left(\nabla p-\rho_o g\nabla d\right)\right]+\alpha\rho_o\phi\left[\left(1-S_w\right)\left(c_o+c_r\right)\frac{\partial p}{\partial t}-\frac{\partial S_w}{\partial t}\right]-\alpha\rho_o q_o'''=0\,,
\end{aligned}
\qquad (2.58, 2.59)
$$

where the subscript '*o*' has been dropped for the pressure because the absence of capillary pressure implies that $p_o = p_w$.

2.4.2 Nature of the equations

The nature of two-phase flow equations is discussed by e.g. Peaceman (1977), Aziz and Settari (1979), Ewing (1983) and Lake (1989). They illustrate that the pressure behavior is essentially diffusive, i.e. that the corresponding equations are parabolic and become elliptic in the limit of zero compressibility. The saturation behavior is mixed diffusive-convective, i.e. the corresponding equations are mixed parabolic-hyperbolic and become completely hyperbolic in the case of zero capillary pressure. This can be seen by rewriting the equations as follows. Consider equations (2.56) and (2.57) for 1-D flow through a conduit with constant cross-sectional area *A*, for small compressibilities such that we may assume that ρ is constant but *c* finite, in the absence of gravity terms and capillary pressure and source terms[†], and with isotropic permeability *k*:

[†] Absence of source terms corresponds to considering the (1-D) flow between an injector and a producer, in which case the flow is driven through the boundary conditions.

$$
\begin{aligned}
-\frac{\partial}{\partial x}\left(\lambda_w \frac{\partial p}{\partial x}\right)+\phi\left[S_w\left(c_w+c_r\right)\frac{\partial p}{\partial t}+\frac{\partial S_w}{\partial t}\right]_w &= 0,\\
-\frac{\partial}{\partial x}\left(\lambda_o \frac{\partial p}{\partial x}\right)+\phi\left[\left(1-S_w\right)\left(c_o+c_r\right)\frac{\partial p}{\partial t}-\frac{\partial S_w}{\partial t}\right] &= 0.
\end{aligned}
\qquad (2.60, 2.61)
$$

Here we introduced the water and oil mobilities

$$
\lambda_w \triangleq \frac{kk_{rw}\left(S_w\right)}{\mu_w} \quad \text{and} \quad \lambda_o \triangleq \frac{kk_{ro}\left(S_w\right)}{\mu_o} \ . \qquad (2.62, 2.63)
$$

Addition of equations (2.60) and (2.61) results in a PDE with only the pressure as primary variable[‡]:

$$
-\frac{\partial}{\partial x}\lambda_t \frac{\partial p}{\partial x}+\phi c_t \frac{\partial p}{\partial t}=0\,, \qquad (2.64)
$$

where the total mobility λ_t, and the total compressibility c_t have been defined as

$$
\lambda_t \triangleq \lambda_w+\lambda_o \ , \quad c_t \triangleq S_w c_w+\left(1-S_w\right)c_o+c_r \ . \qquad (2.65, 2.66)
$$

Equation (2.64) is a parabolic equation with non-constant coefficients. In the incompressible case the equation reduces to an elliptic equation:

$$
\frac{\partial}{\partial x}\lambda_t \frac{\partial p}{\partial x}=0\,. \qquad (2.67)
$$

Another equation, with only the water saturation as primary variable, can be obtained as follows. Neglecting gravity and considering 1-D flow, Darcy's law for water and oil, as given in equations (2.47) and (2.48), can be expressed as

$$
v_w=-\lambda_w \frac{\partial p}{\partial x} \ , \quad v_o=-\lambda_o \frac{\partial p}{\partial x}\,. \qquad (2.68, 2.69)
$$

Furthermore, we make use of the ratio

$$
f_w \triangleq \frac{v_w}{v_w+v_o} \equiv \frac{\lambda_w}{\lambda_w+\lambda_o} \ , \qquad (2.70)
$$

known as the water *fractional flow*, where $v_w + v_o$ represents the *total velocity* v_t. With the aid of these expressions, and realizing that f_w is a function[†] of S_w and therefore that $\partial f_w/\partial x=(\partial f_w/\partial S_w)(\partial S_w/\partial x)$, we can rewrite equations (2.60) and (2.61) as

$$
\begin{aligned}
v_t \frac{\partial f_w}{\partial S_w}\frac{\partial S_w}{\partial x}+\phi\left[S_w\left(c_w+c_r\right)\frac{\partial p}{\partial t}+\frac{\partial S_w}{\partial t}\right] &= 0\,,\\
-v_t \frac{\partial f_w}{\partial S_w}\frac{\partial S_w}{\partial x}+\phi\left[\left(1-S_w\right)\left(c_o+c_r\right)\frac{\partial p}{\partial t}-\frac{\partial S_w}{\partial t}\right] &= 0\,.
\end{aligned}
\qquad (2.71, 2.72)
$$

Subtraction of equations (2.71) and (2.72) after premultiplication with the appropriate factors allows us to eliminate the $\partial p/\partial t$ term and to obtain the required equation in terms of saturations only. In particular, for the incompressible case we find that:

[‡] The coefficients are still functions of saturation.

[†] $f_w(S_w)$ is sometimes referred to as the *flux function*.

$$v_t \frac{\partial f_w}{\partial S_w}\frac{\partial S_w}{\partial x} + \phi \frac{\partial S_w}{\partial t} = 0 \ . \tag{2.73}$$

Equation (2.73) is a first-order nonlinear hyperbolic equation, with a non-constant coefficient $v_t = v_w + v_o$ that depends on the pressure according to equations (2.68) and (2.69). In theory, the coupled equations (2.64) and (2.73) are therefore both nonlinear. However, because the changes in saturations usually occur much slower than the pressure changes[†], the nonlinearity in the pressure equation (2.64) is weak, and the equation may often be considered as a linear one with slowly-varying coefficients.

2.4.3 Relative permeabilities

The saturation-dependency of the relative permeabilities is usually determined from laboratory experiments where water is forced through a small *core* (a cylindrical piece of rock) initially saturated with oil. The values to be used in reservoir simulation are typically provided in the form of tables or simple mathematical expressions with parameters that have been fitted using the experimental results. Several of such expressions are known in the literature. Here we use the so-called Corey model given by

$$k_{rw} = k_{rw}^0 S^{n_w} \ , \quad k_{ro} = k_{ro}^0 \left(1-S\right)^{n_o} \ , \tag{2.74, 2.75}$$

where S is a scaled saturation defined as

$$S \triangleq \frac{S_w - S_{wc}}{1 - S_{or} - S_{wc}} \ , \ \ 0 \le S \le 1 \ , \tag{2.76}$$

k_{rw}^0 and k_{ro}^0 are the *end point relative permeabilities*, n_w and n_o are the *Corey exponents*, S_{wc} is the *connate water saturation* and S_{or} is the *residual oil saturation*. Note that the water fractional flow can also be expressed as

$$f_w = \frac{\lambda_w}{\lambda_w + \lambda_o} \equiv \frac{\dfrac{k_{rw}}{\mu_w}}{\dfrac{k_{rw}}{\mu_w} + \dfrac{k_{ro}}{\mu_o}} \equiv \frac{k_{rw}}{k_{rw} + k_{ro}\dfrac{\mu_w}{\mu_o}} \ . \tag{2.77}$$

Moreover, in the next section we will make use of the derivative df_w/dS_w which can, for the Corey model, be expressed analytically as

$$\frac{df_w}{dS_w} = \frac{\dfrac{dk_{rw}}{dS_w}}{k_{rw} + k_{ro}\dfrac{\mu_w}{\mu_o}} - \frac{k_{rw}\left(\dfrac{dk_{rw}}{dS_w} + \dfrac{dk_{ro}}{dS_w}\dfrac{\mu_w}{\mu_o}\right)}{\left(k_{rw} + k_{ro}\dfrac{\mu_w}{\mu_o}\right)^2} \ , \tag{2.78}$$

where

$$\frac{dk_{rw}}{dS_w} = \frac{k_{rw}^0 n_w S^{n_w - 1}}{1 - S_{wc} - S_{or}} \ , \quad \frac{dk_{ro}}{dS_w} = -\frac{k_{ro}^0 n_o \left(1-S\right)^{n_o - 1}}{1 - S_{wc} - S_{or}} \ . \tag{2.79, 2.80}$$

[†] In fact, the hyperbolic saturation equation in the form given in equation (2.73) is coupled to the elliptic pressure equation (2.67) because we assumed incompressible flow in its derivation. In that case the pressure changes are instantaneous.

2.4.4 Example 2 – Two-phase flow in a simple reservoir

Example 2 consists of the same six-block reservoir as in Example 1, but with two fluids, oil and water, instead of just oil. The additional reservoir and fluid properties have been specified in Table 2.3, and the corresponding relative permeabilities and the water fractional flow have been plotted in Figures 2.4 and 2.5. The water and oil compressibilities are equal and identical to the oil compressibility from Example 1, such that the total compressibility also remains unchanged. In this particular case the accumulation terms are therefore not a function of the saturations. The water viscosity is twice the oil viscosity. Moreover, the end-point permeability of water is two thirds of the end-point permeability of oil, such that the end-point water-oil mobility ratio is equal to one third, i.e. favorable. Figures 2.4 and 2.5 clearly display the strong saturation dependency of the relative permeabilities and the corresponding water fractional flow.

Table 2.3: Additional reservoir and fluid properties for Example 2.

Symbol	*Variable*	*Value*	*SI units*	*Value*	*Field units*
μ_w	Water dynamic viscosity	1.0×10^{-3}	Pa s	1.0	cP
k_{ro}^0	End point relative permeability, oil			0.9	–
k_{rw}^0	End point relative permeability, water			0.6	–
n_o	Corey exponent, oil			2.0	–
n_w	Corey exponent, water			2.0	–
S_{or}	Residual oil saturation			0.2	–
S_{wc}	Connate water saturation			0.2	–

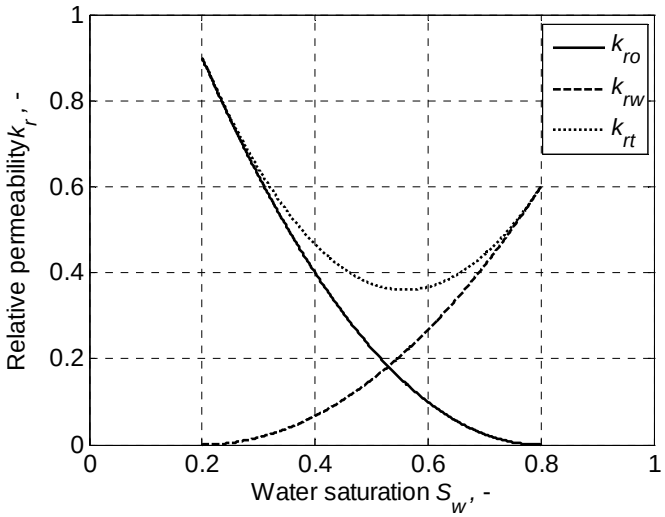


Figure 2.4: Relative permeabilities for Example 2.

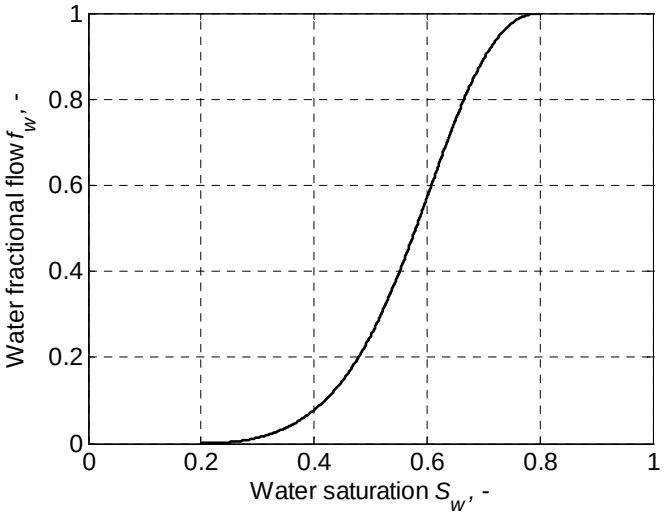


Figure 2.5: Water fractional flow for Example 2.

2.4.5 Buckley-Leverett equation*

Equation (2.73) is often referred to as the *Buckley-Leverett* equation after the authors who first presented and analysed it in the petroleum literature (Buckley and Leverett, 1942). It describes the one-dimensional saturation distribution of two incompressible immiscible fluids, neglecting the effects of gravity and capillary pressure. Without loss of generality consider a *core flooding* experiment with boundary and initial conditions given by

$$S_w\left(x,0\right) = S_{wc} \ , \tag{2.81}$$

$$S_w\left(0,t\right) = 1 - S_{or} \ , \tag{2.82}$$

representing a situation where the core is initially filled with oil except for a small fraction S_{wc} of connate water, whereafter water flooding takes place by injecting water at $x = 0$ such that the oil is replaced by water except for a small fraction S_{ro} of residual oil. As for all hyperbolic equations (which typically describe wave propagation problems) it is possible to find *characteristics*, i.e. relationships between x and t for which the dependent variables do not change. In our case of a single dependent variable S_w this means that the total derivative dS_w/dt should remain equal to zero:

$$\frac{dS_w}{dt} \equiv \frac{\partial S_w}{\partial t} + \frac{\partial S_w}{\partial x}\frac{\partial x}{\partial t} = 0 \ . \tag{2.83}$$

For a given saturation $S_w = \hat{S}_w$ we can therefore write

$$\left.\frac{dx}{dt}\right|_{S_w=\hat{S}_w} = \frac{\left.\dfrac{\partial S_w}{\partial t}\right|_{S_w=\hat{S}_w}}{\left.\dfrac{\partial S_w}{\partial x}\right|_{S_w=\hat{S}_w}} \ , \tag{2.84}$$

and combination of equations (2.73) and (2.84) then gives

$$\left.\frac{dx}{dt}\right|_{S_w=\hat{S}_w} = \frac{v_t}{\phi}\left.\frac{df_w}{dS_w}\right|_{S_w=\hat{S}_w} \ . \tag{2.85}$$

The position of the point where $S_w = \hat{S}_w$ follows by integrating equation (2.85) resulting in

$$x\big|_{S_w=\hat{S}_w} = \frac{v_t t}{\phi} \frac{df_w}{dS_w}\bigg|_{S_w=\hat{S}_w} , \tag{2.86}$$

where the integration constant has been set equal to zero which implies that $x = 0$ at $t = 0$. If the core has length L and cross sectional area A, it is convenient to rescale equation (2.86) as

$$x_D\big|_{S_w=\hat{S}_w} = t_D \frac{df_w}{dS_w}\bigg|_{S_w=\hat{S}_w} , \tag{2.87}$$

which leads to the dimensionless Buckley-Leverett velocity

$$v_D\big|_{S_w=\hat{S}_w} = \frac{df_w}{dS_w}\bigg|_{S_w=\hat{S}_w} . \tag{2.88}$$

Here the dimensionless length and time are defined as

$$x_D \triangleq \frac{x}{L} , \tag{2.89}$$

$$t_D \triangleq \frac{A v_t t}{AL\phi} \equiv \frac{q_t t}{V_p} , \tag{2.90}$$

where V_p is the pore volume of the core. Figure 2.6 displays the derivative df_w/dS_w as a function of S_w, and Figure 2.7 the corresponding Buckley-Leverett solution (2.87) at dimensionless time $t_D = 0.2$, i.e. after injection of 20% of the pore volume or $(1 - S_{or} - S_{wc}) \times 20\% = 33.3\%$ of the mobile pore volume.

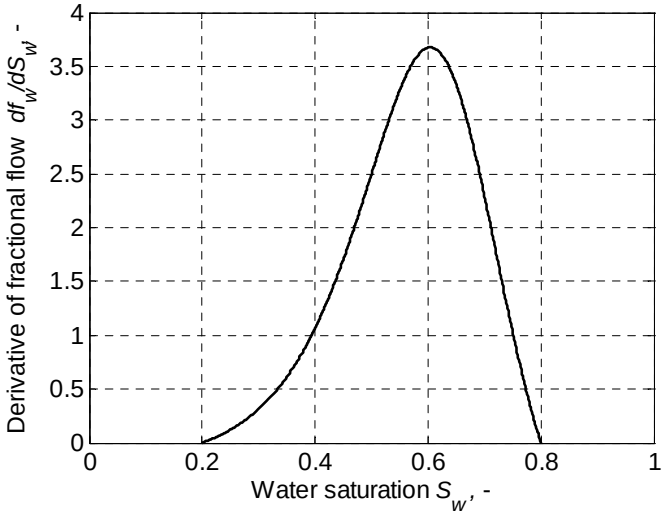


Figure 2.6: Derivative of the water fractional flow for Example 2.

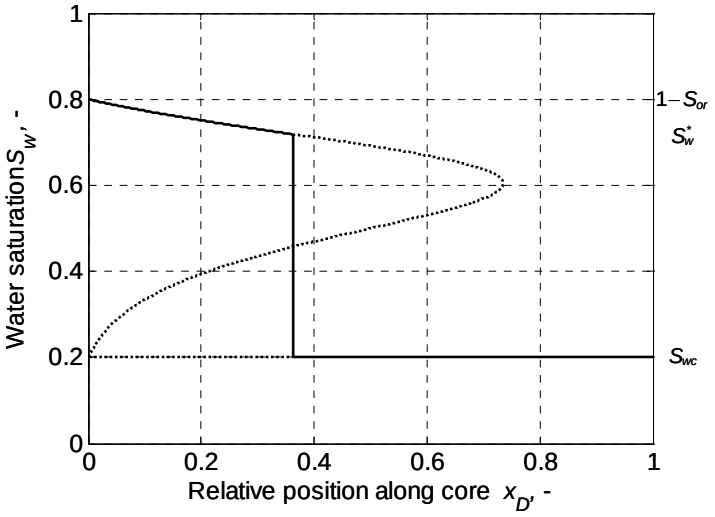


Figure 2.7: Buckley–Leverett solution (dotted line) and shock solution (solid line) corresponding to $t_D = 0.2$ *or 33.3% mobile pore volume injected.*

Figure 2.7 illustrates that the Buckley-Leverett solution predicts the existence of three values of S_w over a large part of the core (for the example of $t_D = 0.2$ this occurs for values of x_D below 0.73), a situation that is clearly physically impossible. The unphysical solutions originate from neglecting the effect of capillary pressure which in reality produces a sharp increase in water saturation at a value of x_D somewhere in the triple value region. This effect can be approximated in the form of a step-wise increase of saturation known as a *shock* in the theory of hyperbolic differential equations. The magnitude and position of the shock follow from requiring that mass is conserved in a control volume around the shock[†]. If the shock of magnitude $\Delta S_w = S_w^* - S_{wc}$ moves a distance Δx_D in a time interval Δt_D it can be derived that the *shock velocity* v_D^* should obey (see e.g. Lake (1989) for details)

$$v_D^* = \frac{\Delta x_D}{\Delta t_D} \equiv \frac{f\left(S_w^*\right)}{S_w^* - S_{wc}} \ , \tag{2.91}$$

where we used a superscripted star to indicate the variables at shock conditions. However, the velocity should also satisfy equation (2.88), and we may therefore equate expressions (2.88) and (2.91) which leads to a condition for the shock saturation S_w^* in the form of

$$\left.\frac{df_w}{dS_w}\right|_{S_w = S_w^*} = \frac{f\left(S_w^*\right)}{S_w^* - S_{wc}} \ . \tag{2.92}$$

Usually equation (2.92) cannot be solved explicitly for S_w^*, and requires an iterative numerical solution[‡]. The solid line in Figure 2.7 displays the saturation profile along the core taking into account the shock formation. Combining equations (2.87) and (2.91) we can express the full solution as

---

[†] We may either consider the water mass or the oil mass. Moreover, because we assume the fluids to be incompressible, it is sufficient to consider a volume balance rather than a mass balance.

[‡] As was first shown in Welge (1952), equation (2.92) implies that the tangent at f_w in S_w^* and the secant from S_{wc} to S_w^* are identical, leading to a simple graphical solution procedure, known as the *Welge method*, which was popular before the advent of computers.

$$x_D\left(S_w, t_D\right) = \begin{cases} \dfrac{df_w}{dS_w} t_D, & S_w^* \le S_w \le 1 - S_{or} \\ v_D^* t_D, & S_{wc} \le S_w \le S_w^* \end{cases} . \tag{2.93}$$

This kind of analysis, known as *fractional flow theory* or the *method of characteristics*, has been successfully extended to multiple components and thermal behaviour, see e.g. Lake (1989), but is typically restricted to one-dimensional flow. Moreover, in reality there will always be dispersive effects caused by capillary forces, compressibility and reservoir hererogeneities, and true shocks will therefore never be present. However, sharp saturation increases do certainly occur and the underlying characteristic hyperbolic behaviour of the saturation equation is an important feature of multiphase flow in porous media.

2.4.6 Linear approximation*

In the special case of linear relative permeabilities with end values equal to one and residual saturations equal to zero we have

$$k_{rw}\left(S_w\right) = S_w \quad \text{and} \quad k_{ro}\left(S_w\right) = 1 - S_w \ . \tag{2.94, 2.95}$$

If, in addition, we take

$$\mu \triangleq \mu_o = \mu_w \ , \tag{2.96}$$

we have

$$\lambda \triangleq \frac{k}{\mu} \quad \text{and} \quad f_w = S_w \ . \tag{2.97, 2.98}$$

such that we can rewrite equations (2.67) and (2.73) as

$$\frac{\partial}{\partial x} \lambda \frac{\partial p}{\partial x} = 0 \, , \tag{2.99}$$

$$v \frac{\partial S_w}{\partial x} + \phi \frac{\partial S_w}{\partial t} = 0 \ , \tag{2.100}$$

which are linear elliptic and hyperbolic (convection) equations with spatially varying coefficients, with

$$v = -\lambda \frac{\partial p}{\partial x} \, . \tag{2.101, 2.102}$$

If an additional diffusion term is introduced in equation (2.100) we obtain a linear convection-diffusion equation

$$-D \frac{\partial^2 S_w}{\partial x^2} + v \frac{\partial S_w}{\partial x} + \phi \frac{\partial S_w}{\partial t} = 0 \ , \tag{2.103}$$

where D is the diffusion constant. In this case equations (2.99) and (2.103) (or (2.100)) can be interpreted as describing the flow of two incompressible *miscible* fluids with identical properties such as water with two different colors (sometimes referred to as a *blue and red water* situation). Alternatively, the equations can be interpreted to describe the flow of *immiscible* fluids, in which case D represents the effect of dispersion due to geological heterogeneities.

2.4.7 Formation volume factors*

In our derivation we used equations of state (2.53) and (2.54) to relate pressure, temperature and densities of the reservoir fluids. These equations of state can also be expressed as relationships between pressure, temperature and volumes; e.g. for the oil we can write, instead of equation (2.53):

$$c_o \triangleq -\frac{1}{V_o}\frac{\partial V_o}{\partial p_o}\bigg|_{T_R} . \tag{2.104}$$

In many practical reservoir engineering applications, the fluid densities and volumes at reservoir conditions are expressed in terms of those at *standard conditions*† with the aid of a *formation volume factor*. In particular gas and to a lesser extent oil change volume when flowing from the reservoir to surface. The oil formation volume factor B_o is defined as the ratio of a unit volume of oil at downhole conditions p and T, including dissolved gas, and the volume it occupies after it has been transferred to standard conditions p_{sc} and T_{sc}, during which journey gas has escaped from the oil:

$$B_o \triangleq \frac{V_o\big|_{p,T}}{V_o\big|_{p_{sc},T_{sc}}} . \tag{2.105}$$

An equivalent definition holds for the water formation volume factor, although usually very little gas dissolves in water. The gas formation volume factor is defined as the ratio of a unit volume of gas at downhole conditions, and the volume it occupies after it has been transferred to standard conditions, during which journey possibly some liquid drop-out may have occurred. Gas originally at reservoir conditions dramatically expands when the pressure approaches standard conditions, even if it looses some liquids, and the therefore B_g is typically much smaller than one. Oppositely, oil at reservoir conditions always contains a large amount of dissolved gas which escapes from the oil when transferred to standard conditions, and therefore B_o is always larger than one. Water hardly changes volume, compared to oil, and therefore B_w is always close to one. If we substitute equation (2.105) in equation (2.104), taking $p = p_o$, and $T = T_R$, it follows that

$$c_o \triangleq -\frac{1}{B_o}\frac{\partial B_o}{\partial p_o}\bigg|_{T_R} , \tag{2.106}$$

and similar expressions can be obtained for gas and water. Formation volume factors change with pressure and temperature, and therefore also the fluid compressibilities are functions of pressure and temperature. Even if we take the temperature constant at its reservoir value, the pressure dependence of the fluid compressibilities is often large enough to take them into account, thus introducing an additional nonlinearity in the reservoir simulation equations. This hold especially for gas reservoirs, and to a lesser extent for oil reservoirs that experience large pressure changes, e.g. during primary recovery. Determination of the pressure and temperature dependency of formation volume factors is usually done with the aid of laboratory experiments on fluid samples taken from exploration wells. In absence of samples,

† In the E&P industry standard conditions are usually defined as a pressure p_{sc} = 100 kPa (14.7 psi) and a temperature T_{sc} = 15 °C (60 °F), which can be considered as typical for atmospheric conditions in temperate climates. Oil at standard conditions is often referred to as *stock tank oil*.

a large number of correlations available from literature can be used to estimate the pressure and temperature dependence of the reservoir fluids; see e.g. Whitson and Brulé (2002). Most of the reservoir engineering literature traditionally uses pressure-dependent formation volume factors rather than pressure-dependent fluid compressibilities. For liquid flow, and as long as the pressure changes in the reservoir remain small compared to the total pressure, the nonlinearity of the oil compressibility remains small, while the water compressibility remains as good as constant, a situation that applies to all the examples that we use in this text. To keep the equations as simple as possible, we have therefore chosen not to use formation volume factors in our derivations, but use (constant) compressibilities instead.

2.4.8 Finite difference discretization

This section gives a brief overview of the semi-discretization of equations (2.58) with the aid of finite differences for 2-D flow. For details and for alternative discretization schemes we refer to the references mentioned in Section 2.1. Following the same approach as used for one-phase flow, we rewrite equation (2.58) and (2.59) in scalar 2-D form, assuming isotropic permeability, pressure independence of the parameters, and absence of gravity forces[†]:

$$-\frac{h}{\mu_w}\left[\frac{\partial}{\partial x}\left(kk_{rw}\frac{\partial p}{\partial x}\right)+\frac{\partial}{\partial y}\left(kk_{rw}\frac{\partial p}{\partial y}\right)\right]+h\left[\phi S_w\left(c_w+c_r\right)\frac{\partial p}{\partial t}+\phi\frac{\partial S_w}{\partial t}\right]-hq_w'''=0\,,$$
$$-\frac{h}{\mu_o}\left[\frac{\partial}{\partial x}\left(kk_{ro}\frac{\partial p}{\partial x}\right)+\frac{\partial}{\partial y}\left(kk_{ro}\frac{\partial p}{\partial y}\right)\right]+h\left[\phi\left(1-S_w\right)\left(c_o+c_r\right)\frac{\partial p}{\partial t}-\phi\frac{\partial S_w}{\partial t}\right]-hq_o'''=0\,. \tag{2.107, 2.108}$$

The first term in equation (2.107) can be rewritten as

$$\frac{h}{\mu_w}\frac{\partial}{\partial x}\left(kk_{rw}\frac{\partial p}{\partial x}\right)\approx\frac{h}{\mu_w}\frac{\Delta}{\Delta x}\left(kk_{rw}\frac{\Delta p}{\Delta x}\right)=\frac{h}{\mu_w}\frac{\left(kk_{rw}\right)_{i+\frac{1}{2},j}\left(p_{i+1,j}-p_{i,j}\right)-\left(kk_{rw}\right)_{i-\frac{1}{2},j}\left(p_{i,j}-p_{i-1,j}\right)}{\left(\Delta x\right)^2}\,, \tag{2.109}$$

where the absolute permeabilities k are based on harmonic averages just as in the single-phase case; see equation (2.28). However, the relative permeabilities k_{rw} need to be determined through upstream weighting to obtain the correct convective behavior; see Aziz and Settari (1979), p. 153. This implies that

$$\left(k_{rw}\right)_{i+\frac{1}{2},j}\triangleq\begin{cases}\left(k_{rw}\right)_{i,j} & \text{if} \quad p_{i,j}\geq p_{i+1,j}\\ \left(k_{rw}\right)_{i+1,j} & \text{if} \quad p_{i,j}<p_{i+1,j}\end{cases}\,, \tag{2.110}$$

The second term in equation (2.107) can be rewritten in a similar fashion. Combining and reorganizing all terms results in

[†] To stay in line with the notation used in the single-phase flow case, we should have used ϕ_0, $\rho_{o,0}$ and $\rho_{w,0}$ to indicate the pressure-independence of these parameters, but we have dropped the subscripts 0 to simplify the notation.

$$V\left[\phi S_w\left(c_w+c_r\right)\frac{\partial p}{\partial t}+\phi\frac{\partial S_w}{\partial t}\right]_{i,j}-\left(T_w\right)_{i-\frac{1}{2},j}p_{i-1,j}-\left(T_w\right)_{i,j-\frac{1}{2}}p_{i,j-1}+$$
$$\left[\left(T_w\right)_{i-\frac{1}{2},j}+\left(T_w\right)_{i,j-\frac{1}{2}}+\left(T_w\right)_{i,j+\frac{1}{2}}+\left(T_w\right)_{i+\frac{1}{2},j}\right]p_{i,j}-\left(T_w\right)_{i,j+\frac{1}{2}}p_{i,j+1}-\left(T_w\right)_{i+\frac{1}{2},j}p_{i+1,j}=V\left(q_w'''\right)_{i,j}\ , \tag{2.111}$$

where the transmissibilities are now defined as

$$\left(T_w\right)_{i-\frac{1}{2},j}\triangleq\frac{\Delta y}{\Delta x}\frac{h}{\mu_w}\left(kk_{rw}\right)_{i-\frac{1}{2},j}\text{, etc.} \tag{2.112}$$

A similar discretization can be obtained for equation (2.108):

$$V\left[\phi\left(1-S_w\right)\left(c_o+c_r\right)\frac{\partial p}{\partial t}-\phi\frac{\partial S_w}{\partial t}\right]_{i,j}-\left(T_o\right)_{i-\frac{1}{2},j}p_{i-1,j}-\left(T_o\right)_{i,j-\frac{1}{2}}p_{i,j-1}+$$
$$\left[\left(T_o\right)_{i-\frac{1}{2},j}+\left(T_o\right)_{i,j-\frac{1}{2}}+\left(T_o\right)_{i,j+\frac{1}{2}}+\left(T_o\right)_{i+\frac{1}{2},j}\right]p_{i,j}-\left(T_o\right)_{i,j+\frac{1}{2}}p_{i,j+1}-\left(T_o\right)_{i+\frac{1}{2},j}p_{i+1,j}=V\left(q_o'''\right)_{i,j}\ , \tag{2.113}$$

Equations (2.111) and (2.113) can be written in matrix form as

$$\begin{bmatrix}\mathbf{V}_{wp} & \mathbf{V}_{ws}\\ \mathbf{V}_{op} & \mathbf{V}_{os}\end{bmatrix}\begin{bmatrix}\dot{\mathbf{p}}\\ \dot{\mathbf{s}}\end{bmatrix}+\begin{bmatrix}\mathbf{T}_w & \mathbf{0}\\ \mathbf{T}_o & \mathbf{0}\end{bmatrix}\begin{bmatrix}\mathbf{p}\\ \mathbf{s}\end{bmatrix}=\begin{bmatrix}\mathbf{q}_w\\ \mathbf{q}_o\end{bmatrix}, \tag{2.114}$$

where the vectors **p** and **s** contain pressures and water saturations:

$$\mathbf{p}^T\triangleq\begin{bmatrix}p_{i,j-1} & \cdots & p_{i-1,j} & p_{i,j} & p_{i+1,j} & \cdots & p_{i,j+1}\end{bmatrix}, \tag{2.115}$$

$$\mathbf{s}^T\triangleq\begin{bmatrix}\left(S_w\right)_{i,j-1} & \cdots & \left(S_w\right)_{i-1,j} & \left(S_w\right)_{i,j} & \left(S_w\right)_{i+1,j} & \cdots & \left(S_w\right)_{i,j+1}\end{bmatrix}, \tag{2.116}$$

and where the sub-matrices $\mathbf{V}_{wp}$, $\mathbf{V}_{ws}$, $\mathbf{V}_{op}$ and $\mathbf{V}_{os}$ contain accumulation terms†

$$\mathbf{V}_{wp}\triangleq V\left(c_w+c_r\right)\begin{bmatrix}0 & \cdots & 0 & \phi_{i,j}\times\left(S_w\right)_{i,j} & 0 & \cdots & 0\end{bmatrix};\ \mathbf{V}_{ws}\triangleq V\begin{bmatrix}0 & \cdots & 0 & \phi_{i,j} & 0 & \cdots & 0\end{bmatrix}, \tag{2.117, 2.118}$$

$$\mathbf{V}_{op}\triangleq V\left(c_o+c_r\right)\begin{bmatrix}0 & \cdots & 0 & \phi_{i,j}\times\left(1-S_w\right)_{i,j} & 0 & \cdots & 0\end{bmatrix};\ \mathbf{V}_{os}\triangleq -V\begin{bmatrix}0 & \cdots & 0 & \phi_{i,j} & 0 & \cdots & 0\end{bmatrix}, \tag{2.119), (2.120}$$

the sub-matrices $\mathbf{T}_w$ and $\mathbf{T}_o$ contain transmissibility terms

$$\mathbf{T}_w\triangleq\begin{bmatrix}-\left(T_w\right)_{i,j-\frac{1}{2}} & \cdots & -\left(T_w\right)_{i-\frac{1}{2},j} & \left(\left(T_w\right)_{i,j-\frac{1}{2}}+\left(T_w\right)_{i-\frac{1}{2},j}+\left(T_w\right)_{i+\frac{1}{2},j}+\left(T_w\right)_{i,j+\frac{1}{2}}\right) & -\left(T_w\right)_{i+\frac{1}{2},j} & \cdots & -\left(T_w\right)_{i,j+\frac{1}{2}}\end{bmatrix}, \tag{2.121}$$

$$\mathbf{T}_o\triangleq\begin{bmatrix}-\left(T_o\right)_{i,j-\frac{1}{2}} & \cdots & -\left(T_o\right)_{i-\frac{1}{2},j} & \left(\left(T_o\right)_{i,j-\frac{1}{2}}+\left(T_o\right)_{i-\frac{1}{2},j}+\left(T_o\right)_{i+\frac{1}{2},j}+\left(T_o\right)_{i,j+\frac{1}{2}}\right) & -\left(T_o\right)_{i+\frac{1}{2},j} & \cdots & -\left(T_o\right)_{i,j+\frac{1}{2}}\end{bmatrix}, \tag{2.122}$$

and vectors $\mathbf{q}_w$ and $\mathbf{q}_o$ contain the flow rates (source terms) with elements expressed in m^3/s:

† Here the sub-matrices are displayed as vectors, which form, however, building blocks for matrices when the equations for multiple grid blocks are combined.

$$\mathbf{q}_w^T \triangleq \begin{bmatrix} \cdots & (q_w)_{i,j} & \cdots \end{bmatrix}, \tag{2.123}$$

$$\mathbf{q}_o^T \triangleq \begin{bmatrix} \cdots & (q_o)_{i,j} & \cdots \end{bmatrix}. \tag{2.124}$$

2.4.9 Example 3 – Inverted five-spot

Example 3 concerns a square reservoir with heterogeneous permeability and porosity fields as depicted in Figure 2.3, modeled with $21 \times 21 = 441$ grid blocks. It can be seen that the permeability displays a marked streak running from the South-West (bottom left) corner to just below the North-East (top right) corner, and that the porosity is mildly correlated with the permeability. The other relevant parameters have been listed in Table 2.4. As in the earlier examples, gravity and capillary forces are neglected.

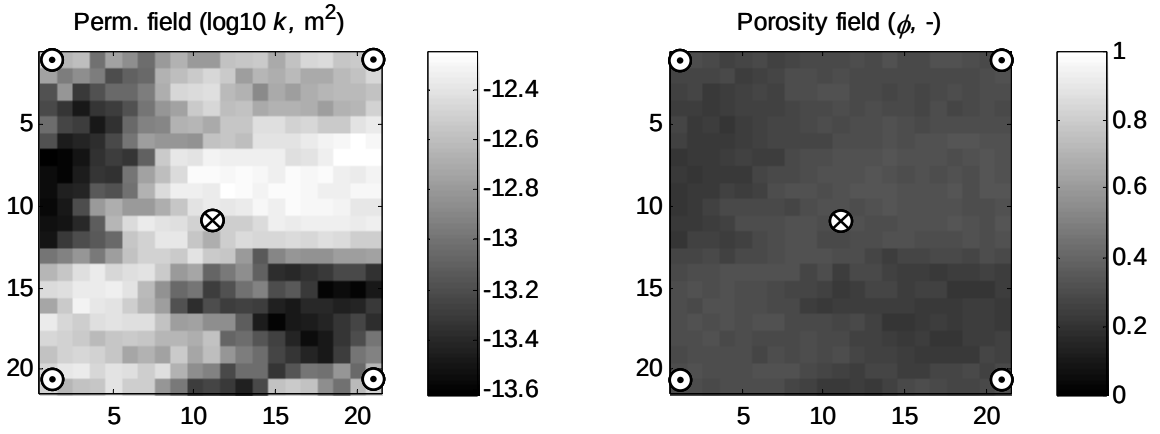


Figure 2.3: Well configuration, and permeability and porosity fields for Example 3. The reservoir is produced with a central injector and five producers in the corners.

Table 2.4: Reservoir and fluid properties for Example 3.

Symbol	Variable	Value	SI units	Value	Field units
h	Grid block height	2	m	6.56	ft
Δx, Δy	Grid block length/width	33.33	m	109.36	ft
μ_o	Oil dynamic viscosity	5.0×10^{-4}	Pa s	0.5	cP
μ_w	Water dynamic viscosity	1.0×10^{-3}	Pa s	1.0	cP
c_t	Total compressibility	3.0×10^{-9}	Pa^{-1}	2.1×10^{-5}	psi^{-1}
$\breve{p}_R$	Initial reservoir pressure	30×10^6	Pa	4351.1	psi
r_{well}	Well bore radius	0.114	m	4.50	in
k_{ro}^0	End point relative permeability, oil			0.9	–
k_{rw}^0	End point relative permeability, water			0.6	–
n_o	Corey exponent, oil			2.0	–
n_w	Corey exponent, water			2.0	–
S_{or}	Residual oil saturation			0.2	–
S_{wc}	Connate water saturation			0.2	–

Figure 2.4. illustrates the sparse structure of the secant matrices for Example 3. The top-left figure corresponds to the accumulation matrix **V** which has a perfect diagonal structure in each of its four quadrants. The top-right figure corresponds to the transmissbility matrix **T** which has an almost penta-diagonal structure (tri-diagonal with 'holes' and two side bands) in the two left quadrants. The two right quadrants are completely filled with zeros because we have neglected capillary forces; see also equation (3.134). The bottom-left and bottom-right figures display details of **T** and illustrate the side bands and the 'holes' in the tri-diagonals at every 21st row which are typical for a regular numbering scheme. In this case the 441 grid blocks have been numbered row-wise from top-left to bottom-right. A grid block i in the center of the grid, i.e. not at an edge or at a corner, is connected to its Western and Eastern neighbors with numbers i–1 and i+1 respectively, which results in the tri-diagonals, and to its Northern and Southern neighbors i–21 and i+21 respectively which results in the penta-diagonal side bands. Grid blocks at an edge are missing one of the connections which results in irregularities in the structure in the form of 'holes' in the tri-diagonals.

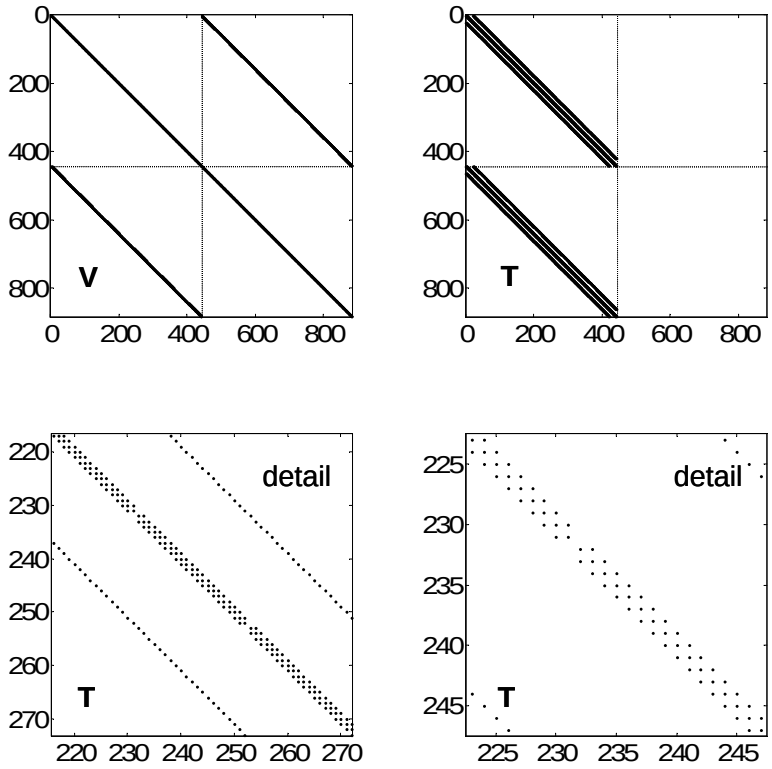


Figure 2.4: System matrices for Example 3.

2.4.10 Sources of nonlinearity

Although we use a matrix-vector notation, which suggests a linear system of equations, equations (2.114) are nonlinear because the coefficients of sub-matrices $\mathbf{V}_{wp}$, $\mathbf{V}_{op}$ $\mathbf{T}_w$ and $\mathbf{T}_o$ are functions of S_w. In particular, the coefficients of the transmissibility matrices $\mathbf{T}_w$ and $\mathbf{T}_o$ contain the saturation-dependent relative permeabilities, for which we use the Corey model given by equations (2.74) to (2.76) Actually, the coefficients of the transmissibility matrices are also functions of p because of the upstream weighting of the relative permeabilities. That is, if the pressures in two adjacent grid blocks change slightly, but such that the flow through the grid block boundary changes direction, the upstream relative permeability and therefore the transmissibility may change strongly. This nonlinear effect may cause problems during the iterative solution of the system equations during implicit time integration, if the flow direction keeps changing during subsequent iterations. However, because it is a discontinuous nonlinearity, we can not differentiate the transmissibilities with respect to p, and we can not take it into account during linearization of the equations as required for e.g. implicit integration with Newton-Raphson iteration. Another source of nonlinearity are the source terms $\mathbf{q}_o$ and $\mathbf{q}_w$ in equation (2.114) which cannot always be prescribed directly. In the case of a water injection well, the oil flow rates are equal to zero, and it is possible to prescribe the water injection rates. In a production well, however, the proportions of oil and water in the total flow rate q_t depend on the fractional flows f_o and f_w, , i.e. on the relative magnitude of the oil and water mobilities around the well according to

$$q_o = f_o q_t = \frac{\lambda_o}{\lambda_o + \lambda_w} q_t \ , \quad q_w = f_w q_t = \frac{\lambda_w}{\lambda_o + \lambda_w} q_t \, , \qquad (2.125, 2.126)$$

where the saturation-dependent mobilities λ_o and λ_w are given by equations (2.62) and (2.63). Therefore we need to specify

$$\underbrace{\begin{bmatrix}\mathbf{q}_w\\ \mathbf{q}_o\end{bmatrix}}_{\mathbf{q}} = \underbrace{\begin{bmatrix}\mathbf{F}_w(\mathbf{s})\\ \mathbf{F}_o(\mathbf{s})\end{bmatrix}}_{\mathbf{F}}\mathbf{q}_t\,, \tag{2.127}$$

where $\mathbf{F}_o$ and $\mathbf{F}_w$ are diagonal matrices of which the non-zero entries contain fractional flows:

$$\mathbf{F}_w \triangleq \begin{bmatrix}0 & \cdots & 0 & (f_w)_{i,j} & 0 & \cdots & 0\end{bmatrix};\ \mathbf{F}_o \triangleq \begin{bmatrix}0 & \cdots & 0 & (f_o)_{i,j} & 0 & \cdots & 0\end{bmatrix}. \tag{2.128, 2.129}$$

To emphasize the nonlinearities, equation (2.114) may therefore be rewritten as[†]

$$\begin{bmatrix}\mathbf{V}_{wp}(\mathbf{s}) & \mathbf{V}_{ws}\\ \mathbf{V}_{op}(\mathbf{s}) & \mathbf{V}_{os}\end{bmatrix}\begin{bmatrix}\dot{\mathbf{p}}\\ \dot{\mathbf{s}}\end{bmatrix}+\begin{bmatrix}\mathbf{T}_w(\mathbf{s}) & \mathbf{0}\\ \mathbf{T}_o(\mathbf{s}) & \mathbf{0}\end{bmatrix}\begin{bmatrix}\mathbf{p}\\ \mathbf{s}\end{bmatrix}=\begin{bmatrix}\mathbf{F}_w(\mathbf{s})\\ \mathbf{F}_o(\mathbf{s})\end{bmatrix}\mathbf{q}_t\ . \tag{2.130}$$

In an injection well we have $\mathbf{q}_t = \mathbf{q}_w$, and we expect that soon after injection has started the fractional flows for water and oil close to the injection will approach one and zero respectively. However, before injection starts, the initial condition for the saturation is usually equal to the connate water saturation, which means that the fractional flows for water and oil are zero and one respectively, which implies that it is impossible to ever inject water. This paradox, which illustrates a shortcoming of the relative permeability concept, is usually circumvented by simply specifying a fractional flow equal to one for every injection well.

2.4.11 Incompressible flow

In the special case that the fluid and rock compressibilities are so small that they may be neglected, it follows from equations (2.117) and (2.119) that $\mathbf{V}_{wp} = \mathbf{V}_{op} = \mathbf{0}$. In that case we can rewrite equation (2.130) as

$$\begin{bmatrix}\mathbf{0} & \mathbf{V}_{ws}\\ \mathbf{0} & \mathbf{V}_{os}\end{bmatrix}\begin{bmatrix}\dot{\mathbf{p}}\\ \dot{\mathbf{s}}\end{bmatrix}+\begin{bmatrix}\mathbf{T}_w(\mathbf{s}) & \mathbf{0}\\ \mathbf{T}_o(\mathbf{s}) & \mathbf{0}\end{bmatrix}\begin{bmatrix}\mathbf{p}\\ \mathbf{s}\end{bmatrix}=\begin{bmatrix}\mathbf{F}_w(\mathbf{s})\\ \mathbf{F}_o(\mathbf{s})\end{bmatrix}\mathbf{q}_t\,, \tag{2.131}$$

which may also be expressed as:

$$\begin{aligned}\mathbf{V}_{ws}\dot{\mathbf{s}}+\mathbf{T}_w(\mathbf{s})\mathbf{p}&=\mathbf{F}_w(\mathbf{s})\mathbf{q}_t\ ,\\ \mathbf{V}_{os}\dot{\mathbf{s}}+\mathbf{T}_o(\mathbf{s})\mathbf{p}&=\mathbf{F}_o(\mathbf{s})\mathbf{q}_t\ .\end{aligned} \tag{2.132, 2.133}$$

Because $\mathbf{V}_{ws} = -\mathbf{V}_{os}$ we can add the two equations to obtain the pressure equation for incompressible flow

$$\mathbf{T}_t(\mathbf{s})\mathbf{p} = \mathbf{q}_t\ , \tag{2.134}$$

where we used the equality $\mathbf{F}_w + \mathbf{F}_o = \mathbf{I}$, and where $\mathbf{T}_t = \mathbf{T}_w + \mathbf{T}_o$ is the total transmissibility matrix , which is still a function of saturation. Note that the pressure equation is no longer a differential equation but has degenerated to an algebraic equation. The physical background is that the vanishing of compressibilities means that there is no longer a possibility to store energy in the system. Just as in the case of single-phase flow, discussed in Section 0, it

[†] As discussed before, we disregard the dependency of the transmissibility terms on pressure.

appears as if equation (2.134) can simply be solved for the constant pressure **p**. However, the total transmissibility matrix $\mathbf{T}_t$ is singular which implies that we cannot directly solve the equation. Just as in the single-phase case, the singularity can be removed through prescribing the pressure in at least one of the grid blocks, or through the use of a well model, in which case we may indeed solve for **p**. Thereafter, one of the two equations (2.132) or (2.133) can be used to compute the water saturations.

2.4.12 Fluid velocities*

*Total velocity**

Darcy's law, which specifies an empirical relationship between pressure gradients and fluid velocities, is at the heart of the description of flow through porous media. However, in our formulation of the flow equations in the previous sections we rapidly lost the fluid velocities as variables, through substitution in the mass balance equations; see equations (2.4), (2.7) and (2.10) for the single-phase flow case, and equations (2.45) to (2.50) for the two-phase case. To recover the fluid velocities, after solving the flow equations, we have to revert to Darcy's law. Often we are interested in the *total velocity* which is the sum of the phase velocities. In the case of two-phase oil-water flow, Darcy's law for the total velocity is obtained by adding equations (2.47) and (2.48):

$$\vec{\mathbf{v}}_t = -\frac{k_{rw}}{\mu_w}\vec{\mathbf{K}}\left(\nabla p_w - \rho_w g \nabla d\right) - \frac{k_{ro}}{\mu_o}\vec{\mathbf{K}}\left(\nabla p_o - \rho_o g \nabla d\right) \tag{2.135}$$

If, as before, we neglect gravity and capillary forces, assume isotropic permeability, and express the equations in scalar form we obtain

$$\begin{aligned} v_{t,x} &= -k\left(\frac{k_{rw}}{\mu_w} + \frac{k_{ro}}{\mu_o}\right)\frac{\partial p}{\partial x}, \\ v_{t,y} &= -k\left(\frac{k_{rw}}{\mu_w} + \frac{k_{ro}}{\mu_o}\right)\frac{\partial p}{\partial y}. \end{aligned} \tag{2.136, 2.137}$$

Starting from these equations we can now use a finite difference discretization to find numerical approximations of the velocity components:

$$\begin{aligned} v_{t,x} \approx \left(v_t\right)_{i+\frac{1}{2},j} &\triangleq -\left(\frac{kk_{rw}}{\mu_w} + \frac{kk_{ro}}{\mu_o}\right)_{i+\frac{1}{2},j} \frac{p_{i+1,j} - p_{i,j}}{\Delta x}, \\ v_{t,y} \approx \left(v_t\right)_{i,j+\frac{1}{2}} &\triangleq -\left(\frac{kk_{rw}}{\mu_w} + \frac{kk_{ro}}{\mu_o}\right)_{i,j+\frac{1}{2}} \frac{p_{i,j+1} - p_{i,j}}{\Delta y}. \end{aligned} \tag{2.138, 2.139}$$

In these discretized equations the pressures are taken in the grid block centers whereas the velocities and the (averaged) parameters are taken at the grid block boundaries. Note that equations (2.138) and (2.139) represent the velocities at the left and bottom boundaries of a grid block. Similar expressions can be obtained for the right and top boundaries.

*Velocities at grid block boundaries**

In the case of tracing streamlines, as discussed in Section 3.4.5, we need the velocities at the grid block boundaries which can be obtained from equations (2.138) and (2.139). Alternatively, we can use the following, slightly different, approach that directly exploits the

connectivity structure of the grid blocks. Consider a block-centered finite-difference reservoir model with n_{gb} grid blocks and n_{con} grid block connectivities. If we represent the grid block pressures by an $n_{gb} \times 1$ vector $\mathbf{p}$, we can define a linear transformation

$$\boldsymbol{p} = \mathbf{L}_{\boldsymbol{p}\mathbf{p}}\mathbf{p} \ , \tag{2.140}$$

where $\boldsymbol{p}$ is an $n_{con} \times 1$ vector of pressure differences between the grid block centers[†], and $\mathbf{L}_{\boldsymbol{p}\mathbf{p}}$ is an $n_{con} \times n_{gb}$ selection matrix with entries –1, 0 and 1 in the appropriate places[‡]. E.g. for our 6-grid block Example 1 we find that $\mathbf{L}_{\boldsymbol{p}\mathbf{p}}$ is a 7×6 matrix given by

$$\mathbf{L}_{\boldsymbol{p}\mathbf{p}} = \begin{bmatrix} -1 & 1 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 1 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 1 & 0 \\ 0 & 0 & -1 & 0 & 0 & 1 \\ 0 & 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & 0 & -1 & 1 \end{bmatrix}, \tag{2.141}$$

In passing we note that we can almost never invert equation (2.140) to reconstruct the pressure vector $\mathbf{p}$ from the pressure difference vector $\boldsymbol{p}$, because $\mathbf{L}_{\boldsymbol{p}\mathbf{p}}$ is nearly always rectangular with $n_{con} > n_{gb}$. This illustrates that knowledge of just the inter-grid block pressure *differences* does not give us enough information to compute the absolute pressures in the grid blocks. The $n_{con} \times 1$ vector $\boldsymbol{v}_t$ of total Darcy velocities over the grid block boundaries can now be written as a function of $\boldsymbol{p}$ according to

$$\boldsymbol{v}_t = -diag\left(\boldsymbol{\lambda}\right) diag\left(\boldsymbol{\gamma}\right) \boldsymbol{p} \ . \tag{2.142}$$

Here $\boldsymbol{\lambda}$ is an $n_{con} \times 1$ vector of averaged mobilities,

$$\lambda_i \triangleq \left(\frac{k_i k_{ro,i}}{\mu_o} + \frac{k_i k_{rw,i}}{\mu_w} \right), \quad i = 1, \ldots, n_{con} \ , \tag{2.143}$$

where the absolute permeabilities k_i are harmonic averages of the adjacent grid blocks while the relative permeabilities $k_{ro,i}$ and $k_{rw,i}$ are usually upstream-weighted. The vector $\boldsymbol{\gamma}$ is another $n_{con} \times 1$ vector with elements γ_i that are geometric factors depending on the grid block properties. E.g. in the case of flow in the x direction γ is simply equal to $1/\Delta x$. Combining equations (2.140) and (2.142) we can write

$$\boldsymbol{v}_t = \mathbf{S}\mathbf{p} \ , \tag{2.144}$$

where the $n_{con} \times n_{gb}$ matrix $\mathbf{S}$ is given by

$$\mathbf{S} = -diag\left(\boldsymbol{\lambda}\right) diag\left(\boldsymbol{\gamma}\right) \mathbf{L}_{\boldsymbol{p}\mathbf{p}} \ . \tag{2.145}$$

[†] We use ***bold italics*** to represent vectors with properties at the grid blocks boundaries, whereas the conventional **bold** notation is used to represent vectors with properties at the grid blocks centers. In particular we use $\boldsymbol{p}$ and $\mathbf{p}$ for pressures, $\boldsymbol{\lambda}$ and $\boldsymbol{\lambda}$ for mobilities, $\boldsymbol{v}$ and $\mathbf{v}$ for Darcy velocities, and $\tilde{\boldsymbol{v}}$ and $\tilde{\mathbf{v}}$ for interstitial velocities.

[‡] This is an example of an *incidence matrix* as used in the analysis of e.g. electrical or mechanical networks, to define the pattern of nodes and edges. Other names used in the literature are *topology matrix* or *connectivity matrix*. Note that we use the term connectivity in a slightly different sense; see p. 69.

Equation (2.145) can be used to compute the inter grid block velocities for streamline tracking as described in Section 3.4.5.

2.5 References for Chapter 2

Aziz, K. and Settari, A., 1979: *Petroleum reservoir simulation*, Applied Science Publishers, London.

Bear, J., 1972: *Dynamics of fluids in porous media*, Elsevier, New York. Reprinted in 1988 by Dover, New York.

Buckley, S.E. and Leverett, M.C.: Mechanisms of fluid displacement in sands. *Petroleum Transactions AIME* **146**, 107-116.

Chen, Z., Huan, G. and Ma, Y., 2006: *Computational methods for multiphase flows in porous media*, SIAM, Philadelphia.

Ding, Y., Lemonnier, P.A., Estebenet, T. and Magras, J-F., 2000: Control-volume method for simulation in the well vicinity for arbitrary well configurations. *SPE Journal* **5** (1) 118-125. DOI: 10.2118/62169-PA.

Ewing, R.E., 1983: Problems arising in the modeling of processes for hydrocarbon recovery, in: *The mathematics of reservoir simulation*, Ewing, R.E. (ed.), SIAM, Philadelphia.

Helmig, R., 1997: *Multiphase flow and transport processes in the subsurface*, Springer.

Lake, L.W., 1989: *Enhanced Oil Recovery*, Prentice Hall, Upper Saddle River.

Mattax, C.C. and Dalton, R.L., 1990: *Reservoir simulation*. SPE Monograph Series **13**, SPE, Richardson.

Patankar, S.V., 1980: Numerical heat transfer and fluid flow. *Series in computational methods and in mechanics and thermal sciences*, Hemisphere.

Peaceman, D.W., 1977: *Fundamentals of numerical reservoir simulation*, Elsevier, Amsterdam.

Peaceman, D.W., 1978: Interpretation of well-block pressures in numerical reservoir simulation. *SPE Journal* **18** (3) 183-194. DOI: 10.2118/6893-PA.

Peaceman, D.W., 1983: Interpretation of well-block pressures in numerical reservoir simulation with nonsquare grid blocks and anisotropic permeability. *SPE Journal* **23** (3) 531-543. DOI: 10.2118/10528-PA.

Russel, T.F. and Wheeler, M.F., 1983: Finite element and finite difference methods for continuous flows in porous media, in: *The mathematics of reservoir simulation*, Ewing, R.E. (ed.), SIAM, Philadelphia.

Welge, H.J., 1952: A simplified method for computing oil recovery by gas or water drive. *Petroleum Transactions AIME* **195**, 91-98.

Whitson, C.H. and Brulé, M.R., 2000: Phase behavior. *SPE Monograph Series* **20**, SPE, Richardson.

3 System models

3.1 Notation

In this chapter we will present examples of dynamic system models and their system equations expressed in *state space* notation. This notation is the standard in most of the measurement and control literature and we make use of it to facilitate the application of system analysis techniques to reservoir engineering problems.

3.2 System equations

3.2.1 Partial differential equations

To describe the physics of fluid flow in a porous medium we generally use *partial differential equations* (PDEs). Typically the *independent* variables are time, t, and spatial coordinates x, y and z. Furthermore, in reservoir engineering we normally encounter only first-order derivatives in time, but higher order (typically second-order) derivatives in space. Indicating the, arbitrary, *dependent* variable with a fat dot, $\bullet$, the equations can be represented in general form as[†]

$$\varepsilon(t,x,y,z,\bullet)\times\frac{\partial(\bullet)}{\partial t}=\varphi(t,x,y,z,\bullet)\times L(\bullet)+\psi(t,x,y,z,\bullet), \tag{3.1}$$

where ε and φ are parameters that may be functions of time and space, L is a spatial *differential operator* and ψ is the *source term*. Note that ε, φ and ψ may be functions of the dependent variable, in which case the equation is nonlinear. The left-hand term in equation (3.1) is known as the *accumulation term*, the first term at the right-hand side as the *transport term.* A specific example of the general PDE (3.1) is the mass conservation equation (2.4) in Section 2.3.1 in which case the spatial difference operator L is the divergence $\nabla(\bullet)\triangleq\partial(\bullet)/\partial x+\partial(\bullet)/\partial y+\partial(\bullet)/\partial z$. In addition to the PDE (3.1), we need to specify the spatial *domain* Ω and the time domain T on which it is valid. At the *boundary* Γ of Ω we need to specify *boundary conditions*, and at a specific point in time an *initial condition*, to complete the problem formulation. A PDE describes the evolution of the dependent variables in time and space in a continuous fashion, and the solution is therefore specified in an infinitely large number of points. Closed-form solutions of PDEs, i.e. to 'infinite dimensional problems', are generally restricted to simple domains and parameters that are spatially homogeneous. For more realistic geometries we need to solve the equations numerically, which requires some form of *discretization* of the equations, because digital computers can only deal with finite dimensional problems.

3.2.2 Ordinary differential equations

An often followed approach is to first perform a spatial discretization of the PDEs, and only perform the time discretization at a later stage. The initial *semi-discretization* of the equations, i.e. the discretization in space, can be done using the method of finite differences, finite volumes or finite elements. An example of a finite difference discretization as applied to porous media flow has been worked out in Chapter 2. All of the discretization methods

[†] The dependent variables follow from the physics of the problem. In case of multi-phase flow through porous media they are typically pressures, component masses or phase saturations; see Chapter 2. Here we use only a single dependent variable, but in general multiple dependent variables will occur, in which case multiple differential equations are required to describe the problem.

result in a system of *ordinary differential equations* (ODEs) which can typically be represented as

$$\begin{cases} \hat{e}_1\left(\bullet_1, \dfrac{d(\bullet_1)}{dt}\right) = \hat{f}_1\left(t, \bullet_1, \bullet_2, \ldots, \bullet_n, \psi_1\right), \\ \hat{e}_2\left(\bullet_2, \dfrac{d(\bullet_2)}{dt}\right) = \hat{f}_2\left(t, \bullet_1, \bullet_2, \ldots, \bullet_n, \psi_2\right), \\ \qquad\qquad \vdots \\ \hat{e}_n\left(\bullet_n, \dfrac{d(\bullet_n)}{dt}\right) = \hat{f}_n\left(t, \bullet_1, \bullet_2, \ldots, \bullet_n, \psi_n\right), \end{cases} \tag{3.2}$$

where the *continuous* dependent variable $\bullet$ and the continuous source term ψ of equation (3.1) are now represented with a finite number of *discrete* values $\bullet_i$ and ψ_i, corresponding to discrete points in space, and where $\hat{e}_i$ and $\hat{f}_i$ are, in the general case, nonlinear functions[†]. Normally the functions $\hat{e}_i$ are linear in the derivatives $d(\bullet_i)/dt$, which makes it possible to transform the system of equations (3.2) such that the derivatives in the left-hand side terms are isolated, leading to:

$$\begin{cases} \dfrac{d(\bullet_1)}{dt} = f_1\left(t, \bullet_1, \bullet_2, \ldots, \bullet_n, \psi_1\right), \\ \dfrac{d(\bullet_2)}{dt} = f_2\left(t, \bullet_1, \bullet_2, \ldots, \bullet_n, \psi_2\right), \\ \qquad \vdots \\ \dfrac{d(\bullet_n)}{dt} = f_n\left(t, \bullet_1, \bullet_2, \ldots, \bullet_n, \psi_n\right), \end{cases} \tag{3.3}$$

where the functions f_i are different from the functions $\hat{f}_i$ in equation (3.2). Note that equations (3.2) and (3.3) are both coupled systems of ODEs because each of the dependent variables $\bullet_i$ is present in more than one equation. If the functions f_i are linear in the dependent variables, a further simplification is possible that decouples the equations leading to

$$\frac{d(\tilde{\bullet}_i)}{dt} = \tilde{f}_i\left(t, \tilde{\bullet}_i, \tilde{\psi}_i\right), \quad i = 1, 2, \ldots, n, \tag{3.4}$$

where the transformed dependent variables $\tilde{\bullet}_i$ are linear combinations of the original variables $\bullet_i$, and $\tilde{f}_i$ are functions again. This decoupling procedure will be addressed in more detail in Section 4.1.2 below. In reservoir simulation the functions $\hat{e}_i$ and $\hat{f}_i$ are typically linear in the derivatives and nonlinear in the dependent variables, which at first sight implies that equations (3.3) as the most relevant representation. Moreover, system-theoretical results are usually derived using equations of this particular form. However, for large scale computations it is more efficient to use representation (3.2), and in this text we will therefore make use of both formulations.

[†] Typically, most of the values $\bullet_i$ are equal to zero in a single equation. E.g. in the case of one-dimensional single-phase flow modeled with first-order finite differences, the only three non-zero values $\bullet_i$ in the i^{th} equation in a system of n equations with $1 < i < n$ are given by: $\hat{e}_i(\bullet_i, d(\bullet_i)/dt) = \hat{f}_i(\bullet_{i-1}, \bullet_i, \bullet_{i+1}, \psi_i)$.

3.2.3 State space representation

State equations

A more concise form of equations (3.3) can be obtained through the use of vector notation. We introduce the vectors $\mathbf{x} = [x_1 \quad x_2 \quad \ldots \quad x_n]^T$ and $\mathbf{u} = [u_1 \quad u_2 \quad \ldots \quad u_m]^T$ to represent the discrete values of the dependent variables, instead of $\bullet_i$ and ψ_i which we used until now. The reason to use **x** and **u** is to adhere to the notation convention in the systems and control literature. Note that $x_1, x_2, \ldots x_n$ do *not* represent spatial coordinates. Also note that we have indicated that the source term **u** has *m* elements instead of *n*. This anticipates a situation where many of the source terms are equal to zero, such that $m \ll n$, in which case it may be computationally advantageous to use a shorter vector **u**. Equations (3.3) can now be written as

$$\dot{\mathbf{x}}(t) = \mathbf{f}\left(t, \mathbf{u}(t), \mathbf{x}(t)\right), \tag{3.5}$$

where **f** is a nonlinear vector function of **x**, **u** and *t*, and where we have emphasized the dependence of **x** and **u** on *t* by writing **x**(*t*) and **u**(*t*). In reservoir simulation, the equations are usually nonlinear but with coefficients that do not depend on time directly, such that we can write

$$\dot{\mathbf{x}}(t) = \mathbf{f}\left(\mathbf{u}(t), \mathbf{x}(t)\right). \tag{3.6}$$

In the special case that **f** is a linear function of **x** and **u**, we can use a vector-matrix notation and write equation (3.6) as a *linear time-varying* (LTV) vector differential equation

$$\dot{\mathbf{x}}(t) = \mathbf{A}(t)\mathbf{x}(t) + \mathbf{B}(t)\mathbf{u}(t), \tag{3.7}$$

where the coefficients of the $n \times n$ matrix **A** and the $n \times m$ matrix **B** may still be functions of *t*. The matrices **A** and **B** are usually referred to as the *system matrix* and the *input matrix* respectively[‡]. In the case that **A** and **B** are independent of *t*, we obtain a *linear time-invariant* (LTI) equation given by

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t). \tag{3.8}$$

From now on we will mostly not explicitly indicate the dependence on time of the variables, and we will write, e.g., $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{u}, \mathbf{x})$ instead of $\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{u}(t), \mathbf{x}(t))$. First-order systems of equations such as (3.5), (3.6), (3.7) and (3.8) are referred to as *state equations* in the systems and control literature. In this representation, the elements of vector **x** are the state variables which completely define the dynamic state of the system. A continuous sequence of values of **x** over a certain time interval is often referred to as a *trajectory* in state space. In reservoir engineering applications it is sometimes preferred to start from equations (3.2) rather than from equation (3.3), even if the functions e_i are linear. We will refer to equations of the type of equation (3.2) as *generalized state equations*[†]. In LTI form they can be written as

$$\hat{\mathbf{E}}\dot{\mathbf{x}} = \hat{\mathbf{A}}\mathbf{x} + \hat{\mathbf{B}}\mathbf{u}. \tag{3.9}$$

[‡] An alternative name for the input matrix is *distribution matrix* because it distributes the inputs **u** over the states **x**.

[†] Sometimes this form of generalized state equations is referred to as a *descriptor system*.

Output equations

In addition to equations (3.5) to (3.9), which define the relationship between the input vector **u** and the state vector **x** of a dynamic system, we can also define a relationship between an output vector **y** and the state **x**. Moreover, the output may to some extent also depend directly on the input **u**, such that we can write

$$\mathbf{y} = \mathbf{h}(\mathbf{u}, \mathbf{x}), \tag{3.10}$$

for the nonlinear case or

$$\mathbf{y} = \mathbf{C}\mathbf{x} + \mathbf{D}\mathbf{u}, \tag{3.11}$$

for the linear case, where **C** is known as the *output matrix* and **D** as the *direct throughput matrix*. If the output vector **y** has p elements, the matrices **C** and **D** have dimensions $p \times n$ and $p \times m$, respectively.

Implicit nonlinear equations

In addition to the general nonlinear system functions (3.6) and (3.10) we will sometimes use even more general nonlinear functions

$$\mathbf{g}(\mathbf{u}, \mathbf{x}, \dot{\mathbf{x}}) = \mathbf{0}, \tag{3.12}$$

$$\mathbf{j}(\mathbf{u}, \mathbf{x}, \mathbf{y}) = \mathbf{0}, \tag{3.13}$$

where **g** and **j** are nonlinear vector-valued functions[‡]. Note that the explicit equations (3.6) and (3.10) can always simply be expressed in the implicit form of equations (3.12) and (3.13), i.e.

$$\mathbf{g}(\mathbf{u}, \mathbf{x}, \dot{\mathbf{x}}) = \dot{\mathbf{x}} - \mathbf{f}(\mathbf{u}, \mathbf{x}), \tag{3.14}$$

$$\mathbf{j}(\mathbf{u}, \mathbf{x}, \mathbf{y}) = \mathbf{y} - \mathbf{h}(\mathbf{u}, \mathbf{x}). \tag{3.15}$$

The reverse is not always true, i.e. it may not be possible to derive an explicit expression **f** for a given implicit representation **g**. However, usually the implicit representation may be solved numerically for $\dot{\mathbf{x}}$, typically using some form of time discretization and an iterative algorithm. In that case we can still conceptually write the nonlinear equations in their explicit forms (3.6) and (3.10) which is often preferred for analysis purposes. In most cases the functions **f**, **g**, **h** and **j** are to be interpreted as numerical operations, e.g. **f** could represent a complete reservoir simulator. Detailed examples of the state variable description of reservoir systems will be discussed below.

Error terms

In systems and control applications it is customary to introduce *error terms* to account for the fact that a system description is only an approximation of reality. E.g. we can write

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} + \boldsymbol{\varepsilon}, \tag{3.16}$$

$$\mathbf{y} = \mathbf{C}\mathbf{x} + \mathbf{D}\mathbf{u} + \boldsymbol{\eta}, \tag{3.17}$$

for the linear case, where $\boldsymbol{\varepsilon}$ is called the *model error* and $\boldsymbol{\eta}$ the *measurement error*. Both are random variables[†] that are often, although not necessarily, taken as zero-mean Gaussian. As a

[‡] System equations expressed as $\mathbf{g}(\ldots) = \mathbf{0}$ are sometimes referred to as equations in *residual* form.

result of the random error terms, **x** and **y** also become random variables such that to completely quantify them it will be necessary to specify their probability distributions as functions of time. In the special case of a linear equations, Gaussian error terms will result in Gaussian states and outputs which can be completely specified by their first and second moments (mean values and covariance matrices). In the more general, nonlinear case, it will be necessary to specify higher moments or ensembles of representative *realizations* of **x** and **y**. In reservoir simulation it is not customary to introduce error terms that are additive to the states (as in equation (3.16). Instead it is much more common to consider the parameters of the system equations, in particular the grid block permabilities, as uncertain. Typically these parameter uncertainties are considerd to be so large that they dominate the model errors. Measurement errors are normally introduced in computer-assisted history matching as will be discussed in detail in Chapter 8. Until then we will not make use of error tems in the system description.

3.2.4 Linearized equations

Jacobians

To analyze the nature of nonlinear system equations $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{u}, \mathbf{x})$, or to approximate their solution through numerical computation, it is usually necessary to linearize them around a point in state-input space. Using a Taylor expansion and starting from equation (3.6) we can write:

$$\dot{\mathbf{x}} = \mathbf{f}\left(\mathbf{u}, \mathbf{x}\right) \approx \mathbf{f}\left(\mathbf{u}^0, \mathbf{x}^0\right) + \frac{\partial \mathbf{f}\left(\mathbf{u}^0, \mathbf{x}^0\right)}{\partial \mathbf{u}}\left(\mathbf{u} - \mathbf{u}^0\right) + \frac{\partial \mathbf{f}\left(\mathbf{u}^0, \mathbf{x}^0\right)}{\partial \mathbf{x}}\left(\mathbf{x} - \mathbf{x}^0\right) , \tag{3.18}$$

where we have neglected terms of second order and higher, and applied the usual short-cut notation

$$\frac{\partial \mathbf{f}\left(\mathbf{u}^0, \mathbf{x}^0\right)}{\partial \mathbf{x}} \triangleq \frac{\partial \mathbf{f}\left(\mathbf{u}, \mathbf{x}\right)}{\partial \mathbf{x}}\bigg|_{\mathbf{u}=\mathbf{u}^0,\, \mathbf{x}=\mathbf{x}^0} . \tag{3.19}$$

Defining

$$\overline{\mathbf{u}} \triangleq \mathbf{u} - \mathbf{u}^0 , \overline{\mathbf{x}} \triangleq \mathbf{x} - \mathbf{x}^0 , \tag{3.20, 3.21}$$

we can rewrite equation (3.18) as

$$\dot{\overline{\mathbf{x}}} + \dot{\mathbf{x}}^0 \approx \mathbf{f}\left(\mathbf{u}^0, \mathbf{x}^0\right) + \frac{\partial \mathbf{f}\left(\mathbf{u}^0, \mathbf{x}^0\right)}{\partial \mathbf{u}}\overline{\mathbf{u}} + \frac{\partial \mathbf{f}\left(\mathbf{u}^0, \mathbf{x}^0\right)}{\partial \mathbf{x}}\overline{\mathbf{x}} , \tag{3.22}$$

which, because

$$\dot{\mathbf{x}}^0 = \mathbf{f}\left(\mathbf{u}^0, \mathbf{x}^0\right) , \tag{3.23}$$

can be reduced to the *linearized system equations*

$$\dot{\overline{\mathbf{x}}} = \overline{\mathbf{A}}\left(\mathbf{u}^0, \mathbf{x}^0\right)\overline{\mathbf{x}} + \overline{\mathbf{B}}\left(\mathbf{u}^0, \mathbf{x}^0\right)\overline{\mathbf{u}} , \tag{3.24}$$

where the *Jacobian matrices*† $\overline{\mathbf{A}}$ and $\overline{\mathbf{B}}$ are defined as

---

† The random model errors are also referred to as *random input*, or as a *stochastic forcing term*.

† Usually simply referred to as *Jacobians*.

$$\bar{\mathbf{A}}\left(\mathbf{u}^0,\mathbf{x}^0\right) \triangleq \frac{\partial \mathbf{f}\left(\mathbf{u}^0,\mathbf{x}^0\right)}{\partial \mathbf{x}} \ , \quad \bar{\mathbf{B}}\left(\mathbf{u}^0,\mathbf{x}^0\right) \triangleq \frac{\partial \mathbf{f}\left(\mathbf{u}^0,\mathbf{x}^0\right)}{\partial \mathbf{u}} \ . \qquad (3.25, 3.26)$$

In a similar fashion we can linearize a nonlinear output function $\mathbf{y} = \mathbf{h}(\mathbf{u},\mathbf{x})$ to obtain

$$\bar{\mathbf{y}} = \bar{\mathbf{C}}\left(\mathbf{u}^0,\mathbf{x}^0\right)\bar{\mathbf{x}} + \bar{\mathbf{D}}\left(\mathbf{u}^0,\mathbf{x}^0\right)\bar{\mathbf{u}} \ , \qquad (3.27)$$

where the Jacobians $\bar{\mathbf{C}}$ and $\bar{\mathbf{D}}$ are defined as

$$\bar{\mathbf{C}}\left(\mathbf{u}^0,\mathbf{x}^0\right) \triangleq \frac{\partial \mathbf{h}\left(\mathbf{u}^0,\mathbf{x}^0\right)}{\partial \mathbf{x}} \ , \quad \bar{\mathbf{D}}\left(\mathbf{u}^0,\mathbf{x}^0\right) \triangleq \frac{\partial \mathbf{h}\left(\mathbf{u}^0,\mathbf{x}^0\right)}{\partial \mathbf{u}} \ . \qquad (3.28, 3.29)$$

If the system and output equations are given in implicit form $\mathbf{g}(\mathbf{u},\mathbf{x},\dot{\mathbf{x}}) = \mathbf{0}$ and $\mathbf{j}(\mathbf{u},\mathbf{x},\mathbf{y}) = \mathbf{0}$ we obtain linearized equations in terms of Jacobians

$$\bar{\hat{\mathbf{A}}} \triangleq \frac{\partial \mathbf{g}\left(\mathbf{u},\mathbf{x},\dot{\mathbf{x}}\right)}{\partial \mathbf{x}} \ , \quad \bar{\hat{\mathbf{B}}} \triangleq \frac{\partial \mathbf{g}\left(\mathbf{u},\mathbf{x},\dot{\mathbf{x}}\right)}{\partial \mathbf{u}} \ , \quad \bar{\hat{\mathbf{E}}} \triangleq \frac{\partial \mathbf{g}\left(\mathbf{u},\mathbf{x},\dot{\mathbf{x}}\right)}{\partial \dot{\mathbf{x}}} \ , \quad (3.30, 3.31, 3.32)$$

$$\bar{\hat{\mathbf{C}}} \triangleq \frac{\partial \mathbf{j}\left(\mathbf{u},\mathbf{x},\mathbf{y}\right)}{\partial \mathbf{x}} \ , \quad \bar{\hat{\mathbf{D}}} \triangleq \frac{\partial \mathbf{j}\left(\mathbf{u},\mathbf{x},\mathbf{y}\right)}{\partial \mathbf{u}} \ , \quad \bar{\hat{\mathbf{F}}} \triangleq \frac{\partial \mathbf{g}\left(\mathbf{u},\mathbf{x},\mathbf{y}\right)}{\partial \mathbf{y}} \ , \quad (3.33, 3.34, 3.35)$$

where we have dropped the supercripts 0 for clarity.

Secant and tangent matrices

In reservoir simulation one often encounters systems $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{u},\mathbf{x})$ that can be expressed in the form*

$$\dot{\mathbf{x}} = \mathbf{A}\left(\mathbf{x}\right)\mathbf{x} + \mathbf{B}\left(\mathbf{x}\right)\mathbf{u} \ . \qquad (3.36)$$

In that case we obtain the linearized equations (3.24) with Jacobians defined as

$$\bar{\mathbf{A}}\left(\mathbf{u}^0,\mathbf{x}^0\right) \triangleq \mathbf{A}\left(\mathbf{x}^0\right) + \frac{\partial \mathbf{A}\left(\mathbf{x}^0\right)}{\partial \mathbf{x}}\mathbf{x}^0 + \frac{\partial \mathbf{B}\left(\mathbf{x}^0\right)}{\partial \mathbf{x}}\mathbf{u}^0 \ , \quad \bar{\mathbf{B}}\left(\mathbf{x}^0\right) \triangleq \mathbf{B}\left(\mathbf{x}^0\right) \ . \quad (3.37, 3.38)$$

If we linearize the state equations along all points of a given trajectory $(\mathbf{x}^0(t),\mathbf{u}^0(t))$ in state-input space, the resulting model is referred to as the *tangent-linear* approximation of the nonlinear model, or simply the tangent-linear model. The Jacobians $\bar{\mathbf{A}}$ and $\bar{\mathbf{B}}$ are therefore also referred to as the *tangent matrices* of the system. Note that the matrices **A** and **B** are not tangent matrices because they do not describe the system dynamics tangent to the the state trajectory. Instead they can be interpreted as *secant matrices*; see Figure 3.1.

* In the systems and control literature this is known as a *control affine* nonlinear equation.

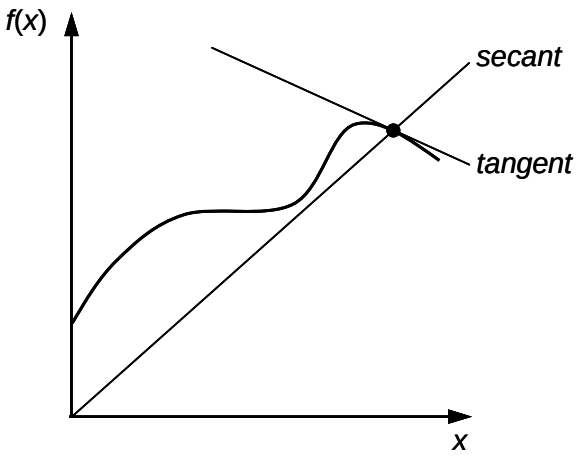


Figure 3.1: The secant and the tangent to a function $f(x)$ in a point $(x^0, f(x^0))$.

*Generalized state space form**

In reservoir simulation we often encounter systems that can be expressed in the generalized state space form

$$\hat{\mathbf{E}}(\mathbf{x})\dot{\mathbf{x}} = \hat{\mathbf{A}}(\mathbf{x})\mathbf{x} + \hat{\mathbf{B}}(\mathbf{x})\mathbf{u} \ . \tag{3.39}$$

In that case we can linearize around a point $(\mathbf{u}^0, \mathbf{x}^0, \dot{\mathbf{x}}^0)$ to obtain the linearized equations

$$\bar{\hat{\mathbf{E}}}(\mathbf{x}^0)\dot{\bar{\mathbf{x}}} = \bar{\hat{\mathbf{A}}}(\mathbf{u}^0, \mathbf{x}^0, \dot{\mathbf{x}}^0)\bar{\mathbf{x}} + \bar{\hat{\mathbf{B}}}(\mathbf{x}^0)\bar{\mathbf{u}} \ , \tag{3.40}$$

with Jacobians defined as

$$\bar{\hat{\mathbf{A}}}(\mathbf{u}^0, \mathbf{x}^0, \dot{\mathbf{x}}^0) \triangleq \hat{\mathbf{A}}(\mathbf{x}^0) + \frac{\partial \hat{\mathbf{A}}(\mathbf{x}^0)}{\partial \mathbf{x}}\mathbf{x}^0 + \frac{\partial \hat{\mathbf{B}}(\mathbf{x}^0)}{\partial \mathbf{x}}\mathbf{u}^0 - \frac{\partial \hat{\mathbf{E}}(\mathbf{x}^0)}{\partial \mathbf{x}}\dot{\mathbf{x}}^0 \ ,$$

$$\bar{\hat{\mathbf{B}}}(\mathbf{x}^0) \triangleq \hat{\mathbf{B}}(\mathbf{x}^0) \ , \ \bar{\hat{\mathbf{E}}}(\mathbf{x}^0) \triangleq \hat{\mathbf{E}}(\mathbf{x}^0). \tag{3.41, 3.42, 3.43}$$

However, for analysis purposes it is normally more usefull to bring this equation in the linearized ordinary state space form (3.24):

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{u}, \mathbf{x}) = \underbrace{\left(\hat{\mathbf{E}}(\mathbf{x})\right)^{-1}\hat{\mathbf{A}}(\mathbf{x})}_{\mathbf{A}(\mathbf{x})}\mathbf{x} + \underbrace{\left(\hat{\mathbf{E}}(\mathbf{x})\right)^{-1}\hat{\mathbf{B}}(\mathbf{x})}_{\mathbf{B}(\mathbf{x})}\mathbf{u} \ . \tag{3.44}$$

in which case the Jacobians $\bar{\mathbf{A}} = \partial\mathbf{f}(\mathbf{u}, \mathbf{x})/\partial\mathbf{x}$ and $\bar{\mathbf{B}} = \partial\mathbf{f}(\mathbf{u}, \mathbf{x})/\partial\mathbf{u}$ can be obtained as

$$\bar{\mathbf{A}}(\mathbf{u}^0, \mathbf{x}^0) \triangleq \left(\hat{\mathbf{E}}(\mathbf{x}^0)\right)^{-1} \left\{ \begin{array}{l} \hat{\mathbf{A}}(\mathbf{x}^0) + \left[\frac{\partial \hat{\mathbf{A}}(\mathbf{x}^0)}{\partial \mathbf{x}} - \frac{\partial \hat{\mathbf{E}}(\mathbf{x}^0)}{\partial \mathbf{x}}\left(\hat{\mathbf{E}}(\mathbf{x}^0)\right)^{-1}\hat{\mathbf{A}}(\mathbf{x}^0)\right]\mathbf{x}^0 \\ \quad + \left[\frac{\partial \hat{\mathbf{B}}(\mathbf{x}^0)}{\partial \mathbf{x}} - \frac{\partial \hat{\mathbf{E}}(\mathbf{x}^0)}{\partial \mathbf{x}}\left(\hat{\mathbf{E}}(\mathbf{x}^0)\right)^{-1}\hat{\mathbf{B}}(\mathbf{x}^0)\right]\mathbf{u}^0 \end{array} \right\}, \tag{3.45}$$

$$\bar{\mathbf{B}}(\mathbf{x}^0) \triangleq \left(\hat{\mathbf{E}}(\mathbf{x}^0)\right)^{-1}\mathbf{B}(\mathbf{x}^0) \ . \tag{3.46}$$

Alternatively we can write the generalized state equation (3.39) in implicit form

$$\hat{\mathbf{g}}\left(\mathbf{u},\mathbf{x},\dot{\mathbf{x}}\right) \triangleq \hat{\mathbf{E}}\left(\mathbf{x}\right)\dot{\mathbf{x}} - \hat{\mathbf{A}}\left(\mathbf{x}\right)\mathbf{x} - \hat{\mathbf{B}}\left(\mathbf{x}\right)\mathbf{u} = \mathbf{0} \ , \tag{3.47}$$

and use implicit differentiation to obtain the Jacobian $\bar{\mathbf{A}}$ related to the ordinary state space representation. I.e., because

$$\frac{d\hat{\mathbf{g}}}{d\mathbf{x}} = \frac{\partial\hat{\mathbf{g}}}{\partial\mathbf{x}} + \frac{\partial\hat{\mathbf{g}}}{\partial\dot{\mathbf{x}}}\frac{\partial\dot{\mathbf{x}}}{\partial\mathbf{x}} = \mathbf{0} \ , \tag{3.48}$$

we have

$$\frac{\partial\dot{\mathbf{x}}}{\partial\mathbf{x}} = -\left(\frac{\partial\hat{\mathbf{g}}}{\partial\dot{\mathbf{x}}}\right)^{-1}\frac{\partial\hat{\mathbf{g}}}{\partial\mathbf{x}} \ , \tag{3.49}$$

and because $\bar{\mathbf{A}} = \partial\mathbf{f}(\mathbf{u},\mathbf{x})/\partial\mathbf{x} = \partial\dot{\mathbf{x}}/\partial\mathbf{x}$ we find that

$$\bar{\mathbf{A}}\left(\mathbf{u}^0,\mathbf{x}^0,\dot{\mathbf{x}}^0\right) = \left(\hat{\mathbf{E}}\left(\mathbf{x}^0\right)\right)^{-1}\left[\hat{\mathbf{A}}\left(\mathbf{x}^0\right) + \frac{\partial\hat{\mathbf{A}}\left(\mathbf{x}^0\right)}{\partial\mathbf{x}}\mathbf{x}^0 + \frac{\partial\hat{\mathbf{B}}\left(\mathbf{x}^0\right)}{\partial\mathbf{x}}\mathbf{u}^0 - \frac{\partial\hat{\mathbf{E}}\left(\mathbf{x}^0\right)}{\partial\mathbf{x}}\dot{\mathbf{x}}^0\right] , \tag{3.50}$$

which, with the aid of equations (3.41) and (3.43), can also be expressed as

$$\bar{\mathbf{A}}\left(\mathbf{u}^0,\mathbf{x}^0,\dot{\mathbf{x}}^0\right) = \left(\bar{\hat{\mathbf{E}}}\left(\mathbf{x}^0\right)\right)^{-1}\bar{\hat{\mathbf{A}}}\left(\mathbf{u}^0,\mathbf{x}^0,\dot{\mathbf{x}}^0\right) \ . \tag{3.51}$$

3.3 Single-phase flow

3.3.1 System equations

As a first application, we consider flow of a weakly compressible single-phase liquid through a porous medium. The derivation of the governing PDEs and the semi-discretization has been presented in Chapter 2. We used a finite difference discretization, but the following theory is equally applicable to results derived with other semi-discretization methods. Use of any of the methods produces a system of ODEs that can be written in matrix form as:

$$\mathbf{V}\dot{\mathbf{p}} + \mathbf{T}\mathbf{p} = \mathbf{q} \, , \tag{3.52}$$

Here **V** and **T** are matrices with entries that depend on dynamic and static reservoir properties, **p** is a vector of pressures and **q** is a vector of volumetric flow rates. **V** is a diagonal matrix known as the *accumulation matrix* and **T** is a symmetric banded matrix, known as the *transmissibility matrix*. The flow rates **q** correspond to flow in to or out of the reservoir, i.e. to wells, and are expressed in m^3/s. Positive values imply injection and negative values imply production. Because usually only a few grid blocks are penetrated by wells, only a few elements of **q** have a non-zero value. In the case of a reservoir modeled with n grid blocks and produced with m wells, **V** and **T** would be $n \times n$ matrices, and **p** and **q** would be $n \times 1$ vectors, of which **q** would have m non-zero entries. Equation (3.52) can be re-casted in state variable form (3.8) through definition of

$$\mathbf{u} \triangleq \mathbf{L}_{uq}\mathbf{q} \ , \ \ \mathbf{x} \triangleq \mathbf{p} \ . \tag{3.53, 3.54}$$

In single-phase flow the state variables **x** are just identical to the pressures **p**. The vector **u** represents the inputs to the system, which are in our case the non-zero elements of the flow rate vector **q**. The matrix $\mathbf{L}_{uq}$ is therefore a *location matrix*, also known as a *selection matrix* which contains only ones and zeros at the appropriate places. The inverse relationship is given by

$$\mathbf{q} = \mathbf{L}_{qu}\mathbf{u} \ , \tag{3.55}$$

where

$$\mathbf{L}_{qu} = \mathbf{L}_{uq}^{T} \ , \tag{3.56}$$

Substitution of relationships (3.53) and (3.55) in equation (3.52) results in the generalized state-space representation (3.9) with matrices $\hat{\mathbf{A}}$ and $\hat{\mathbf{B}}$ and $\hat{\mathbf{E}}$ given by:

$$\hat{\mathbf{A}} \triangleq -\mathbf{T}, \ \hat{\mathbf{B}} \triangleq \mathbf{L}_{qu}, \ \hat{\mathbf{E}} \triangleq \mathbf{V} \ . \tag{3.57, 3.58, 3.59}$$

The ordinary state-space form (3.8) is obtained by defining the matrices **A** and **B** as[†]

$$\mathbf{A} \triangleq -\mathbf{V}^{-1}\mathbf{T}, \ \mathbf{B} \triangleq \mathbf{V}^{-1}\mathbf{L}_{qu} \ . \tag{3.60, 3.61}$$

If we choose the output vector **y** to consist of only those pressures that are accessible to measurements, the matrix **C** is therefore also a selection matrix. Matrix **D** is zero because there is no direct dependency of the output on the input. In reality the outputs are usually surface measurements of the tubing head pressure, and therefore we should include a description of the dynamic behavior of the well between the reservoir and the surface. However, as a first assumption, we neglect well dynamics and assume that the wells are equipped with permanent downhole gauges (PDGs) to measure the pressures. In the case of a reservoir modeled with n grid blocks and produced with m wells, of which m_p contain PDGs, matrices **A** and **B** have dimension $n \times n$, matrix **C** dimension $m_p \times n$, and vectors **u**, **x** and **y** dimensions $m \times 1$, $n \times 1$ and $m_p \times 1$ respectively. Matrix equation (3.8) represents a system of linear first order ODEs with constant coefficients, i.e. an LTI system. Starting from an initial value $\breve{\mathbf{x}}$, the ODEs for **x** can be integrated in time, and because the equations are linear the solution can be expressed analytically. Alternatively, the integration can be performed numerically as will be discussed in Chapter 4.

3.3.2 Example 1 continued – Location matrix

Reconsidering the six grid block example introduced in Section 2.3.3, the location matrix $\mathbf{L}_{uq}$ as defined in equation (3.53) is given by

$$\underbrace{\begin{bmatrix} u_1 \\ u_2 \end{bmatrix}}_{\mathbf{u}} = \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}}_{\mathbf{L}_{uq}} \underbrace{\begin{bmatrix} q_1 \\ 0 \\ 0 \\ 0 \\ 0 \\ q_6 \end{bmatrix}}_{\mathbf{q}} . \tag{3.62}$$

If the output **y** consists of the pressures in the two wells, the output matrix $\mathbf{C} = \mathbf{L}_{qu} = \mathbf{L}_{uq}^{T}$ is given by :

[†] In a numerical implementation, the inverse $\mathbf{V}^{-1}$ of the diagonal matrix **V** can be computed very efficiently by just taking the reciprocals of the diagonal elements.

$$\underbrace{\begin{bmatrix} y_1 \\ y_2 \end{bmatrix}}_{\mathbf{y}} = \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}}_{\mathbf{C}} \underbrace{\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix}}_{\mathbf{x}} . \tag{3.63}$$

3.3.3 Prescribed pressures and flow rates

Until now we have assumed that the source terms, i.e. the flow rates in the wells, were the input variables, and that their values can be prescribed as a function of time. However, it is also possible to control the system by prescribing the state variables, i.e. the pressures in the wells. Note that it is not possible to prescribe both pressure and flow rate in a well; either one of them should be fixed and the other left free, or a mathematical relationship between them should be specified which may be in algebraic or differential form. The most commonly used method in reservoir engineering is through the definition of a *well model*, which is an algebraic relationship between the grid block pressure and the well flow rate. Alternatively, one of the pressures may be prescribed directly, resulting in a reduction of the length of the state vector with one element. We will discuss both methods in the following sections. In order to take into account prescribed pressures and flow rates in a structured way it is convenient to re-order the variables in equation (3.52) such that the prescribed and the non-prescribed, free, values are grouped. In addition, we take the opportunity to make a distinction between prescribed flow rates in grid blocks with and without wells[†]. We can formally describe the re-ordering with the aid of a *permutation matrix*. $\mathbf{L}$ as:

$$\mathbf{p}^* \triangleq \begin{bmatrix} \mathbf{p}_1^* \\ \mathbf{p}_2^* \\ \mathbf{p}_3^* \end{bmatrix} \equiv \mathbf{L}_{p^*p}\mathbf{p} \ , \tag{3.64,}$$

where $\mathbf{p}_1^*$ are the pressures in the grid blocks that are not penetrated by a well, i.e. where the source terms $\mathbf{q}_1^*$ are equal to zero, $\mathbf{p}_2^*$ are the pressures in the blocks where the source terms $\mathbf{q}_2^*$ are prescribed as well flow rates, and $\mathbf{p}_3^*$ are the pressures in the blocks where the source terms $\mathbf{q}_3^*$ are obtained through prescription of the bottom hole pressures in the wells. Similarly we can write

$$\mathbf{q}^* \triangleq \begin{bmatrix} \mathbf{0} \\ \mathbf{q}_2^* \\ \mathbf{q}_3^* \end{bmatrix} \equiv \mathbf{L}_{q^*q}\mathbf{q} \ , \tag{3.65}$$

where we choose

$$\mathbf{L}_{q^*q} = \mathbf{L}_{p^*p} \ , \tag{3.66}$$

[†] In grid blocks that are not penetrated by a well the prescribed flow rates are of course equal to zero.

which means that we reorder the elements of **p** and **q** in equation (3.52) identically, i.e. we interchange the rows of the equations. The permutation matrix $\mathbf{L}_{p^*p} = \mathbf{L}_{q^*q}$ is an identity matrix with interchanged rows. Permutation matrices are orthogonal, which implies that

$$\mathbf{L}_{p^*p}\mathbf{L}_{p^*p}^T = \mathbf{I} \ . \tag{3.67}$$

The inverse relationships corresponding to expressions (3.64) and (3.65) are therefore given by

$$\mathbf{p} = \mathbf{L}_{pp^*}\mathbf{p}^* \ , \ \mathbf{q} = \mathbf{L}_{qq^*}\mathbf{q}^* \ , \tag{3.68, 3.69}$$

where

$$\mathbf{L}_{pp^*} = \mathbf{L}_{qq^*} = \mathbf{L}_{p^*p}^T = \mathbf{L}_{q^*q}^T \ . \tag{3.70}$$

Substitution of equations (3.68) and (3.69) in equations (3.52) and reorganizing the terms results in

$$\mathbf{V}^*\dot{\mathbf{p}}^* + \mathbf{T}^*\mathbf{p}^* = \mathbf{q}^* , \tag{3.71}$$

where $\mathbf{T}^*$ and $\mathbf{V}^*$ are given by

$$\mathbf{T}^* = \mathbf{L}_{q^*q}\mathbf{T}\mathbf{L}_{pp^*} \ , \ \mathbf{V}^* = \mathbf{L}_{q^*q}\mathbf{V}\mathbf{L}_{pp^*} \ . \tag{3.72, 3.73}$$

Equation (3.71) can be written in partitioned form as:

$$\begin{bmatrix} \mathbf{V}_{11}^* & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{V}_{22}^* & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{V}_{33}^* \end{bmatrix} \begin{bmatrix} \dot{\mathbf{p}}_1^* \\ \dot{\mathbf{p}}_2^* \\ \dot{\mathbf{p}}_3^* \end{bmatrix} + \begin{bmatrix} \mathbf{T}_{11}^* & \mathbf{T}_{12}^* & \mathbf{T}_{13}^* \\ \mathbf{T}_{21}^* & \mathbf{T}_{22}^* & \mathbf{T}_{23}^* \\ \mathbf{T}_{31}^* & \mathbf{T}_{32}^* & \mathbf{T}_{33}^* \end{bmatrix} \begin{bmatrix} \mathbf{p}_1^* \\ \mathbf{p}_2^* \\ \mathbf{p}_3^* \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{q}_2^* \\ \mathbf{q}_3^* \end{bmatrix} \ . \tag{3.74}$$

Note that the diagonal structure of matrix **V** has been maintained in $\mathbf{V}^*$. We can also apply the partitioning to the state space representation, in which case we may choose not to partition **u** and **y**, or to partition them also. We choose to partition them, according to

$$\mathbf{u}^* \triangleq \mathbf{L}_{u^*u}\mathbf{u} \ , \ \mathbf{y}^* \triangleq \mathbf{L}_{y^*y}\mathbf{y} \ , \tag{3.75, 3.76}$$

where details of the partitioning are left open for the moment. Substitution of equations (3.68), (3.69) and the inverse of equation (3.75) in equations (3.53) and (3.54) results in

$$\mathbf{u}^* = \mathbf{L}_{u^*q^*}\mathbf{q}^* \ , \ \mathbf{x}^* = \mathbf{p}^* \ , \tag{3.77, 3.78}$$

where

$$\mathbf{L}_{u^*q^*} = \mathbf{L}_{u^*u}\mathbf{L}_{uq}\mathbf{L}_{qq^*} \ , \ \mathbf{x}^* = \mathbf{L}_{p^*p}\mathbf{x} \ . \tag{3.79, 3.80}$$

The partitioned state space representation can then be written as

$$\dot{\mathbf{x}}^* = \mathbf{A}^*\mathbf{x}^* + \mathbf{B}^*\mathbf{u}^* \ , \ \mathbf{y}^* = \mathbf{C}^*\mathbf{x}^* \ , \tag{3.81, 3.82}$$

where

$$\mathbf{A}^* \triangleq -\left(\mathbf{V}^*\right)^{-1}\mathbf{T}^* \equiv \mathbf{L}_{q^*q}\mathbf{A}\mathbf{L}_{pp^*} \ , \tag{3.83}$$

$$\mathbf{B}^* \triangleq \left(\mathbf{V}^*\right)^{-1}\mathbf{L}_{q^*u^*} \equiv \mathbf{L}_{q^*q}\mathbf{B}\mathbf{L}_{uu^*} \ , \tag{3.84}$$

$$\mathbf{C}^* \triangleq \mathbf{L}_{y^*y}\mathbf{C}\mathbf{L}_{pp^*}. \tag{3.85}$$

The reordering of vector and matrix elements using permutation matrices as described above is a formal technique. It results in partitioned vectors and matrices that allow for a structured handling of prescribed pressures. However, for a numerical implementation it is not essential to actually perform the reordering. In the following we will therefore omit the star superscripts and simply work with partitioned matrices without the use of permutation matrices.

3.3.4 Well models

Prescribed bottom hole pressures and well flow rates

The standard approach in reservoir simulation to prescribe bottom hole pressures is through the definition of a *well model*. In that case the flow rate q in the grid block where we want to prescribe the pressure is defined as

$$q = J_{well}\left(\breve{p}_{well} - p\right), \tag{3.86}$$

where $\breve{p}_{well}$ is the prescribed bottom hole pressure, p is the grid block pressure and J_{well} is called the *well index* or *productivity index*. The well index is a function of the grid block geometry and reflects the effect of near-well flow which is normally not properly represented by the finite difference discretization because the grid block dimensions are usually much larger than the well diameter; see also Section 2.3.6. Note that, in line with our convention, a negative flow rate indicates production. Use of equation (3.86) can be interpreted as specifying an algebraic relationship between the state variable (i.e. the pressure) and the source term (i.e. the flow rate) in the grid block that contains the well. Equation (3.86) can be generalized for multiple wells to

$$\mathbf{q}_3 = \mathbf{J}_3\left(\breve{\mathbf{p}}_{well} - \mathbf{p}_3\right), \tag{3.87}$$

where $\mathbf{J}_3$ is a diagonal matrix of well indices J_{well}, and $\breve{\mathbf{p}}_{well}$ is a vector of prescribed bottom hole pressures. In a similar fashion, we can write

$$\mathbf{q}_2 = \breve{\mathbf{q}}_{well}, \tag{3.88}$$

where $\breve{\mathbf{q}}_{well}$ are the prescribed well rates. If we combine equations (3.87) and (3.88) with equation (3.74), and reorganize terms, we obtain

$$\begin{bmatrix} \mathbf{V}_{11} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{V}_{22} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{V}_{33} \end{bmatrix} \begin{bmatrix} \dot{\mathbf{p}}_1 \\ \dot{\mathbf{p}}_2 \\ \dot{\mathbf{p}}_3 \end{bmatrix} + \begin{bmatrix} \mathbf{T}_{11} & \mathbf{T}_{12} & \mathbf{T}_{13} \\ \mathbf{T}_{21} & \mathbf{T}_{22} & \mathbf{T}_{23} \\ \mathbf{T}_{31} & \mathbf{T}_{32} & \mathbf{T}_{33} + \mathbf{J}_3 \end{bmatrix} \begin{bmatrix} \mathbf{p}_1 \\ \mathbf{p}_2 \\ \mathbf{p}_3 \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \breve{\mathbf{q}}_{well} \\ \mathbf{J}_3\breve{\mathbf{p}}_{well} \end{bmatrix}. \tag{3.89}$$

An important aspect of the introduction of the well model is that, compared to matrix $\mathbf{T}$ in equation (3.52), the transmissibility matrix in equation (3.89) has elements added to its main diagonal. We will discuss the consequences of this addition in Chapter 4.

Free bottom hole pressures and well flow rates

The flow rates $\bar{\mathbf{q}}_{well} = \mathbf{q}_3$ in the wells where the bottom hole pressures have been prescribed can be obtained directly from equation (3.87) as

$$\bar{\mathbf{q}}_{well} = \mathbf{J}_3\left(\breve{\mathbf{p}}_{well} - \mathbf{p}_3\right). \tag{3.90}$$

To compute the bottom hole pressures $\bar{\mathbf{p}}_{well}$ in the wells where the flow rates have been prescribed we need an additional diagonal matrix $\mathbf{J}_{q,2}$ of well indices J_q. We can then write

$$\breve{\mathbf{q}}_{well} = \mathbf{J}_2 \left(\bar{\mathbf{p}}_{well} - \mathbf{p}_2 \right) , \tag{3.91}$$

from which we obtain

$$\bar{\mathbf{p}}_{well} = \mathbf{p}_2 + \mathbf{J}_2^{-1} \breve{\mathbf{q}}_{well} \ . \tag{3.92}$$

State space representation

If we define the (partitioned) state, input and output vectors

$$\mathbf{x} \triangleq \mathbf{p} \equiv \begin{bmatrix} \mathbf{p}_1 \\ \mathbf{p}_2 \\ \mathbf{p}_3 \end{bmatrix}, \ \mathbf{u} \triangleq \begin{bmatrix} \breve{\mathbf{q}}_{well} \\ \breve{\mathbf{p}}_{well} \end{bmatrix}, \ \mathbf{y} \triangleq \begin{bmatrix} \bar{\mathbf{p}}_{well} \\ \bar{\mathbf{q}}_{well} \end{bmatrix}, \tag{3.93, 3.94, 3.95}$$

equations (3.89), (3.90) and (3.92) can be rewritten in state space form as

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} , \ \ \mathbf{y} = \mathbf{C}\mathbf{x} + \mathbf{D}\mathbf{u} \ , \tag{3.96, 3.97}$$

where the matrices are defined as

$$\mathbf{A} \triangleq -\begin{bmatrix} \mathbf{V}_{11}^{-1}\mathbf{T}_{11} & \mathbf{V}_{11}^{-1}\mathbf{T}_{12} & \mathbf{V}_{11}^{-1}\mathbf{T}_{13} \\ \mathbf{V}_{22}^{-1}\mathbf{T}_{21} & \mathbf{V}_{22}^{-1}\mathbf{T}_{22} & \mathbf{V}_{22}^{-1}\mathbf{T}_{23} \\ \mathbf{V}_{33}^{-1}\mathbf{T}_{31} & \mathbf{V}_{33}^{-1}\mathbf{T}_{32} & \mathbf{V}_{33}^{-1}\left(\mathbf{T}_{33} + \mathbf{J}_3\right) \end{bmatrix}, \ \mathbf{B} \triangleq \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{V}_{22}^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{V}_{33}^{-1}\mathbf{J}_3 \end{bmatrix},$$

$$\mathbf{C} \triangleq \begin{bmatrix} \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & -\mathbf{J}_3 \end{bmatrix}, \ \mathbf{D} \triangleq \begin{bmatrix} \mathbf{J}_2^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{J}_3 \end{bmatrix}. \tag{3.98, 3.99, 3.100, 3.101}$$

Here, matrix **D** is an example of a *direct throughput matrix* which couples the input **u** directly to the output **y**.

3.3.5 Example 1 continued – Well model

In Example 1, discussed in Section 3.3.2, we fixed the flow rates in both wells. Here, we fix the bottom hole pressure of the producer in grid block 6 as: $p_{wf} = 28.00 \times 10^6$ Pa (4061 psi), while we choose an injection rate in block 1 as $q_1 = 0.01$ m^3/s (864 m^3/d, 5434 bpd), where we use the convention that positive flow rates indicate injection. Because we only have one well with a prescribed pressure and one with a prescribed rate, we have

$$\breve{\mathbf{p}}_{well} = \left[28.00 \times 10^6 \right] , \ \breve{\mathbf{q}}_{well} = \left[0.01 \right] . \tag{3.102), (3.103}$$

Correspondingly, the matrices $\mathbf{J}_3$ and $\mathbf{J}_2$ contain only one element each. Using the data for the near-well permeabilities as derived in Section 2.3.6 they become

$$\mathbf{J}_3 = \left[3.72 \times 10^{-9} \right] , \ \ \mathbf{J}_2 = \left[3.72 \times 10^{-8} \right] . \tag{3.104, 3.105}$$

3.3.6 Elimination of prescribed pressures*

An alternative way to implement a prescribed pressure is through directly prescribing the grid block pressure. This means that one of the state variables is fixed, and may be eliminated from the system equations. To illustrate this method, we start again from the partitioned system equation (3.74). We indicate prescribed values with a '˘' above the variable, and free values with a '–':

$$\begin{bmatrix} \mathbf{V}_{11} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{V}_{22} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{V}_{33} \end{bmatrix} \begin{bmatrix} \dot{\bar{\mathbf{p}}}_1 \\ \dot{\bar{\mathbf{p}}}_2 \\ \dot{\breve{\mathbf{p}}}_3 \end{bmatrix} + \begin{bmatrix} \mathbf{T}_{11} & \mathbf{T}_{12} & \mathbf{T}_{13} \\ \mathbf{T}_{21} & \mathbf{T}_{22} & \mathbf{T}_{23} \\ \mathbf{T}_{31} & \mathbf{T}_{32} & \mathbf{T}_{33} \end{bmatrix} \begin{bmatrix} \bar{\mathbf{p}}_1 \\ \bar{\mathbf{p}}_2 \\ \breve{\mathbf{p}}_3 \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \breve{\mathbf{q}}_2 \\ \bar{\mathbf{q}}_3 \end{bmatrix} . \tag{3.106}$$

From the first two rows of matrix equation (3.106) we find the system of differential equations

$$\begin{bmatrix} \mathbf{V}_{11} & \mathbf{0} \\ \mathbf{0} & \mathbf{V}_{22} \end{bmatrix} \begin{bmatrix} \dot{\bar{\mathbf{p}}}_1 \\ \dot{\bar{\mathbf{p}}}_2 \end{bmatrix} = -\begin{bmatrix} \mathbf{T}_{11} & \mathbf{T}_{12} \\ \mathbf{T}_{21} & \mathbf{T}_{22} \end{bmatrix} \begin{bmatrix} \bar{\mathbf{p}}_1 \\ \bar{\mathbf{p}}_2 \end{bmatrix} \underbrace{- \begin{bmatrix} \mathbf{T}_{13} \\ \mathbf{T}_{23} \end{bmatrix} \breve{\mathbf{p}}_3 + \begin{bmatrix} \mathbf{0} \\ \breve{\mathbf{q}}_2 \end{bmatrix}}_{\text{prescribed}} . \tag{3.107}$$

Because we eliminated the prescribed pressures, the length of the pressure vector has been reduced. From the third row of equation (3.106) it follows that

$$\bar{\mathbf{q}}_3 = \begin{bmatrix} \mathbf{T}_{31} & \mathbf{T}_{32} \end{bmatrix} \begin{bmatrix} \bar{\mathbf{p}}_1 \\ \bar{\mathbf{p}}_2 \end{bmatrix} + \underbrace{\mathbf{T}_{33}\breve{\mathbf{p}}_3 + \mathbf{V}_{33}\dot{\breve{\mathbf{p}}}}_{\text{prescribed}} , \tag{3.108}$$

where $\bar{\mathbf{q}}_3$ represents the free flow rates in the wells where the pressures have been prescribed. Apparently the price to pay for the reduced length of the pressure vector is an increase in the number of input parameters to compute the free flow rates in case of time-varying prescribed pressures. Equations (3.107) and (3.108) can be rewritten in partitioned state space form, as in equations (3.96) and (3.97), through definition of

$$\mathbf{A} \triangleq -\begin{bmatrix} \mathbf{V}_{11}^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{V}_{22}^{-1} \end{bmatrix} \begin{bmatrix} \mathbf{T}_{11} & \mathbf{T}_{12} \\ \mathbf{T}_{21} & \mathbf{T}_{22} \end{bmatrix}, \quad \mathbf{x} \triangleq \begin{bmatrix} \dot{\bar{\mathbf{p}}}_1 \\ \dot{\bar{\mathbf{p}}}_2 \end{bmatrix}, \quad \mathbf{B} \triangleq \begin{bmatrix} \mathbf{0} & -\mathbf{V}_{11}^{-1}\mathbf{T}_{13} & \mathbf{0} \\ \mathbf{V}_{22}^{-1} & -\mathbf{V}_{22}^{-1}\mathbf{T}_{23} & \mathbf{0} \end{bmatrix}, \quad \mathbf{u} \triangleq \begin{bmatrix} \breve{\mathbf{q}}_2 \\ \breve{\mathbf{p}}_3 \\ \dot{\breve{\mathbf{p}}}_3 \end{bmatrix},$$

$$\mathbf{y} \triangleq \begin{bmatrix} \bar{\mathbf{p}}_2 \\ \bar{\mathbf{q}}_3 \end{bmatrix}, \quad \mathbf{C} \triangleq \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ \mathbf{T}_{31} & \mathbf{T}_{32} \end{bmatrix}, \quad \mathbf{D} \triangleq \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{T}_{33} & \mathbf{V}_{33} \end{bmatrix}.$$

(3.109, 3.110, 3.111, 3.112, 3.113, 3.114, 3.115)

As before, we have chosen the output vector **y** such that it contains the free pressures and flow rates in the wells. However the input vector **u** now not only contains the prescribed flow rates and pressures in the wells, but also the time derivatives of the pressures. This technique to eliminate the prescribed state variables is mainly of theoretical value, and is not commonly used, if at all, in reservoir engineering applications.

3.3.7 System energy*

The *energy balance* in flow through porous media is governed by three components:

1) *Potential energy* in the form of compressed fluids inside compressed rock and in the form of elevated fluid mass.
2) *Energy dissipation* caused by resistance to fluids flowing through the pore network.
3) *Energy transport* through the system boundaries in the form of *work* done by injecting or producing fluids under a pressure differential in the wells.

Note that we do not consider *kinetic energy*, because of the assumption that inertia forces may be neglected due to the very low flow velocities inside the pores[§]. Moreover, we will not take into account the effect of elevation on the potential energy because we restrict the theory and examples to two-dimensional horizontal reservoirs where gravity forces can be neglected.[‡] Finally, we maintain our earlier assumption of isothermal conditions, which implies that the heat generated by energy dissipation is instantaneously transferred to the surroundings (i.e. to outside the reservoir boundaries) such that there is no increase in reservoir temperature.

*Potential energy**

Starting from the assumption of a small, pressure-independent, total compressibility c_t, the fluid volume $\bar{V}(p)$ in a single grid block with volume V is expressed as[†]

$$\bar{V}(p) = V\phi_0\left[1 + c_t\left(p - p_0\right)\right], \tag{3.116}$$

where p_0 is a reference pressure, and ϕ_0 the corresponding porosity. The difference in potential energy when the fluids in the grid block experience a pressure increase from p_0 to p can therefore be expressed as

$$E_{pot}(p) = \int_{p_0}^{p} \frac{\partial \bar{V}(\pi)}{\partial \pi}\pi d\pi = \int_{p_0}^{p} V\phi_0 c_t \pi d\pi = \tfrac{1}{2}V\phi_0 c_t\left(p^2 - p_0^2\right) . \tag{3.117}$$

If we choose the reference value for E_{pot} as zero at the reference pressure (which we may conveniently take to be the initial reservoir pressure p_R), we can compute the total potential energy in a reservoir model through summation over the grid blocks according to

$$E_{pot,tot}(t) = \sum_{i=1}^{n_{gb}} \tfrac{1}{2}V_i\phi_i c_t p_i^2(t) , \tag{3.118}$$

where n_{gb} is the total number of grid blocks, and where V_i, ϕ_i and p_i are the grid block volumes, porosities and pressures respectively, with only the pressures being a function of time.

*Disspation energy**

The energy dissipated per unit time by resistance to flow through a grid block boundary can be expressed as

$$\frac{dE_{dis}}{dt} = -\tilde{q}\Delta p = T\Delta p^2 , \tag{3.119}$$

where $\tilde{q}$ is the volumetric flowrate, Δp is the pressure difference between the two grid block centers, and T is the grid block transmissibility as defined in equation (2.30). In addition to

---

[§] I.e. we do not consider kinetic energy at a macroscopic level. We do take into account energy dissipation, which is the change of mechanical energy into thermal energy, or heat, and which at an atomic level can be interpreted as kinetic energy again.

[‡] Elevation-related potential energy plays an important role in well bore flow. Most reservoirs have enough potential energy, at least initially, to *lift* the oil to surface naturally in the production wells. This lift effect is in most cases caused by the difference in density between oil and water, such that in an oil-filled well that drains a hydrostatically-pressured reservoir the oil will be lifted to surface because of elevation-related potential energy.

[†] For a detailed derivation of pressure-related potential energy for the case of pressure-dependent rock and fluid compressibilities, see Hubbert (1940).

dissipation at the grid block boundaries, a large amount of energy is dissipated in the near-well bore region where large pressure gradients are present. The energy dissipated per unit time by resistance to flow in a well grid block can be expressed as

$$\frac{dE_{dis}}{dt} = q_{well}\left(p_{well} - p_{gb}\right) = J_{well}\left(p_{well} - p_{gb}\right)^2 \ , \tag{3.120}$$

where q_{well} is the well flow rate (positive for injection), p_{well} is the flowing bottom hole pressure, p_{gb} is the well grid block pressure and J_{well} is the well index, as defined in equation (2.44). The total amount of energy dissipated per unit time in a reservoir model is therefore obtained through summation over all grid block connectivities and all wells as

$$\frac{dE_{dis,tot}(t)}{dt} = -\sum_{j=1}^{n_{con}} T_j(t)\Delta p_j^2(t) + \sum_{k=1}^{n_{well}} J_{well,k}(t)\left[p_{well,k}(t) - p_{gb,k}(t)\right]^2 \ , \tag{3.121}$$

where n_{con} is the number of connectivities, n_{well} is the number of wells, and where the transmissibilities, well indices and pressure drops may be functions of time.

*Work done by wells**

The *power* (i.e. the work per unit time[†]) delivered by fluids injected into the reservoir can be expressed as

$$P_{well} = q_{well}\, p_{well} \ . \tag{3.122}$$

The same equation holds for production wells, where we use the convention that flowrates in the producers have negative values. The total power delivered by all injectors and producers to the reservoir is therefore given by

$$P_{well,tot}(t) = \sum_{k=1}^{n_{well}} q_{well,k}(t) \times p_{well,k}(t) \ . \tag{3.123}$$

*Energy balance**

The total energy balance for the reservoir over a time interval $\Delta t = t_2 - t_1$ can now be expressed as

$$\underbrace{E_{pot,tot}(t_2) - E_{pot,tot}(t_1)}_{\text{accumulation}} + \underbrace{\int_{t_1}^{t_2} \frac{dE_{dis,tot}(t)}{dt}\, dt}_{\textit{dissipation}} = \underbrace{\int_{t_1}^{t_2} P_{well,tot}(t)\, dt}_{\textit{source}} \ . \tag{3.124}$$

Note that the potential energy accumulation has simply been expressed as the difference between the values at the begin and end times which illustrates that potential energy is not dependent on the pressure history, i.e. it is path-independent. This is unlike the energy lost by disippation and and gained by work, which are both (pressure) path-dependent as indicated by the integrals which need to be evaluated over the entire time interval. Alternatively, the the energy balance per unit time, i.e. the *power balance*, can be expressed as

[†] Energy delivered to a system through mechanical or hydraulic action is often referred to as *work*. Work (or energy) per unit time is then called *power*. In strict SI units time is expressed in s (seconds), energy and work in J (Joule) and power in W (Watt), such that 1 W is equal to 1 J/s.

$$\frac{dE_{pot,tot}(t)}{dt}+\frac{dE_{dis,tot}(t)}{dt}=P_{well,tot}(t)\ . \tag{3.125}$$

or, according to equations (3.118), (3.121) and (3.123), as

$$\begin{aligned}&\sum_{i=1}^{n_{gb}}V_i\phi_i c_t p_i(t)\frac{dp_i(t)}{dt}+\sum_{j=1}^{n_{con}}T_j(t)\Delta p_j^2(t)+\sum_{k=1}^{n_{well}}J_{well,k}(t)\left[p_{well,k}(t)-p_{gb,k}(t)\right]^2\\&=\sum_{k=1}^{n_{well}}q_{well,k}(t)\times p_{well,k}(t)\,.\end{aligned} \tag{3.126}$$

Expression (3.126) can also be written in matrix-vector notation as

$$\begin{aligned}&\begin{bmatrix}\mathbf{p}_1 & \mathbf{p}_2 & \mathbf{p}_3\end{bmatrix}\begin{bmatrix}\mathbf{V}_{11} & \mathbf{0} & \mathbf{0}\\ \mathbf{0} & \mathbf{V}_{22} & \mathbf{0}\\ \mathbf{0} & \mathbf{0} & \mathbf{V}_{33}\end{bmatrix}\begin{bmatrix}\dot{\mathbf{p}}_1\\ \dot{\mathbf{p}}_2\\ \dot{\mathbf{p}}_3\end{bmatrix}\\&+\begin{bmatrix}(\mathbf{p}_1-\mathbf{p}_{av}) & (\mathbf{p}_2-\mathbf{p}_{av}) & (\mathbf{p}_3-\mathbf{p}_{av})\end{bmatrix}\begin{bmatrix}\mathbf{T}_{11} & \mathbf{T}_{12} & \mathbf{T}_{13}\\ \mathbf{T}_{21} & \mathbf{T}_{22} & \mathbf{T}_{23}\\ \mathbf{T}_{31} & \mathbf{T}_{32} & \mathbf{T}_{33}\end{bmatrix}\begin{bmatrix}\mathbf{p}_1\\ \mathbf{p}_2\\ \mathbf{p}_3\end{bmatrix}\\&+\begin{bmatrix}\mathbf{0} & (\overline{\mathbf{p}}_{well}-\mathbf{p}_2) & (\breve{\mathbf{p}}_{well}-\mathbf{p}_3)\end{bmatrix}\begin{bmatrix}\mathbf{0} & \mathbf{0} & \mathbf{0}\\ \mathbf{0} & \mathbf{J}_2 & \mathbf{0}\\ \mathbf{0} & \mathbf{0} & \mathbf{J}_3\end{bmatrix}\begin{bmatrix}\mathbf{0}\\ \overline{\mathbf{p}}_{well}-\mathbf{p}_2\\ \breve{\mathbf{p}}_{well}-\mathbf{p}_3\end{bmatrix}\\&=\begin{bmatrix}\mathbf{0} & \overline{\mathbf{p}}_{well} & \overline{\mathbf{q}}_{well}\end{bmatrix}\begin{bmatrix}\mathbf{0}\\ \breve{\mathbf{q}}_{well}\\ \breve{\mathbf{p}}_{well}\end{bmatrix},\end{aligned} \tag{3.127}$$

where we have used the partitioned vectors and matrices as introduced in Sections 3.3.3 and 3.3.4. The vector $\mathbf{p}_{av}$ represents the time-dependent average reservoir pressure defined as

$$\mathbf{p}_{av}(t)\triangleq\frac{1}{n_{gb}}\sum_{i=1}^{n_{gb}}p_i(t)\begin{bmatrix}1\\1\\\vdots\\1\end{bmatrix}, \tag{3.128}$$

with a number of elements as appropriate to match those of $\mathbf{p}_1$, $\mathbf{p}_2$ and $\mathbf{p}_3$. The equivalence of equations (3.126) and (3.127) can be confirmed by inspection of the matrices $\mathbf{V}$, $\mathbf{T}$ and $\mathbf{J}_3$ and the underlying matrices as defined in equation (2.31).

*Minimum energy interpretation**

With the aid of relationships (3.90) and (3.91) we can rewrite the power balance (3.127) as

$$
\begin{aligned}
&\begin{bmatrix}\mathbf{p}_1 & \mathbf{p}_2 & \mathbf{p}_3\end{bmatrix}\begin{bmatrix}\mathbf{V}_{11} & \mathbf{0} & \mathbf{0}\\ \mathbf{0} & \mathbf{V}_{22} & \mathbf{0}\\ \mathbf{0} & \mathbf{0} & \mathbf{V}_{33}\end{bmatrix}\begin{bmatrix}\dot{\mathbf{p}}_1\\ \dot{\mathbf{p}}_2\\ \dot{\mathbf{p}}_3\end{bmatrix}\\
&+\begin{bmatrix}\left(\mathbf{p}_1-\mathbf{p}_{av}\right) & \left(\mathbf{p}_2-\mathbf{p}_{av}\right) & \left(\mathbf{p}_3-\mathbf{p}_{av}\right)\end{bmatrix}\begin{bmatrix}\mathbf{T}_{11} & \mathbf{T}_{12} & \mathbf{T}_{13}\\ \mathbf{T}_{21} & \mathbf{T}_{22} & \mathbf{T}_{23}\\ \mathbf{T}_{31} & \mathbf{T}_{32} & \mathbf{T}_{33}\end{bmatrix}\begin{bmatrix}\mathbf{p}_1\\ \mathbf{p}_2\\ \mathbf{p}_3\end{bmatrix}\\
&+\begin{bmatrix}\mathbf{0} & \left(\bar{\mathbf{p}}_{well}-\mathbf{p}_2\right) & \left(\breve{\mathbf{p}}_{well}-\mathbf{p}_3\right)\end{bmatrix}\begin{bmatrix}\mathbf{0}\\ \breve{\mathbf{q}}_{well}\\ \bar{\mathbf{q}}_{well}\end{bmatrix}\\
&=\begin{bmatrix}\mathbf{0} & \bar{\mathbf{p}}_{well} & \bar{\mathbf{q}}_{well}\end{bmatrix}\begin{bmatrix}\mathbf{0}\\ \breve{\mathbf{q}}_{well}\\ \breve{\mathbf{p}}_{well}\end{bmatrix},
\end{aligned}
\tag{3.129}
$$

from which follows

$$
\begin{aligned}
&\begin{bmatrix}\mathbf{p}_1 & \mathbf{p}_2 & \mathbf{p}_3\end{bmatrix}\begin{bmatrix}\mathbf{V}_{11} & \mathbf{0} & \mathbf{0}\\ \mathbf{0} & \mathbf{V}_{22} & \mathbf{0}\\ \mathbf{0} & \mathbf{0} & \mathbf{V}_{33}\end{bmatrix}\begin{bmatrix}\dot{\mathbf{p}}_1\\ \dot{\mathbf{p}}_2\\ \dot{\mathbf{p}}_3\end{bmatrix}\\
&+\begin{bmatrix}\left(\mathbf{p}_1-\mathbf{p}_{av}\right) & \left(\mathbf{p}_2-\mathbf{p}_{av}\right) & \left(\mathbf{p}_3-\mathbf{p}_{av}\right)\end{bmatrix}\begin{bmatrix}\mathbf{T}_{11} & \mathbf{T}_{12} & \mathbf{T}_{13}\\ \mathbf{T}_{21} & \mathbf{T}_{22} & \mathbf{T}_{23}\\ \mathbf{T}_{31} & \mathbf{T}_{32} & \mathbf{T}_{33}\end{bmatrix}\begin{bmatrix}\mathbf{p}_1\\ \mathbf{p}_2\\ \mathbf{p}_3\end{bmatrix}\\
&=\begin{bmatrix}\mathbf{0} & \mathbf{p}_2 & \mathbf{p}_3\end{bmatrix}\begin{bmatrix}\mathbf{0}\\ \breve{\mathbf{q}}_{well}\\ \bar{\mathbf{q}}_{well}\end{bmatrix}.
\end{aligned}
\tag{3.130}
$$

Equation (3.130) is again an expression for the power balance in the system, but now expressed solely in terms of the state variables[†] $\mathbf{p}_1$, $\mathbf{p}_2$ and $\mathbf{p}_3$. It can be written compactly as $dE_{sys}(\mathbf{p}_1,\mathbf{p}_2,\mathbf{p}_3)/dt = P_{well}(\mathbf{p}_1,\mathbf{p}_2,\mathbf{p}_3)$, where E_{sys} is the system as governed by the state variables proper. Equation (3.130) is a single scalar equation in the n_{gb} state variables, and therefore does not have a unique solution. However, using thermodynamic arguments it can be argued that all natural systems tend to organize themselves in such a way that they minimize the amount of energy required to maintain equilibrium between internal and external forces; see e.g. xxx. In our particular case of a system comprising flow through a porous medium this implies that the values of the state variables $\mathbf{p}_1$, $\mathbf{p}_2$ and $\mathbf{p}_3$ will be such that the power flow through the system becomes minimal i.e. that the first derivates of P_{well} with respect to the state variables, and therefore also the first derivatives of dE_{sys}/dt with respect to the state variables, become zero. Taking derivatives of equation (3.130), setting the results equal to zero and combining them in matrix form results in

[†] In comparison, equation (3.127) was also a function of the well bore pressures $\breve{\mathbf{p}}_{well}$ and $\bar{\mathbf{p}}_{well}$.

$$\begin{bmatrix} \mathbf{V}_{11} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{V}_{22} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{V}_{33} \end{bmatrix} \begin{bmatrix} \dot{\mathbf{p}}_1 \\ \dot{\mathbf{p}}_2 \\ \dot{\mathbf{p}}_3 \end{bmatrix} + \begin{bmatrix} \mathbf{T}_{11} & \mathbf{T}_{12} & \mathbf{T}_{13} \\ \mathbf{T}_{21} & \mathbf{T}_{22} & \mathbf{T}_{23} \\ \mathbf{T}_{31} & \mathbf{T}_{32} & \mathbf{T}_{33} \end{bmatrix} \begin{bmatrix} \mathbf{p}_1 \\ \mathbf{p}_2 \\ \mathbf{p}_3 \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \breve{\mathbf{q}}_{well} \\ \bar{\mathbf{q}}_{well} \end{bmatrix} . \tag{3.131}$$

Here we made use of the fact that

$$\frac{d}{d\mathbf{p}}\left[\left(\mathbf{p}-\mathbf{p}_{av}\right)\mathbf{T}\mathbf{p}\right] = 2\mathbf{T}\mathbf{p} - \underbrace{\frac{d\mathbf{p}_{av}}{d\mathbf{p}}}_{\mathbf{I}}\mathbf{T}\mathbf{p} - \underbrace{\mathbf{T}\mathbf{p}_{av}}_{\mathbf{0}} = \mathbf{T}\mathbf{p} \ , \tag{3.132}$$

where we used the compact notation $\mathbf{p} = [(\mathbf{p}_1)^T (\mathbf{p}_2)^T (\mathbf{p}_3)^T]^T$, etc., as introduced in Section 3.3.3, leaving out the supercripted stars for clarity. Using equation (3.90), we can finally rewrite equation (3.131) as

$$\begin{bmatrix} \mathbf{V}_{11} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{V}_{22} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{V}_{33} \end{bmatrix} \begin{bmatrix} \dot{\mathbf{p}}_1 \\ \dot{\mathbf{p}}_2 \\ \dot{\mathbf{p}}_3 \end{bmatrix} + \begin{bmatrix} \mathbf{T}_{11} & \mathbf{T}_{12} & \mathbf{T}_{13} \\ \mathbf{T}_{21} & \mathbf{T}_{22} & \mathbf{T}_{23} \\ \mathbf{T}_{31} & \mathbf{T}_{32} & \mathbf{T}_{33}+\mathbf{J}_3 \end{bmatrix} \begin{bmatrix} \mathbf{p}_1 \\ \mathbf{p}_2 \\ \mathbf{p}_3 \end{bmatrix} = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{J}_3 \end{bmatrix} \begin{bmatrix} \mathbf{0} \\ \breve{\mathbf{q}}_{well} \\ \breve{\mathbf{p}}_{well} \end{bmatrix} . \tag{3.133}$$

Equation (3.133) is completely identical to system equations (3.89) which were derived from balance equations for mass and momentum only. The alternative way to derive the system equations using the concept of energy, as described for porous medium flow in this section, is frequently used in the fields of theoretical and applied mechanics; see e.g. Langhaar (1962) and Lanczos (1970). Closely related are other *energy methods* also known as *variational methods*, which are used to compute approximate solutions for applied mechanics problems in complex-shaped domains. In particular, they often form the basis to derive numerical approximations using the finite element method; see e.g. Zienckiewicz and Taylor (1989). Direct use of energy methods in porous media flow does not seem to have an advantage over the conventional direct methods, and has therefore scarcely been described in the literature. An exception is the paper by Karney and Seneviratne (1991) who propose to use energy concepts for adaptive time step control in numerical simulation.

3.4 Two-phase flow

3.4.1 System equations

Nonlinear equations

As a next step we consider a simplified description of two-phase flow of oil and water, as derived in some detail in Section 2.4. After semi-discretization of the equations in we obtain the following system of nonlinear first-order differential equations,

$$\underbrace{\begin{bmatrix} \mathbf{V}_{wp}(\mathbf{s}) & \mathbf{V}_{ws} \\ \mathbf{V}_{op}(\mathbf{s}) & \mathbf{V}_{os} \end{bmatrix}}_{\mathbf{V}} \begin{bmatrix} \dot{\mathbf{p}} \\ \dot{\mathbf{s}} \end{bmatrix} + \underbrace{\begin{bmatrix} \mathbf{T}_w(\mathbf{s}) & \mathbf{0} \\ \mathbf{T}_o(\mathbf{s}) & \mathbf{0} \end{bmatrix}}_{\mathbf{T}} \begin{bmatrix} \mathbf{p} \\ \mathbf{s} \end{bmatrix} = \underbrace{\begin{bmatrix} \mathbf{F}_w(\mathbf{s}) \\ \mathbf{F}_o(\mathbf{s}) \end{bmatrix}}_{\mathbf{F}} \mathbf{q}_{well,t} , \tag{3.134}$$

where **p** and **s** are vectors of pressures p_o and water saturations S_w in the grid block centers, **V** is the accumulation matrix (with entries that are functions of the porosity ϕ, and the oil, water and rock compressibilities c_o, c_w and c_r), **T** is the transmissibility matrix (with entries that are functions of the rock permeabilities k, the oil and water relative permeabilities k_{ro} and k_{rw} and the oil and water viscosities μ_o and μ_w), **F** is the fractional flow matrix (with entries that have

functional dependencies similar to those of $\mathbf{T}$), and $\mathbf{q}_{well,t}$ is a vector of total well flow rates with non-zero values in those elements that correspond to grid blocks penetrated by a well. The nonlinearities in equation (3.134) result from various sources; see also Section 2.4.10. If the oil and water phases have different compressibilities, the accumulation terms $\mathbf{V}_{wp}(\mathbf{s})$ and $\mathbf{V}_{op}(\mathbf{s})$ are a (weak) function of the saturations because the liquid compressibility is a saturation-weighted average of the oil and water compressibilities. Moreover, the porosity and compressibility values in these terms may be a weak function of pressure, but we do not take this effect into account in the examples in this text. The transmissibility terms $\mathbf{T}_w(\mathbf{s})$ and $\mathbf{T}_o(\mathbf{s})$ are a much stronger function of the saturations, because the relative permeabilities for oil and water are strongly saturation-dependent. The viscosities may also be weakly pressure-dependent, but, yet again, the pressure dependency is disregarded in the examples. Finally, matrices $\mathbf{F}_o(\mathbf{s})$ and $\mathbf{F}_w(\mathbf{s})$ contain saturation-dependent terms that relate the oil and water flow rates in the wells to the total flow rates.

Well model

In practice the source terms are often not the flow rates in the wells but rather the pressures. This can be accounted for by rewriting equation (3.134) in partitioned form as

$$\left[\begin{array}{ccc|ccc} \mathbf{V}_{wp,11} & \mathbf{0} & \mathbf{0} & \mathbf{V}_{ws,11} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{V}_{wp,22} & \mathbf{0} & \mathbf{0} & \mathbf{V}_{ws,22} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{V}_{wp,33} & \mathbf{0} & \mathbf{0} & \mathbf{V}_{ws,33} \\ \hline \mathbf{V}_{op,11} & \mathbf{0} & \mathbf{0} & \mathbf{V}_{os,11} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{V}_{op,22} & \mathbf{0} & \mathbf{0} & \mathbf{V}_{os,22} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{V}_{op,33} & \mathbf{0} & \mathbf{0} & \mathbf{V}_{os,33} \end{array}\right] \left[\begin{array}{c} \dot{\mathbf{p}}_1 \\ \dot{\mathbf{p}}_2 \\ \dot{\mathbf{p}}_3 \\ \hline \dot{\mathbf{s}}_1 \\ \dot{\mathbf{s}}_2 \\ \dot{\mathbf{s}}_3 \end{array}\right] +$$

$$\left[\begin{array}{ccc|ccc} \mathbf{T}_{w,11} & \mathbf{T}_{w,12} & \mathbf{T}_{w,13} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{T}_{w,21} & \mathbf{T}_{w,22} & \mathbf{T}_{w,23} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{T}_{w,31} & \mathbf{T}_{w,32} & \mathbf{T}_{w,33} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \hline \mathbf{T}_{o,11} & \mathbf{T}_{o,12} & \mathbf{T}_{o,13} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{T}_{o,21} & \mathbf{T}_{o,22} & \mathbf{T}_{o,23} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{T}_{o,31} & \mathbf{T}_{o,32} & \mathbf{T}_{o,33} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{array}\right] \left[\begin{array}{c} \mathbf{p}_1 \\ \mathbf{p}_2 \\ \mathbf{p}_3 \\ \hline \mathbf{s}_1 \\ \mathbf{s}_2 \\ \mathbf{s}_3 \end{array}\right] = \left[\begin{array}{ccc} \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{F}_{w,22} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{F}_{w,33} \\ \hline \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{F}_{o,22} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{F}_{o,33} \end{array}\right] \left[\begin{array}{c} \mathbf{0} \\ \breve{\mathbf{q}}_{well,t} \\ \mathbf{J}_3\left(\breve{\mathbf{p}}_{well} - \mathbf{p}_3\right) \end{array}\right]. \tag{3.135}$$

Here, the elements of vector $\mathbf{p}_1$ are the pressures in those grid blocks that are not penetrated by a well. The elements of $\mathbf{p}_2$ are the pressures in the blocks where the source terms are prescribed total well flow rates $\breve{\mathbf{q}}_{well,t}$, and those of $\mathbf{p}_3$ are the pressures in the blocks where the source terms are obtained through prescription of the bottom hole pressures $\breve{\mathbf{p}}_{well}$ with the aid of a diagonal matrix of well indices $\mathbf{J}_3$. To compute the oil and water flow rates in the wells with prescribed pressures we use the well model

$$\left[\begin{array}{c} \bar{\mathbf{q}}_{well,w} \\ \bar{\mathbf{q}}_{well,o} \end{array}\right] = \left[\begin{array}{c} \mathbf{F}_{w,33} \\ \mathbf{F}_{o,33} \end{array}\right] \mathbf{J}_3\left(\breve{\mathbf{p}}_{well} - \mathbf{p}_3\right) . \tag{3.136}$$

To compute the bottom hole pressures $\bar{\mathbf{p}}_{well}$ in the wells with prescribed total flow rates we need an additional diagonal matrix $\mathbf{J}_{q,2}$ of well indices such that

$$\breve{\mathbf{q}}_{well,t} = \mathbf{J}_2\left(\bar{\mathbf{p}}_{well} - \mathbf{p}_2\right) , \tag{3.137}$$

from which we obtain

$$\bar{\mathbf{p}}_{well} = \mathbf{J}_2^{-1}\breve{\mathbf{q}}_{well,t} - \mathbf{p}_2 \ . \tag{3.138}$$

State space form

To bring these equations in state space form we define the state vector **x**, input vector **u** and output vector **y** as

$$\mathbf{u} \triangleq \begin{bmatrix} \breve{\mathbf{q}}_{well,t} \\ \breve{\mathbf{p}}_{well} \end{bmatrix}, \ \mathbf{x} \triangleq \begin{bmatrix} \mathbf{p} \\ \mathbf{s} \end{bmatrix} = \begin{bmatrix} \mathbf{p}_1 \\ \mathbf{p}_2 \\ \mathbf{p}_3 \\ \hline \mathbf{s}_1 \\ \mathbf{s}_2 \\ \mathbf{s}_3 \end{bmatrix}, \ \mathbf{y} \triangleq \begin{bmatrix} \bar{\mathbf{p}}_{well} \\ \bar{\mathbf{q}}_{well,w} \\ \bar{\mathbf{q}}_{well,o} \end{bmatrix} . \tag{3.139, 3.140, 3.141}$$

Equations (3.135), (3.136) and (3.138) can then be rewritten in nonlinear state space form

$$\dot{\mathbf{x}} = \mathbf{f}\left(\mathbf{u},\mathbf{x}\right) \ , \ \ \mathbf{y} = \mathbf{h}\left(\mathbf{u},\mathbf{x}\right) \ , \tag{3.142, 3.143}$$

where the functions **f** and **h** are defined as

$$\mathbf{f} \triangleq \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \ , \ \ \mathbf{h} \triangleq \mathbf{C}\mathbf{x} + \mathbf{D}\mathbf{u} \tag{3.144, 3.145}$$

with state-dependent secant matrices **A**(**x**), **B**(**x**), **C**(**x**) and **D**(**x**) given by

$$\mathbf{A} \triangleq -\mathbf{V}^{-1}\underbrace{\left[\begin{array}{ccc|ccc} \mathbf{T}_{w,11} & \mathbf{T}_{w,12} & \mathbf{T}_{w,13} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{T}_{w,21} & \mathbf{T}_{w,22} & \mathbf{T}_{w,23} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{T}_{w,31} & \mathbf{T}_{w,32} & \mathbf{T}_{w,33}+\mathbf{F}_{w,33}\mathbf{J}_3 & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \hline \mathbf{T}_{o,11} & \mathbf{T}_{o,12} & \mathbf{T}_{o,13} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{T}_{o,21} & \mathbf{T}_{o,22} & \mathbf{T}_{o,23} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{T}_{o,31} & \mathbf{T}_{o,32} & \mathbf{T}_{o,33}+\mathbf{F}_{o,33}\mathbf{J}_3 & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{array}\right]}_{-\hat{\mathbf{A}}}, \quad \mathbf{B} \triangleq \mathbf{V}^{-1}\underbrace{\left[\begin{array}{cc} \mathbf{0} & \mathbf{0} \\ \mathbf{F}_{w,22} & \mathbf{0} \\ \mathbf{0} & \mathbf{F}_{w,33}\mathbf{J}_3 \\ \hline \mathbf{0} & \mathbf{0} \\ \mathbf{F}_{o,22} & \mathbf{0} \\ \mathbf{0} & \mathbf{F}_{o,33}\mathbf{J}_3 \end{array}\right]}_{\hat{\mathbf{B}}},$$

$$\mathbf{C} \triangleq \left[\begin{array}{ccc|ccc} \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & -\mathbf{F}_{w,33}\mathbf{J}_3 & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & -\mathbf{F}_{o,33}\mathbf{J}_3 & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{array}\right], \quad \mathbf{D} \triangleq \begin{bmatrix} \mathbf{J}_2^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{F}_{w,33}\mathbf{J}_3 \\ \mathbf{0} & \mathbf{F}_{o,33}\mathbf{J}_3 \end{bmatrix}$$

(3.146, 3.147, 3.148, 3.149)

We note that the explicit representations (3.142) and (3.143) are primarily of theoretical interest because they allow direct application of concepts from systems and control theory. For computational purposes it is usually required to express the system equations in fully implicit (residual) state space form

$$\mathbf{g}\left(\mathbf{u},\mathbf{x},\dot{\mathbf{x}}\right) \equiv \hat{\mathbf{E}}\dot{\mathbf{x}} - \hat{\mathbf{A}}\mathbf{x} - \hat{\mathbf{B}}\mathbf{u} = \mathbf{0} \ , \tag{3.150}$$

where $\hat{\mathbf{E}} = \mathbf{V}$ and where $\hat{\mathbf{A}}$ and $\hat{\mathbf{B}}$ are have been defined in equations (3.146) and (3.147). The computation of the inverse of **V** as required in the explicit form can be performed

efficiently by using the analytical expression for the inverse of a 2×2 block matrix with diagonal blocks of equal size[‡]:

$$\begin{bmatrix} \mathbf{V}_{wp} & \mathbf{V}_{ws} \\ \mathbf{V}_{op} & \mathbf{V}_{os} \end{bmatrix}^{-1} = \begin{bmatrix} \tilde{\mathbf{V}}^{-1}\mathbf{V}_{os} & -\tilde{\mathbf{V}}^{-1}\mathbf{V}_{ws} \\ -\tilde{\mathbf{V}}^{-1}\mathbf{V}_{op} & \tilde{\mathbf{V}}^{-1}\mathbf{V}_{wp} \end{bmatrix}, \tag{3.151}$$

where

$$\tilde{\mathbf{V}} = \mathbf{V}_{wp}\mathbf{V}_{os} - \mathbf{V}_{ws}\mathbf{V}_{op} \ . \tag{3.152}$$

Because the four sub-matrices of $\mathbf{V}$ are diagonal, $\tilde{\mathbf{V}}$ and the four sub matrices of $\mathbf{V}^{-1}$ are also diagonal. Moreover, the inverse $\tilde{\mathbf{V}}^{-1}$ can be obtained simply by taking the reciprocals of the diagonal elements. However, we re-emphasize that there is no need to perform the inverse operation if the equations serve as a basis for computation, and that the explicit state space form (3.142) is only required for analysis of the system-theoretical properties of the equations.

Extended output vector

In the formulation discussed so far we considered system outputs in the sense of response signals, and we tacitly assumed that all system inputs were known. However, in reality, both inputs and outputs have to be measured and we can therefore also define an extended output vector that contains all measured signals. E.g. it may be required to know the oil and water flow rates in those wells were the total flow rates have been prescribed, which leads to

$$\begin{bmatrix} \tilde{\breve{\mathbf{q}}}_{well,w} \\ \tilde{\breve{\mathbf{q}}}_{well,o} \end{bmatrix} = \begin{bmatrix} \mathbf{F}_{w,22} \\ \mathbf{F}_{o,22} \end{bmatrix} \breve{\mathbf{q}}_{well,t} \ , \tag{3.153}$$

where we have added tildes to indicate that the variables are measurements rather than real prescribed variables. Moreover, we may want to include measurements of the prescribed pressure $\breve{\mathbf{p}}_{well}$ in the output. The extended output vector then becomes

$$\mathbf{y} \triangleq \begin{bmatrix} \bar{\mathbf{p}}_{well} \\ \bar{\mathbf{q}}_{well,w} \\ \bar{\mathbf{q}}_{well,o} \\ \hdashline \tilde{\breve{\mathbf{p}}}_{well} \\ \tilde{\breve{\mathbf{q}}}_{well,w} \\ \tilde{\breve{\mathbf{q}}}_{well,o} \end{bmatrix}, \tag{3.154}$$

where the elements above and below the dotted line represent measurements related to output and input variables respectively[†]. In this case the matrices $\mathbf{C}$ and $\mathbf{D}$ can be expressed as

---

[‡] The general expression for the inverse of a 2×2 block matrix is given by $\begin{bmatrix} \mathbf{V}_{11} & \mathbf{V}_{12} \\ \mathbf{V}_{21} & \mathbf{V}_{22} \end{bmatrix}^{-1} = \begin{bmatrix} \tilde{\mathbf{V}}_1^{-1} & -\mathbf{V}_{11}^{-1}\mathbf{V}_{12}\tilde{\mathbf{V}}_2^{-1} \\ -\mathbf{V}_{22}^{-1}\mathbf{V}_{21}\tilde{\mathbf{V}}_1^{-1} & \tilde{\mathbf{V}}_2^{-1} \end{bmatrix}$, where $\tilde{\mathbf{V}}_1 = \mathbf{V}_{11} - \mathbf{V}_{12}\mathbf{V}_{22}^{-1}\mathbf{V}_{21}$ and $\tilde{\mathbf{V}}_2 = \mathbf{V}_{22} - \mathbf{V}_{21}\mathbf{V}_{11}^{-1}\mathbf{V}_{12}$ are the *Schur complements* of $\mathbf{V}_{11}$ and $\mathbf{V}_{22}$ respectively; see e.g. Friedland (1986), pp. 479-481. Using the property that equally sized diagonal matrices commute, we can derive equation (3.151) from this more general expression.

[†] This distinction is not very clear cut. E.g. to compute the oil and water 'input' rates, we make use of the fractional flows around the wells which are a direct function of the saturations, i.e. of state variables. In this sense the rates also contain indirect output information on the saturations around the wells.

$$\mathbf{C} \triangleq \left[\begin{array}{ccc|ccc} \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & -\mathbf{F}_{w,33}\mathbf{J}_3 & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & -\mathbf{F}_{o,33}\mathbf{J}_3 & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{array}\right], \quad \mathbf{D} \triangleq \left[\begin{array}{cc} \mathbf{J}_2^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{F}_{w,33}\mathbf{J}_3 \\ \mathbf{0} & \mathbf{F}_{o,33}\mathbf{J}_3 \\ \hline \mathbf{0} & \mathbf{I} \\ \mathbf{F}_{w,22} & \mathbf{0} \\ \mathbf{F}_{o,22} & \mathbf{0} \end{array}\right]. \qquad (3.155, 3.156)$$

3.4.2 Well operating constraints

During the operation of an oil field it often occurs that wells or groups of wells are operated on liquid constraints because the surface facilities are not capable of processing more than a certain throughput of gas, oil and water. Water injection wells are often operated on pressure constraints to avoid or limit fracturing of the formation around the wells. Production wells are often constrained to operate at a tubing head pressure above a certain minimum, as determined by the working pressure of the first separator plus some additional pressure to displace the fluids through the flow line to that separator. Moreover, during the producing life of a field the well operating constraints may change because of changes in reservoir pressure and well bore stream composition. In practice the control of tubing head pressures or phase rates is often done indirectly, through adjusting valve settings or changing out chokes and monitoring the resulting pressure of flow rate response. Methods for well control vary drastically. At the high end we find sophisticated remotely controlled valves with remotely observed pressure gauges and multi-phase flow meters. At the low end we have manual change out of chokes, and infrequent, say monthly, observations of well head pressures and measurements of the well phase rates by temporarily re-routing the well through a test separator. In reservoir simulation we can prescribe pressures or flow rates, but, in addition we may specify constraints in the form of maximum or minimum values for pressures and flow rates in wells or groups of wells. During the simulation the conditions may change such that a well changes from being operated at a prescribed rate to being operated at a prescribed pressure or vice versa. E.g. if a production well is operated at a prescribed total liquid rate, reservoir depletion may cause the bottom hole pressure required to maintain this flow rate to gradually drop until it reaches the minimum pressure required to lift the well bore fluid to surface, i.e. until it reaches its minimum pressure constraint. From that moment on the well needs to be operated at a prescribed bottom hole pressure. Most reservoir simulators therefore allow the user to define minimum and maximum constraints for pressures and phase rates and automatically determine the *most constraining constraint* at any moment in time during the simulation. Examples of operating constraints as implemented in reservoir simulation will be discussed in Chapter 4.

3.4.3 Computational aspects

System equations (3.150) and output equations (3.145) have been implemented in a simple two-dimensional, two-phase MATLAB simulator, called `simsim`, which can simulate the flow in a horizontal, rectangular reservoir. Appendix B gives an overview of the structure of the program and a description of the main functionality. Here we discuss some general computational aspects of the numerical implementation of the two-phase system equations.

- Most (sub-)matrices considered so far are *sparse*: the accumulation sub-matrices are diagonal, the transmissibility sub-matrices are penta-diagonally banded with two sub

diagonals, and the fractional flow and well index sub-matrices are sparse diagonal. Most of the matrix elements are therefore equal to zero, and this property may be used to significantly reduce computer memory usage. In `simsim` the matrices have been programmed using MATLAB's sparse matrix option which uses an $n_0 \times 3$ matrix to store only the n_0 nonzero elements, where the first two columns contain the row and column indices of the full matrix and the third column the values of the non-zero elements.

- As mentioned before, the reordering of vector and matrix elements with permutation matrices as used in Sections 3.3.3 to 3.3.6 is not essential in a numerical implementation. There is no computational need to e.g. re-group state or input vector elements; it is merely a convenient notation. In a numerical implementation we may simply use (sparse) matrices with elements that correspond to the relevant state or input variables at the appropriate locations.
- Computation of an element of a transmissibility sub-matrix corresponding to a specific grid block involves computing the transmissibilities for flow to or from the four neighboring grid blocks. Therefore, assembly of the transmissibility (sub-)matrices requires knowledge of the *connectivities* of the grid blocks. In `simsim` this knowledge is administered with the aid of a *connectivity table*, an $n_{con} \times 2$ matrix of which each row corresponds to a connectivity between a pair of grid blocks with grid block numbers stored in the first and second column position. For a rectangular model with $n_x \times n_y$ grid blocks, we have

$$n_{con} = (n_x - 1)n_y + (n_y - 1)n_x \ , \tag{3.157}$$

and for the 2×3 reservoir used in Examples 1 and 2 the 7×2 connectivity table $\mathbf{L}_{con}$ is given by (see also Table **2-2**)

$$\mathbf{L}_{con} = \begin{bmatrix} 1 & 2 \\ 1 & 4 \\ 2 & 3 \\ 2 & 5 \\ 3 & 6 \\ 4 & 5 \\ 5 & 6 \end{bmatrix}, \tag{3.158}$$

- The elements in the two-phase state vector $\mathbf{x} = [\mathbf{p}^T \ \mathbf{s}^T]^T$ have different physical dimensions and strongly varying magnitudes. If we express the pressures in Pa, they are in the order of $10^6 - 10^7$, whereas the saturation values remain, by definition, between 0 and 1. As a result the elements of the transmissibility matrix **T**, and therefore of the system matrix **A**, will also have strongly varying magnitude. This may influence the accuracy of the result when solving a system of equations $\mathbf{A}\,\mathbf{x} = \mathbf{b}$, as will be required in Chapter 4 to simulate the response of the system[†]. The reason for the inaccuracy is in the finite precision representation of the matrix elements in any numerical implementation. We may avoid this problem by scaling the elements of **x** such that difference in magnitude between the pressure and saturation values becomes much smaller. This can be

[†] Here, **b** is an arbitrary right-hand side.

done by dividing the first n_{gb} columns of **A**, which multiply the first n_{gb} 'pressure' elements of **x**, by a factor

$$f_{scal} = \max(\mathbf{p}). \tag{3.159}$$

and multiply the corresponding elements of **x** with f_{scal} after the equations have been solved. In addition, we may also scale the elements of the right-hand side **b** by dividing the first n_{gb} rows of **b** and the corresponding rows of **A** by f_{scal}.

- In an injection well we have $\mathbf{q}_t = \mathbf{q}_w$, and we expect that soon after injection has started the fractional flows for water and oil close to the injection will approach one and zero respectively. However, before injection starts, the initial condition for the saturation is usually equal to the connate water saturation, which means that the fractional flows for water and oil are zero and one respectively. In theory, it would then be impossible to ever inject water. This paradox, which illustrates a shortcoming of the relative permeability concept, is usually circumvented by simply specifying a fractional flow equal to one for every injection well, a strategy that has also been implemented in `simsim`.

3.4.4 Lift tables*

Until now, we have considered prescribed pressures in the form of flowing bottom hole pressures p_{wf}. In most cases, however, it is not the bottom hole pressure that is controlled but the pressure at the top of the well, which is usually referred to as the flowing tubing head pressure, p_{tf}. The difference in pressure between top and bottom of a well is governed by the multi-phase flow behavior of the well bore fluids. Various techniques have been developed to compute well bore pressure drops, ranging from empirical correlations to complex mechanistic models; see e.g. Brill and Mukherjee, 1999. Typically, the tubing head pressure can be computed for given fluid properties, well bore geometry, oil, gas and water flow rates, and bottom hole pressure. Conversely, the bottom hole pressure may be computed for a given tubing head pressure. The computation is performed with the aid of a *well bore simulator* that numerically integrates a one-dimensional averaged version of the multi-phase flow equations along the well bore. Especially in the case of complex mechanistic multi-phase flow models these computations may be too time consuming to perform every time step of the reservoir simulator. An alternative approach is then to perform a large number of well bore flow simulations up-front to generate a multi-dimensional table, known as a *lift table* or *flow performance table*, which can be used as a look-up table by the reservoir simulator. Usually the four entries for a lift table are the tubing head pressure, and the oil, gas and water rates, all expressed at standard conditions[‡]. Typically each of the entries is described with a small number of points, say 5 in which case the table has $5^4 = 625$ points that correspond to the same number of bottom hole pressures. For intermediate values of the entries a linear or higher-order interpolation is used to compute the corresponding bottom hole pressure, which is much faster than performing a full well bore flow simulation. Sometimes a higher number of points is needed, at the cost of a longer pre-processing time, e.g. to prevent convergence problems during the numerical solution of the reservoir equations. No lift tables have been

[‡] Even if the reservoir is above bubble point such that it contains only oil and water and no free gas, the flow in the well bore will be three-phase because associated gas will be released from the oil as the well bore pressure decreases at increasing elevation above the reservoir. Alternatively, the lift table entries can be chosen as tubing head pressure, oil rate, gas-oil ratio, and water-cut. Whatever the choice of the table entries, it is assumed that the fluid properties at standard conditions and the well bore geometry do not change during the reservoir simulation.

implemented in `simsim`, and prescribed pressures will therefore always refer to flowing bottom hole pressures.

3.4.5 Streamlines*

As discussed in Chapter 2, the governing equations for flow through porous media consist of a mass balance equation in combination with Darcy's law which describes the relationship between spatial pressure gradients and fluid velocities. After spatial discretization, Darcy's law can be interpreted as an equation relating pressure differences between adjacent grid blocks to the Darcy velocities (volumetric fluxes) at the corresponding grid block boundaries. This discrete form of Darcy's law can be expressed as

$$\boldsymbol{v}_t = \mathbf{S}\mathbf{p} \ , \tag{3.160}$$

where $\mathbf{p}$ is an $n_{gb} \times 1$ vector of pressures at the grid block centers with n_{gb} the number of grid blocks, $\boldsymbol{v}_t$ is an $n_{con} \times 1$ vector of total Darcy velocities at the grid block boundaries with n_{con} the number of connectivities, i.e. the number of grid block boundaries, and $\mathbf{S}$ is an $n_{con} \times n_{gb}$ matrix of transmissibility coefficients. Expressions for the elements of $\mathbf{S}$ are given in detail in equation (2.145) in Section 2.4.12. Given the velocity vector $\boldsymbol{v}_t$, we can now simply visualize the trajectory of a fluid particle starting from its entrance into the reservoir at an injection well, all the way until it leaves again via a producer. These trajectories are known as *streamlines* and they can be computed using a procedure due to Pollock (1988). Consider a two-dimensional reservoir model with total Darcy velocities at the grid block boundaries given by the vector $\boldsymbol{v}_t$. The corresponding total interstitial velocities are then given by

$$\tilde{\boldsymbol{v}}_t = \frac{\boldsymbol{v}_t}{\phi} \ . \tag{3.161}$$

Assuming a linear change in velocity in the x and y directions we can define the velocity gradients g_x and g_y for a single grid block as

$$g_x = \frac{\tilde{v}_{x_0+\Delta x} - \tilde{v}_{x_0}}{\Delta x} \ , \quad g_y = \frac{\tilde{v}_{y_0+\Delta y} - \tilde{v}_{y_0}}{\Delta y} \ , \tag{3.162, 3.163}$$

where Δx and Δy are the grid block dimensions, where we dropped the subscript t from the velocities for clarity, and where we used subscripts x_0, y_0, $x_0+\Delta x$, and $y_0+\Delta y$ to indicate the four relevant elements out of the m elements of $\tilde{\boldsymbol{v}}_t$; see Figure 3.5.

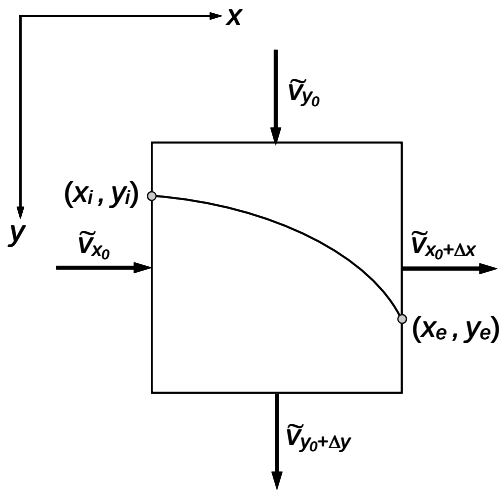


Figure 3.5: Grid block with velocity vectors at the boundaries and a streamline from entrance point (x_i, y_i) *to exit point* (x_e, y_e).

In case of positive velocity components, a fluid particle will enter the grid block either at the left or at the top. We indicate the location of the entrance point as (x_i, y_i), where it should be understood that either $x_i = x_0$ or $y_i = y_0$ (or both, in the special case that the particle enters at the corner). The particle will now travel along a curved path until it reaches the exit point (x_e, y_e), and its velocity at an arbitrary point (x, y) inside the grid block has components

$$\tilde{v}_x = \tilde{v}_{x_0} + g_x\left(x - x_0\right) \ , \ \ \tilde{v}_y = \tilde{v}_{y_0} + g_y\left(y - y_0\right) \ . \qquad (3.164, 3.165)$$

Note that at the exit point either $x_e = x_0 + \Delta x$ or $y_e = y_0 + \Delta x$, except when the particle leaves at the corner. Because we have, by definition,

$$\tilde{v}_x = \frac{dx}{dt} \ , \qquad (3.166)$$

it follows that

$$dt = \frac{1}{\tilde{v}_x} dx \ , \qquad (3.167)$$

which can be integrated to obtain the time $\Delta\tau$ to travel the distance $x_e - x_i$:

$$\int_0^{\Delta\tau} dt = \int_{x_i}^{x_e} \frac{1}{\tilde{v}_x} dx = \int_{x_i}^{x_e} \frac{1}{\tilde{v}_{x_0} + g_x\left(x - x_0\right)} dx = \frac{1}{g_x} \ln\left[\tilde{v}_{x_0} + g_x\left(x - x_0\right)\right]\Big|_{x_i}^{x_e} \ , \qquad (3.168)$$

from which we obtain

$$\Delta\tau = \frac{1}{g_x} \ln\left[\frac{\tilde{v}_{x_0} + g_x\left(x_e - x_0\right)}{\tilde{v}_{x_0} + g_x\left(x_i - x_0\right)}\right] \ . \qquad (3.169)$$

The travel time in the y direction will of course be identical and we can therefore also write

$$\Delta\tau = \frac{1}{g_y} \ln\left[\frac{\tilde{v}_{y_0} + g_y\left(y_e - y_0\right)}{\tilde{v}_{y_0} + g_y\left(y_i - y_0\right)}\right] \ . \qquad (3.170)$$

We do not know in advance whether the particle will exit at the right or at the bottom of the grid block, but we do know that it must be one of the two (or both, in case of an exit at the corner). To determine the correct exit boundary we should first compute the travel times from equations (3.169) and (3.170) using $x_e - x_0 = \Delta x$ and $y_e - y_0 = \Delta y$ respectively, i.e. for the maximum possible travel distance:

$$\Delta\tau_x = \frac{1}{g_x}\ln\left[\frac{\tilde{v}_{x_0} + g_x\Delta x}{\tilde{v}_{x_0} + g_x\left(x_i - x_0\right)}\right], \quad \Delta\tau_y = \frac{1}{g_y}\ln\left[\frac{\tilde{v}_{y_0} + g_y\Delta y}{\tilde{v}_{y_0} + g_y\left(y_i - y_0\right)}\right], \quad (3.171, 3.172)$$

and the determine the correct grid block travel time as

$$\Delta\tau = \min\left(\Delta\tau_x, \Delta\tau_y\right). \quad (3.173)$$

In the case that $\Delta\tau = \tau_y$, we have $y_e = y_0 + \Delta x$, and can we solve for x_e from equation (3.169) as

$$x_e = x_0 + \frac{1}{g_x}\left\{\left[\tilde{v}_{x_0} + g_x\left(x_i - x_0\right)\right]\exp\left(g_x\Delta\tau\right) - \tilde{v}_{x_0}\right\}. \quad (3.174)$$

Similarly, if $\Delta\tau = \tau_x$, we can solve from equation (3.170) for y_e as

$$y_e = y_0 + \frac{1}{g_y}\left\{\left[\tilde{v}_{y_0} + g_y\left(y_i - y_0\right)\right]\exp\left(g_y\Delta\tau\right) - \tilde{v}_{y_0}\right\}. \quad (3.175)$$

The exit point then forms the entry point of the next grid block and we can repeat the procedure to trace the stream line until it reaches one of the producers. If we sum the travel times over all grid blocks we obtain the *arrival time* for a streamline which indicates the moment in time at which a 'virtual' particle travelling with speed $\tilde{v}_t$ along the streamline would reach the producer (assuming it starts at the injector at time zero). A related quantity is the *time of flight* of a virtual particle to required reach a specific point along a streamline, which is equal to the summation of grid block travel times from the injector until that point. Figure 3.6 displays a streamline plot computed with `simsim` for Example 1 after steady state conditions have been reached. If, for a given total number of streamlines, we choose the fraction starting from each injector in proportion to the fraction of total water injected, the distance between streamlines becomes an inverse measure for the flow per unit surface area (which is also known as the *flux*). In other words, the closer the streamlines, the higher the flux. Apart from providing a powerful means to visualize reservoir flow, streamlines can also be used during numerical simulation, as will be briefly discussed in Chapter 4. For a much more in-depth treatment of streamline methods we refer to the classic papers of Bratvedt, Gimse and Tegnander (1996), Batycky, Blunt and Thiele (1997), King and Datta Gupta (1998) and to the text book of Datta-Gupta and King (2007).

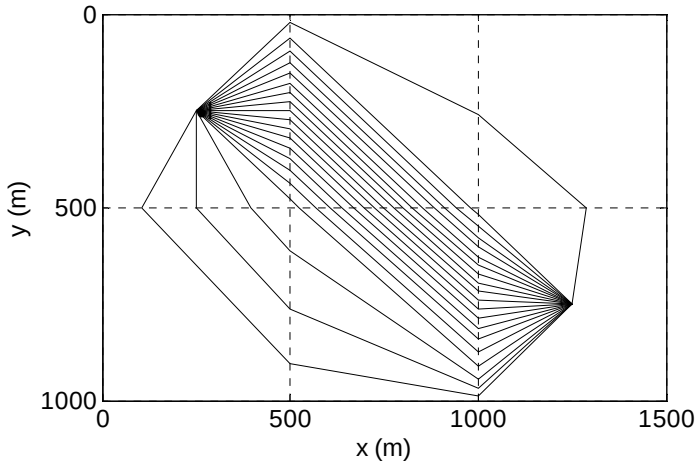


Figure 3.6: Streamlines for steady state flow in Example 1. Note: The tracing algorithm described above computes parabolic trajectories for the streamlines in each grid block. However, the streamlines have been plotted more coarsely as straight lines between the entry and exit points in the grid blocks.

3.4.6 System energy*

In analogy to the single-phase case discussed in Section 3.3.7 we can formulate the energy balance per unit time, i.e. the power balance, for the two-phase case. The power balance can be expressed in terms of potential energy rate, dissipation rate and work, each related to both oil and water flow. Using matrix-vector notation this results in equation (3.176) below, where we applied the vector and matrix partitioning as introduced in Section 3.3.3 to distinguish between gridblocks without wells, gridblocks with wells where the flow rates are prescribed, and those with wells were the bottom hole pressures are prescribed. We note that the presence of gravity forces and capillary pressures would make the expression for the power balance more complex. Using a similar reasoning as in Section 3.3.7 we can recover system equations (3.135) by first simplifying equation (3.176) such that the well index matrices $\mathbf{J}_2$ and $\mathbf{J}_3$ are eliminated, then take derivatives with repect to the state variables $\mathbf{p}_1$, $\mathbf{p}_2$ and $\mathbf{p}_3$, set the result equal to zero, and finally re-introduce the well indices for the prescribed pressures. In Section 4.4.5 we will present a numerical example that illustrates the relative importance of the various terms in the power balance.

$$\underbrace{\begin{bmatrix}\mathbf{p}_1 & \mathbf{p}_2 & \mathbf{p}_3\end{bmatrix}\left\{\begin{bmatrix}\mathbf{V}_{wp,11} & \mathbf{0} & \mathbf{0}\\ \mathbf{0} & \mathbf{V}_{wp,22} & \mathbf{0}\\ \mathbf{0} & \mathbf{0} & \mathbf{V}_{wp,33}\end{bmatrix}\begin{bmatrix}\dot{\mathbf{p}}_1\\ \dot{\mathbf{p}}_2\\ \dot{\mathbf{p}}_3\end{bmatrix}+\begin{bmatrix}\mathbf{V}_{ws,11} & \mathbf{0} & \mathbf{0}\\ \mathbf{0} & \mathbf{V}_{ws,22} & \mathbf{0}\\ \mathbf{0} & \mathbf{0} & \mathbf{V}_{ws,33}\end{bmatrix}\begin{bmatrix}\dot{\mathbf{s}}_1\\ \dot{\mathbf{s}}_2\\ \dot{\mathbf{s}}_3\end{bmatrix}\right\}}_{\left.\frac{dE_{pot}}{dt}\right|_w}$$

$$+\underbrace{\begin{bmatrix}\mathbf{p}_1 & \mathbf{p}_2 & \mathbf{p}_3\end{bmatrix}\left\{\begin{bmatrix}\mathbf{V}_{op,11} & \mathbf{0} & \mathbf{0}\\ \mathbf{0} & \mathbf{V}_{op,22} & \mathbf{0}\\ \mathbf{0} & \mathbf{0} & \mathbf{V}_{op,33}\end{bmatrix}\begin{bmatrix}\dot{\mathbf{p}}_1\\ \dot{\mathbf{p}}_2\\ \dot{\mathbf{p}}_3\end{bmatrix}+\begin{bmatrix}\mathbf{V}_{os,11} & \mathbf{0} & \mathbf{0}\\ \mathbf{0} & \mathbf{V}_{os,22} & \mathbf{0}\\ \mathbf{0} & \mathbf{0} & \mathbf{V}_{os,33}\end{bmatrix}\begin{bmatrix}\dot{\mathbf{s}}_1\\ \dot{\mathbf{s}}_2\\ \dot{\mathbf{s}}_3\end{bmatrix}\right\}}_{\left.\frac{dE_{pot}}{dt}\right|_o}$$

$$+\underbrace{\begin{bmatrix}\left(\mathbf{p}_1-\mathbf{p}_{av}\right) & \left(\mathbf{p}_2-\mathbf{p}_{av}\right) & \left(\mathbf{p}_3-\mathbf{p}_{av}\right)\end{bmatrix}\begin{bmatrix}\mathbf{T}_{w,11} & \mathbf{T}_{w,12} & \mathbf{T}_{w,13}\\ \mathbf{T}_{w,21} & \mathbf{T}_{w,22} & \mathbf{T}_{w,23}\\ \mathbf{T}_{w,31} & \mathbf{T}_{w,32} & \mathbf{T}_{w,33}\end{bmatrix}\begin{bmatrix}\mathbf{p}_1\\ \mathbf{p}_2\\ \mathbf{p}_3\end{bmatrix}}_{\left.\frac{dE_{dis,gb}}{dt}\right|_w}$$

$$+\underbrace{\begin{bmatrix}\left(\mathbf{p}_1-\mathbf{p}_{av}\right) & \left(\mathbf{p}_2-\mathbf{p}_{av}\right) & \left(\mathbf{p}_3-\mathbf{p}_{av}\right)\end{bmatrix}\begin{bmatrix}\mathbf{T}_{o,11} & \mathbf{T}_{o,12} & \mathbf{T}_{o,13}\\ \mathbf{T}_{o,21} & \mathbf{T}_{o,22} & \mathbf{T}_{o,23}\\ \mathbf{T}_{o,31} & \mathbf{T}_{o,32} & \mathbf{T}_{o,33}\end{bmatrix}\begin{bmatrix}\mathbf{p}_1\\ \mathbf{p}_2\\ \mathbf{p}_3\end{bmatrix}}_{\left.\frac{dE_{dis,gb}}{dt}\right|_o}$$

$$+\underbrace{\begin{bmatrix}\mathbf{0} & \left(\overline{\mathbf{p}}_{well}-\mathbf{p}_2\right) & \left(\breve{\mathbf{p}}_{well}-\mathbf{p}_3\right)\end{bmatrix}\begin{bmatrix}\mathbf{0} & \mathbf{0} & \mathbf{0}\\ \mathbf{0} & \mathbf{F}_{w,22} & \mathbf{0}\\ \mathbf{0} & \mathbf{0} & \mathbf{F}_{w,33}\end{bmatrix}\begin{bmatrix}\mathbf{0} & \mathbf{0} & \mathbf{0}\\ \mathbf{0} & \mathbf{J}_2 & \mathbf{0}\\ \mathbf{0} & \mathbf{0} & \mathbf{J}_3\end{bmatrix}\begin{bmatrix}\mathbf{0}\\ \overline{\mathbf{p}}_{well}-\mathbf{p}_2\\ \breve{\mathbf{p}}_{well}-\mathbf{p}_3\end{bmatrix}}_{\left.\frac{dE_{dis,well}}{dt}\right|_w}$$

$$+\underbrace{\begin{bmatrix}\mathbf{0} & \left(\overline{\mathbf{p}}_{well}-\mathbf{p}_2\right) & \left(\breve{\mathbf{p}}_{well}-\mathbf{p}_3\right)\end{bmatrix}\begin{bmatrix}\mathbf{0} & \mathbf{0} & \mathbf{0}\\ \mathbf{0} & \mathbf{F}_{o,22} & \mathbf{0}\\ \mathbf{0} & \mathbf{0} & \mathbf{F}_{o,33}\end{bmatrix}\begin{bmatrix}\mathbf{0} & \mathbf{0} & \mathbf{0}\\ \mathbf{0} & \mathbf{J}_2 & \mathbf{0}\\ \mathbf{0} & \mathbf{0} & \mathbf{J}_3\end{bmatrix}\begin{bmatrix}\mathbf{0}\\ \overline{\mathbf{p}}_{well}-\mathbf{p}_2\\ \breve{\mathbf{p}}_{well}-\mathbf{p}_3\end{bmatrix}}_{\left.\frac{dE_{dis,well}}{dt}\right|_o}$$

$$=\underbrace{\begin{bmatrix}\mathbf{0} & \overline{\mathbf{p}}_{well} & \overline{\mathbf{q}}_{well}\end{bmatrix}\begin{bmatrix}\mathbf{0} & \mathbf{0} & \mathbf{0}\\ \mathbf{0} & \mathbf{F}_{w,22} & \mathbf{0}\\ \mathbf{0} & \mathbf{0} & \mathbf{F}_{w,33}\end{bmatrix}\begin{bmatrix}\mathbf{0}\\ \breve{\mathbf{q}}_{well}\\ \breve{\mathbf{p}}_{well}\end{bmatrix}}_{P_{well}|_w}+\underbrace{\begin{bmatrix}\mathbf{0} & \overline{\mathbf{p}}_{well} & \overline{\mathbf{q}}_{well}\end{bmatrix}\begin{bmatrix}\mathbf{0} & \mathbf{0} & \mathbf{0}\\ \mathbf{0} & \mathbf{F}_{o,22} & \mathbf{0}\\ \mathbf{0} & \mathbf{0} & \mathbf{F}_{o,33}\end{bmatrix}\begin{bmatrix}\mathbf{0}\\ \breve{\mathbf{q}}_{well}\\ \breve{\mathbf{p}}_{well}\end{bmatrix}}_{P_{well}|_o}. \tag{3.176}$$

3.5 References for Chapter 3

Batycky, R.P., Blunt, M.J. and Thiele, M.R. 1997: A 3D field-scale streamline-based reservoir simulator. *SPE Reservoir Engineering* **12** (4) 246-254. DOI: 10.2118/36726-PA.

Bratvedt, F., Gimse, T. and Tegnander, C. (1996), Streamline computations for porous media flow including gravity. *Transport in Porous Media* **25** (1) 63-78. DOI: 10.1007/BF00141262.

Brill, J.P. and Mukherjee, H., 1999: Multi-phase flow in wells. *SPE Monograph Series* **17**, SPE, Richardson.

Datta-Gupta, A. and King, M.J., 2007: *Streamline simulation: Theory and Practice*, SPE Textbook Series, **11**, SPE, Richardson.

Friedland, B., 1986: *Control system design – An introduction to state-space methods*, McGraw-Hill. Reprinted in 2005 by Dover, New York.

Hubbert, M.K., 1940: The theory of ground-water motion. *Journal of Geology* **48** (8) 785-944.

Karney, B.W. and Seneviratne, A., 1991: Application of energy concepts to groundwater flow: time step control and integrated sensitivity analysis. *Water Resources Research* **27** (12) 3225-3235.

King, M.J. and Datta-Gupta, A., 1998: Streamline simulation: a current perspective. *In Situ* **22** (1), 91-140.

Lanczos, C., 1970: *The variational principles of mechanics 4th ed.*, University of Toronto Press, Toronto. Reprinted in 1986 by Dover, New York.

Langhaar, H.L., 1962: *Energy methods in applied mechanics*, Wiley, New York.

Pollock, D.W., 1988: Semi analytical computation of path lines for finite-difference models. *Ground Water* **26** (6), 743-750.

Zienckiewicz, O.C. and Taylor, R.L., 1989: *The Finite Element Method, 4th ed., Vol. 1 & 2*, McGraw-Hill, London.

4 System response

4.1 Free response

4.1.1 Homogeneous equation

Consider the linear or linearized time-invariant state space equations given in equation (3.8) or (3.24). The scalar equivalent of these vector differential equations is given by

$$\dot{x}(t) = ax(t) + bu(t), \tag{4.1}$$

where a and b are now time-invariant scalar coefficients[†]. Because equation (4.1) is first-order in the dependent variable t, it requires a single initial condition:

$$t = \breve{t} : x = \breve{x}. \tag{4.2}$$

If we set $u = 0$ in equation (4.1), we obtain the *homogeneous equation*

$$\dot{x}(t) = ax(t), \tag{4.3}$$

which describes the *free* response (also called the *transient response*) of the system as a response to a non-zero initial condition

$$x(t) = e^{a(t-\breve{t})}\breve{x}. \tag{4.4}$$

For values of the coefficient a smaller than zero, the response $x(t)$ for the limit of t approaching infinity becomes zero, i.e. the response is truly transient. For values of a larger than zero, the response grows to infinity, while for $a = 0$, the response remains equal to the initial condition $\breve{x}$. Just as for scalar ODEs, if we set $\mathbf{u} = \mathbf{0}$, in equation (4.1) we obtain the homogeneous equation

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t), \tag{4.5}$$

with a corresponding initial condition

$$t = \breve{t} : \mathbf{x} = \breve{\mathbf{x}}. \tag{4.6}$$

4.1.2 Diagonalization

A solution to equation (4.5) can be obtained through *diagonalization* of matrix **A**. If **A** is diagonalizable there exists a non-singular matrix **M** of eigenvectors **m** of **A** such that

$$\mathbf{A} = \mathbf{M}\boldsymbol{\Lambda}\mathbf{M}^{-1}. \tag{4.7}$$

where $\boldsymbol{\Lambda} = \text{diag}(\lambda_1, \ldots, \lambda_n)$ is the diagonal matrix of eigenvalues of **A**. Using this decomposition of **A**, equation (4.5) can be written as

$$\dot{\mathbf{x}}(t) = \mathbf{M}\boldsymbol{\Lambda}\mathbf{M}^{-1}\mathbf{x}(t), \tag{4.8}$$

or equivalently, after pre-multiplying (4.8) by the $\mathbf{M}^{-1}$

$$\mathbf{M}^{-1}\dot{\mathbf{x}}(t) = \boldsymbol{\Lambda}\mathbf{M}^{-1}\mathbf{x}(t). \tag{4.9}$$

Defining a transformed state variable **z** as

[†] For an engineering-oriented overview of the theory of first-order scalar and vector ODEs see e.g. the review in Luenberger (1979) or, somewhat more extensively, Boyce and Di Prima (2005). An enormous amount of other textbooks is available covering similar material.

$$\mathbf{z}(t) = \mathbf{M}^{-1}\mathbf{x}(t) \ , \tag{4.10}$$

and substituting it into (4.9) yields the *decoupled* system of homogeneous equations

$$\dot{\mathbf{z}}(t) = \mathbf{\Lambda}\mathbf{z}(t) . \tag{4.11}$$

It is called decoupled since each of the elements z_i of the transformed state $\mathbf{z}$ is given by

$$\dot{z}_i(t) = \lambda_i z_i(t) \ , \tag{4.12}$$

thus without being influenced by any of the other elements of $\mathbf{z}$. The solution of (4.12) is given by (c.f. equation (4.4))

$$z_i(t) = e^{\lambda_i(t-\breve{t})}\breve{z}_i \ , \tag{4.13}$$

and, equivalently, the solution of the 'full' transformed state variable $\mathbf{z}$ can be written as

$$\mathbf{z}(t) = e^{\mathbf{\Lambda}(t-\breve{t})}\breve{\mathbf{z}} . \tag{4.14}$$

An interpretation of the matrix exponential $e^{\mathbf{\Lambda}}$ can be obtained by considering the Taylor expansion around zero for the exponential function,

$$e^{\lambda} = 1 + \lambda + \frac{\lambda^2}{2!} + \frac{\lambda^3}{3!} + \dots \ , \tag{4.15}$$

of which the matrix equivalent can be written as[†]

$$e^{\mathbf{\Lambda}} = \mathbf{I} + \mathbf{\Lambda} + \frac{\mathbf{\Lambda}^2}{2!} + \frac{\mathbf{\Lambda}^3}{3!} + \dots \ . \tag{4.16}$$

To recover the solution in terms of the original state variable $\mathbf{x}$, first substitute relationship (4.10) in equation (4.14) and multiply with $\mathbf{M}$ to obtain

$$\mathbf{x}(t) = \mathbf{M}e^{\mathbf{\Lambda}(t-\breve{t})}\mathbf{M}^{-1}\breve{\mathbf{x}} . \tag{4.17}$$

Using equations (4.7) and (4.16) we can then write

$$\begin{aligned}
\mathbf{M}e^{\mathbf{\Lambda}(t-\breve{t})}\mathbf{M}^{-1} &= \mathbf{M}\left(\mathbf{I} + \mathbf{\Lambda}(t-\breve{t}) + \frac{\left[\mathbf{\Lambda}(t-\breve{t})\right]^2}{2!} + \frac{\left[\mathbf{\Lambda}(t-\breve{t})\right]^3}{3!} + \dots\right)\mathbf{M}^{-1} \\
&= \mathbf{M}\left(\mathbf{I} + \mathbf{M}^{-1}\mathbf{A}\mathbf{M}(t-\breve{t}) + \frac{\left[\mathbf{M}^{-1}\mathbf{A}\mathbf{M}(t-\breve{t})\right]^2}{2!} + \frac{\left[\mathbf{M}^{-1}\mathbf{A}\mathbf{M}(t-\breve{t})\right]^3}{3!} + \dots\right)\mathbf{M}^{-1} \\
&= \mathbf{I} + \mathbf{A}(t-\breve{t}) + \frac{\left[\mathbf{A}(t-\breve{t})\right]^2}{2!} + \frac{\left[\mathbf{A}(t-\breve{t})\right]^3}{3!} + \dots \\
&= e^{\mathbf{A}(t-\breve{t})},
\end{aligned} \tag{4.18}$$

[†] In practice, the computation of a matrix exponential should not be performed using this expression; see Moler and Van Loan (1978) for an overview of various possible methods. The MATLAB command `expm(A)` produces the matrix exponential using a technique known as Padé approximation.

and substitution of this relationship in equation (4.17) finally gives the solution of homogeneous equation (4.5) as

$$\mathbf{x}(t) = e^{\mathbf{A}(t-\bar{t})}\breve{\mathbf{x}}\,. \tag{4.19}$$

Equation (4.19) is the matrix equivalent to scalar equation (4.4) and represents the transient behavior of the LTI system with system matrix **A**. The diagonalization of **A** in equation (4.7) is an example of a *similarity transformation* because the dynamic system characterized by the transformed system matrix $\mathbf{\Lambda}$ has the same dynamic properties as the one represented by the original system matrix **A**, since both matrices have the same eigenvalues.

4.1.3 Stability

If all eigenvalues λ_i of matrix **A** are smaller than zero, it follows that the response **x**(*t*) of homogeneous equation (4.5) for the limit of *t* approaching infinity becomes zero, i.e. the response is truly transient[‡]. This property of a dynamic system is known as *asymptotic stability*. If any of the eigenvalues is larger than zero, the response grows to infinity (that is, in the linear theory), i.e. the system is *unstable*. If at least one of the eigenvalues is equal to zero, whereas the others are smaller than zero, the response for large values of *t* may approach a non-zero steady-state value, a situation known as *marginal stability*. Physical instability requires an internal source of energy in the system. In the case of flow through porous media such a source normally does not exist, and to the contrary, the system is continuously loosing energy through friction of the fluid in the pores as described by Darcy's law. Unlike in classic control engineering, *physical* instability in time is therefore normally not an issue[†]. An exception is the behavior of coupled well bore-reservoir systems where occasionally instable behavior of the well bore flow may lead to large pressure and flow oscillations in the well bore and the near-well bore area. In that case, the source of the instability is in the multiphase flow behavior in the well bore, and not in the reservoir. Analysis of this type of coupled problems requires a dynamic well bore simulator that is capable of computing the well bore dynamics at a time scale of seconds to hours. We will not consider such short-term phenomena, and will restrict our attention to reservoir flow at time scales from days to decades where well bore flow instabilities play no role. A completely different, artificial, source of instability is related to the numerical simulation of reservoir flow. As will be illustrated in the next section, incorrect time discretization of the system equations may lead to *numerical* instabilities which may completely ruin the simulation, or worse, produce output that at first sight looks in order but contains fluctuations that are completely unphysical.

[‡] More precisely, the condition for asymptotic stability requires that all *real parts of the eigenvalues* are smaller than zero, a condition that is also referred to as the system matrix being *Hurwitz*. This condition is only of relevance if the eigenvalues are complex numbers, which implies that the system displays oscillatory behavior. However, because we don't take inertia into account in the description of porous media flow, the system can not store kinetic energy and the pressures will only display exponentially decaying behavior. Correspondingly, the eigenvalues are real numbers and it suffices to require them to be negative-valued to guarantee asymptotic stability.

[†] However, instabilities in *space* do play a role in reservoir engineering, at least in theory. A well known case is reservoir flooding with an unfavorable *mobility ratio*, i.e. with the mobility of the displacing fluid being lower than the mobility of the displaced fluid. In that case, a displacement front may become unstable such that *viscous fingering* takes place of the displacing fluid in the displaced fluid. Similar instabilities may occur when a heavy fluid is injected on top of a lighter one, which may lead to fingering caused by *buoyancy-driven convection*. In practice, geological heterogeneities often completely mask the fingering process.

4.1.4 Singular system matrix

In Section (3.3.3) it was discussed that the elements of an input vector **u** may consist of prescribed flow rates or prescribed bottom hole pressures (or a relation between flow rates and pressures). In section (3.3.4) it was shown that the use of a well model to prescribe the bottom hole pressures results in the addition of a term $\mathbf{J}_p$ to the main block diagonal of the transmissibility matrix **T**, and here we will have a look at an important consequence of that addition. The transmissibility matrix **T** of a reservoir model with only prescribed flow rates, and therefore no prescribed bottom hole pressures, is singular. This can be understood by considering that **T** defines the transmissibilities between the grid blocks, which directly correspond to the steady-state pressure differences between the grid blocks. However, knowing only the pressure *differences* does not give us enough information to compute the absolute pressures in the grid blocks. This implies that the transmissibility matrix is singular with rank deficiency one. Another way to understand this is by considering the structure of **T**: the sum of every row adds up to exactly zero because of the way the transmissibilities enter the matrix; see equation (2.31). Therefore the sum of all columns has to be equal to the zero vector which implies that nontrivial solutions of the homogeneous equation

$$\mathbf{T}\mathbf{p} = \mathbf{0}, \tag{4.20}$$

are given by

$$\mathbf{p} = p\mathbf{1}, \tag{4.21}$$

where p is an arbitrary constant pressure. In other words, the null space of **T** consists of all vectors **p** with arbitrary, equal values p in each grid block. To restore regularity of **T** we need to fix at least one of the pressures. Because the well index matrix $\mathbf{J}_3$ is diagonal, addition of $\mathbf{J}_3$ to the main block diagonal of **T** results in the addition of non-zero elements to the main diagonal of **T** which indeed restores the regularity. It will be shown in Section 4.2 below that singularity of **T**, and thus of **A**, makes it impossible to directly compute the long-term steady-state pressure distribution in the system, or the behavior in the limit of incompressible flow. It will be shown below that it may still be possible to compute the pressures dynamically through numerical integration of the system equations in time, as long as it concerns compressible flow. However, in that case the singularity of **A** could still lead to numerical problems, in particular for long integration times and small compressibilities.

4.1.5 Example 1 continued – Free response

Using MATLAB the eigenvalues and eigenvectors of the system matrix **A** can be computed as

$$\mathbf{\Lambda} = 10^{-5} \times \begin{bmatrix} -0.4163 & 0 & 0 & 0 & 0 & 0 \\ 0 & -0.1723 & 0 & 0 & 0 & 0 \\ 0 & 0 & -0.0660 & 0 & 0 & 0 \\ 0 & 0 & 0 & -0.0285 & 0 & 0 \\ 0 & 0 & 0 & 0 & -0.0059 & 0 \\ 0 & 0 & 0 & 0 & 0 & -0.0000 \end{bmatrix}. \tag{4.22}$$

$$\mathbf{M} = \begin{bmatrix} 0.4108 & 0.6716 & 0.3437 & 0.2351 & 0.2001 & -0.4082 \\ -0.7867 & -0.0760 & 0.3832 & 0.1631 & 0.1875 & -0.4082 \\ 0.0052 & -0.0006 & -0.0107 & 0.0737 & -0.9098 & -0.4082 \\ -0.0571 & 0.0108 & -0.8093 & 0.3551 & 0.2211 & -0.4082 \\ 0.4562 & -0.7269 & 0.2409 & 0.0581 & 0.1881 & -0.4082 \\ -0.0284 & 0.1210 & -0.1479 & -0.8850 & 0.1130 & -0.4082 \end{bmatrix}, \tag{4.23}$$

Note that one of the eigenvalues (the sixth) is zero, in line with the rank-1 deficiency of **A**. Figure 4.1 displays the eigenvectors by plotting their elements on the corresponding grid blocks. It can be seen that the eigenvector belonging to the zero-eigenvalue has an (arbitrary) constant value[‡]. The other five eigen vectors form basis functions in the form of spatial patterns of grid block pressures.

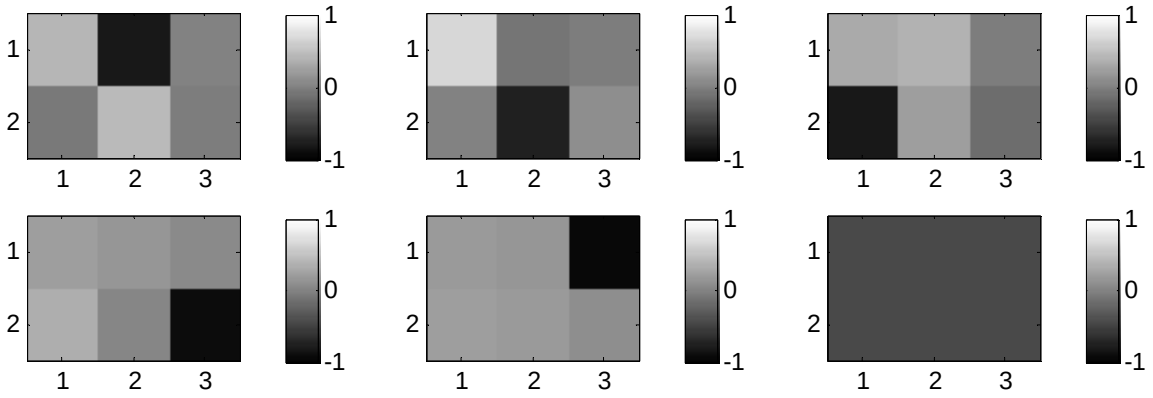


Figure 4.1: Eigenvectors of the system matrix **A** *of Example 1.*

Next, we perform an analytical integration of the state equations of Example 1 according to equation (4.17). Starting from an unbalanced initial condition[†]

$$\mathbf{x}(0) = \breve{p}_R \begin{bmatrix} 0.700 \\ 0.900 \\ 1.100 \\ 0.900 \\ 1.100 \\ 1.300 \end{bmatrix} = \begin{bmatrix} 21.000 \\ 27.000 \\ 33.000 \\ 27.000 \\ 33.000 \\ 39.000 \end{bmatrix} \times 10^6 \,\mathrm{Pa}\,. \tag{4.24}$$

we can compute the response for various values of time *t*. The initial condition in terms of the coefficient vector **z** can be computed as (see equation (4.10))

$$\mathbf{M}\mathbf{z}(0) = \mathbf{x}(0)\ , \tag{4.25}$$

which leads to

[‡] Recall that all eigenvectors are defined up to an arbitrary constant; see Section A.4 in Appendix A.

[†] I.e. an initial condition where not all grid block pressures have identical values. The initial condition in equation (4.24) consists of spatial fluctuations around the initial reservoir pressure $\breve{p}_R = 30 \times 10^6$ specified in Table 2.1.

$$\mathbf{z}(0) = \begin{bmatrix} -0.037 \\ -6.943 \\ -2.455 \\ -11.240 \\ -4.175 \\ -73.485 \end{bmatrix} \times 10^6 \,. \tag{4.26}$$

Note that the coefficients **z** are dimensionless which implies that the elements of the eigenvectors **m** must have a dimension of pressure (Pa). Figure 4.2 displays the values of the 6 coefficients (i.e. the 6 elements of **z**) on a logarithmic scale in time, and after one year they have the values

$$\mathbf{z}\underbrace{(365\times 24\times 3600)}_{\text{one year in seconds}} = \begin{bmatrix} -0.000 \\ -0.000 \\ -0.000 \\ -0.000 \\ -0.651 \\ -73.485 \end{bmatrix} \times 10^6 \,. \tag{4.27}$$

Clearly the importance of the eigenvectors corresponding to the eigenvalues with the largest absolute values reduces the fastest. The physical interpretation is that those 'modes' are the most heavily damped, and the straight lines in the semi-logarithmic plot illustrate the rapid, exponential nature of the decay. Only the value of the sixth coefficient, multiplying the pattern corresponding to the zero-eigenvalue, does not change its value at all. The product of this coefficient with its corresponding pattern represents the average pressure in all grid blocks after the effect of the initial conditions has been dampened out completely[†]:

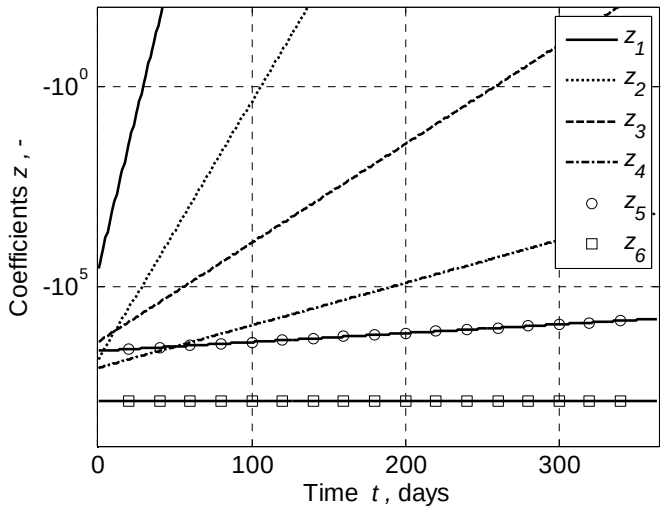


Figure 4.2: Coefficients z_1 to z_6 as a function of time on a logarithmic scale.

[†] Here we use the notation $\mathbf{x}(\infty)$ to indicate $\lim_{t\to\infty} \mathbf{x}(t)$.

$$\mathbf{x}(\infty) = \mathbf{m}_6 z_6(\infty) = \begin{bmatrix} -0.4082 \\ -0.4082 \\ -0.4082 \\ -0.4082 \\ -0.4082 \\ -0.4082 \end{bmatrix} \times -73.485 \times 10^6 = \begin{bmatrix} 30.00 \\ 30.00 \\ 30.00 \\ 30.00 \\ 30.00 \\ 30.00 \end{bmatrix} \times 10^6 \text{ Pa.} \tag{4.28}$$

The pressure vector **x** for the entire period can be recovered as

$$\mathbf{x}(t) = \mathbf{M}\mathbf{z}(t) \ , \tag{4.29}$$

and Figure 4.3 displays the results for the pressures in grid blocks 1, 2, 5 and 6 for a period of one year, which in the end all reach the average value of 30×10^6 Pa .

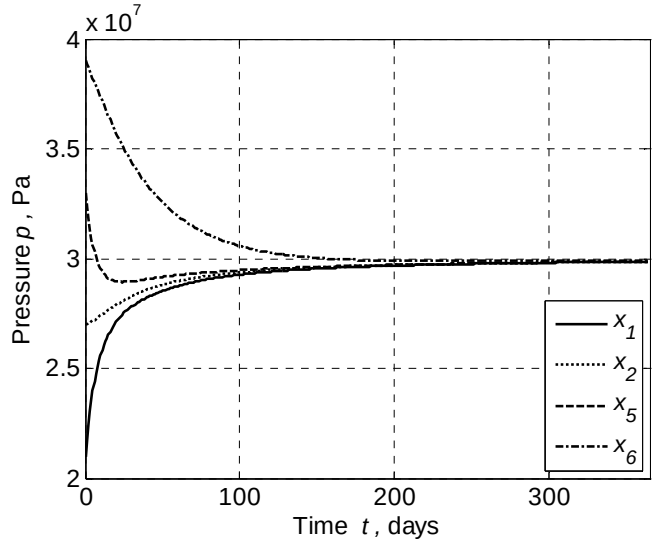


Figure 4.3: Pressures in grid blocks 1, 2, 5 and 6 as a function of time obtained by analytical integration of the state equations for Example 1.

4.2 Forced response

4.2.1 Nonhomogeneous equation

We return to the scalar LTI nonhomogeneous equation (4.1). Following the standard theory of linear differential equations, we can write the *general solution* of equation (4.1) as the sum of the free response, given by equation (4.4), and the *forced response* which depends on the *input term* (also known as *forcing term*) $bu(t)$ and which is often referred to as the *particular solution*

$$x(t) = \int_{\breve{t}}^{t} e^{a(t-\tau)} bu(\tau)\, d\tau \ . \tag{4.30}$$

We can interpret the integral in expression (4.30) as the limit for $\Delta\tau \to 0$ of a summation of transient responses to inputs $u(\tau)$, multiplied with b, over small time intervals $\Delta\tau$ during the period $\breve{t} \le \tau \le t$. Although mathematically there is no problem in considering cases where time runs backwards, we usually restrict the analysis to cases where the underlying physics forces *causality*, which implies that the states and the outputs are only influenced by past

inputs[†]. The general solution of equation (4.1) is now obtained as the sum of the homogeneous solution and the particular solution; i.e. the general response is the sum of the transient response and the forced response:

$$x(t) = e^{a(t-\bar{t})}\breve{x} + \int_{\bar{t}}^{t} e^{a(t-\tau)} bu(\tau)\, d\tau \,. \tag{4.31}$$

In analogy to these scalar results, the forced response of the nonhomogeneous LTI vector differential equation

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t), \tag{4.32}$$

is given by

$$\mathbf{x}(t) = \int_{\bar{t}}^{t} e^{\mathbf{\Lambda}(t-\tau)} \mathbf{B}\mathbf{u}(\tau)\, d\tau \,, \tag{4.33}$$

such that the general solution is obtained as the sum of solutions (4.19) and (4.33):

$$\mathbf{x}(t) = e^{\mathbf{\Lambda}(t-\bar{t})}\breve{\mathbf{x}} + \int_{\bar{t}}^{t} e^{\mathbf{\Lambda}(t-\tau)} \mathbf{B}\mathbf{u}(\tau)\, d\tau \,. \tag{4.34}$$

4.2.2 Diagonalization and modal analysis

Just like in the homogeneous case, the inhomogeneous equations may be decoupled through diagonalization of the system equations. Substitution of equation (4.7) in equation (4.32) results in

$$\dot{\mathbf{x}}(t) = \mathbf{M}\mathbf{\Lambda}\mathbf{M}^{-1}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t), \tag{4.35}$$

which, after pre-multiplication with $\mathbf{M}^{-1}$ can be written as

$$\dot{\mathbf{z}}(t) = \mathbf{\Lambda}\mathbf{z}(t) + \mathbf{M}^{-1}\mathbf{B}\mathbf{u}(t), \tag{4.36}$$

where $\mathbf{z}$ is a transformed state variable as defined before in equation (4.10). The eigenvectors $\mathbf{m}$, i.e. the columns of $\mathbf{M}$, are also known as the *modes* of the dynamic system, and equation (4.36) is therefore referred to as a *modal representation* of the system equations. The general solution, equation (4.34), can be rewritten in modal form as

$$\mathbf{z}(t) = e^{\mathbf{\Lambda}(t-\bar{t})}\breve{\mathbf{z}} + \int_{\bar{t}}^{t} e^{\mathbf{\Lambda}(t-\tau)} \mathbf{M}^{-1}\mathbf{B}\mathbf{u}(\tau)\, d\tau \,. \tag{4.37}$$

In case of physical systems where inertia plays a role, such as e.g. mechanical (mass-spring) systems or electrical (inductance-capacitance) networks, the modes correspond to spatial patterns of oscillations for the undamped homogeneous system. In particular for mechanical systems there exists an extensive branch of *modal analysis* techniques to obtain the modes (eigenvectors) and the associated frequencies (eigenvalues) of a system from experiments. In the case of flow through porous media, inertia is usually neglected, which means that the free response of the system is non-oscillatory and just consists of decaying exponential functions (see equation (4.19)), such that the modes have much less physical significance. Note that

[†] A system where the states are influenced by future inputs is therefore referred to as *non-causal*.

although the homogeneous equations were fully decoupled (see equations (4.11) and (4.12)), the inhomogeneous equations are in general coupled through the input, because except for the special case that $\mathbf{M}^{-1}\mathbf{B}$ is a unit matrix, the elements of the input vector **u** will influence more than just a single mode.

4.2.3 Singular system matrix

Steady-state response

Consider the LTI inhomogeneous equation (4.32) with a regular system matrix **A**. In the special case that the input vector $\mathbf{u}(t)$ becomes a constant $\mathbf{u}(\infty)$ for $t \to \infty$, the steady-state solution can be obtained by putting $\dot{\mathbf{x}} = \mathbf{0}$, resulting in the linear system of equations

$$\mathbf{A}\mathbf{x}(\infty) = -\mathbf{B}\mathbf{u}(\infty), \tag{4.38}$$

which could then be solved for $\mathbf{x}(\infty)$. Formally this can be written as

$$\mathbf{x}(\infty) = -\mathbf{A}^{-1}\mathbf{B}\mathbf{u}(\infty), \tag{4.39}$$

although in practice it is computationally more efficient to solve the system of equations (4.38). Note, however, that for a singular matrix **A** we cannot solve equation (4.38). As discussed in Section 4.1.4 above, **T**, and therefore **A**, are singular if we prescribe only the flow rates in the wells. If we fix at least one of the pressures with the aid of a well model the resulting modified transmissibility matrix $(\mathbf{T}^* + \mathbf{J}_p^*)$ is regular, and therefore also the system matrix $\mathbf{A}^*$, and we can compute the steady-state vectors $\mathbf{x}^*(\infty)$ and $\mathbf{y}^*(\infty)$ from

$$\mathbf{A}^*\mathbf{x}^*(\infty) = -\mathbf{B}^*\mathbf{u}^*(\infty), \quad \mathbf{y}^*(\infty) = \mathbf{C}^*\mathbf{x}^*(\infty) + \mathbf{D}^*\mathbf{u}^*(\infty). \tag{4.40, 4.41}$$

Incompressible flow

A similar situation occurs in the limit of incompressible flow. As discussed in Section 0, in that case the accumulation matrix **V** vanishes, such that differential equation (3.8) is replaced by an algebraic equation

$$\mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) = \mathbf{0}, \tag{4.42}$$

with

$$\mathbf{A} = -\mathbf{T}, \quad \mathbf{B} = \mathbf{L}_{qu}. \tag{4.43, 4.44}$$

The solution can be obtained by solving the system of equations

$$\mathbf{A}\mathbf{x}(t) = -\mathbf{B}\mathbf{u}(t), \tag{4.45}$$

where the dynamic response $\mathbf{x}(t)$ is now assumed to occur instantaneously. Just as in the steady-state case, **A** needs to be regular, i.e. we need to fix at least one of the pressures in the wells.

4.3 Numerical simulation

Until now we have considered the response of simple linear reservoir systems for which it was possible to obtain analytical solutions. For more realistic, nonlinear, reservoir systems we need to simulate the response numerically.

4.3.1 Explicit Euler discretization

To numerically simulate the dynamic system behavior it is required to discretize not only the spatial variables but also the time variable. In other words, we need to discretize the *continuous-time* system of ODEs and derive a *discrete-time* system of ODEs. Starting from the general form of the system equations (3.6), the most simple approach is to discretize the equation by replacing the difference $d\mathbf{x}/dt$ by differential $\Delta\mathbf{x}/\Delta t$:

$$\frac{\Delta \mathbf{x}}{\Delta t} \approx \mathbf{f}\left(\mathbf{u}(t), \mathbf{x}(t)\right) . \tag{4.46}$$

This gives us an algorithm to compute an approximate new value $\mathbf{x}_k$ at t_k from a known value $\mathbf{x}_{k\text{-}1}$ at $t_{k\text{-}1}$:[†]

$$\mathbf{x}_k = \mathbf{x}_{k-1} + \Delta\mathbf{x} \approx \mathbf{x}_{k-1} + \mathbf{f}\left(\mathbf{u}_{k-1}, \mathbf{x}_{k-1}\right)\Delta t \ . \tag{4.47}$$

where $\Delta t = t_k - t_{k\text{-}1}$. The counter k is generally referred to as the discrete time. More formally, the same result is obtained by using a forward Taylor expansion for $\mathbf{x}$ at $t_{k\text{-}1}$:

$$\mathbf{x}_k = \mathbf{x}_{k-1} + \left[\frac{d\mathbf{x}(t)}{dt}\right]_{t=t_{k-1}} \Delta t + \frac{1}{2}\left[\frac{d^2\mathbf{x}(t)}{dt^2}\right]_{t=t_{k-1}} (\Delta t)^2 + \ldots \ . \tag{4.48}$$

Substitution of equation (3.5) into equation (4.48) and disregarding all terms higher than first-order leads indeed to equation (4.47). As an illustration we apply the simple approach to the LTI state and output equations (3.8) and (3.11),

$$\dot{\mathbf{x}}(t) = \mathbf{A}_c\mathbf{x}(t) + \mathbf{B}_c\mathbf{u}(t), \quad \mathbf{y}(t) = \mathbf{C}_c\mathbf{x}(t) + \mathbf{D}_c\mathbf{u}(t) \ , \tag{4.49, 4.50}$$

where we have now added subscripts c to matrices $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$ and $\mathbf{D}$ to indicate that they are related to a continuous-time representation. Following equation (4.47) we obtain

$$\mathbf{x}_k = \mathbf{x}_{k-1} + \left(\mathbf{A}_c\mathbf{x}_{k-1} + \mathbf{B}_c\mathbf{u}_{k-1}\right)\Delta t \ . \tag{4.51}$$

Defining

$$\mathbf{A}_d = \left(\mathbf{I} + \Delta t\mathbf{A}_c\right), \ \mathbf{B}_d = \Delta t\mathbf{B}_c, \ \mathbf{C}_d = \mathbf{C}_c \text{ and } \mathbf{D}_d = \mathbf{D}_c, \tag{4.52, 4.53, 4.54, 4.55}$$

allows us to write the general state space system in discrete-time form:

$$\mathbf{x}_k = \mathbf{A}_d\mathbf{x}_{k-1} + \mathbf{B}_d\mathbf{u}_{k-1}, \quad \mathbf{y}_k = \mathbf{C}_d\mathbf{x}_k + \mathbf{D}_d\mathbf{u}_k \ . \tag{4.56, 4.57}$$

Equation (4.56) is a *difference equation.* It represents one particular discrete-time equivalent of the continuous-time differential equations (3.8). Many other time discretizations are possible and we will discuss some of them later on in this chapter. Similarly, discrete-time equivalents can be obtained for the LTV and nonlinear continuous-time state equations. For the general case of nonlinear systems with time-varying parameters, the discrete-time equivalent to equations (3.6) and (3.10) can be expressed as

$$\mathbf{x}_k = \mathbf{f}_k\left(\mathbf{u}_{k-1}, \mathbf{x}_{k-1}\right), \quad \mathbf{y}_k = \mathbf{h}_k\left(\mathbf{u}_k, \mathbf{x}_k\right) \ . \tag{4.58, 4.59}$$

Comparison with equation (4.47) shows that this implies that

[†] We use the shortcut notation $\mathbf{x}_k$ to indicate $\mathbf{x}(t_k)$, i.e. the value of $\mathbf{x}$ at $t = t_k$.

$$\mathbf{f}_k = \mathbf{x}_{k-1} + \mathbf{f}_{c,k-1}\Delta t \text{ and } \mathbf{h}_k = \mathbf{h}_{c,k}, \qquad (4.60, 4.61)$$

where we used subscripts $c,k-1$ and c,k to indicate continuous-time functions evaluated at discrete times $k-1$ and k respectively.

4.3.2 Implicit Euler discretization

Equation (4.47) is known as an *explicit Euler scheme*, where the term explicit refers to the fact that $\mathbf{x}_{k+1}$ can be obtained as an explicit formula in terms of $\mathbf{x}_k$. This is possible because we chose to evaluate the function $\mathbf{f}$ in equation (4.46) at the 'old' time t_{k-1}. If, alternatively, we choose to evaluate $\mathbf{f}$ at the 'new' time t_k and apply the result to the LTI equation (3.8) again we obtain

$$\mathbf{x}_k = \mathbf{x}_{k-1} + \left(\mathbf{A}_c\mathbf{x}_k + \mathbf{B}_c\mathbf{u}_k\right)\Delta t \ , \qquad (4.62)$$

As before, a more formal derivation of this result can be obtained by using a Taylor expansion; this time a backward one for $\mathbf{x}$ at t_k:

$$\mathbf{x}_{k-1} = \mathbf{x}_k - \left[\frac{d\mathbf{x}(t)}{dt}\right]_{t=t_k}\Delta t - \frac{1}{2}\left[\frac{d^2\mathbf{x}(t)}{dt^2}\right]_{t=t_k}\left(\Delta t\right)^2 - \ldots \ . \qquad (4.63)$$

Reordering the terms leads to:

$$\mathbf{x}_k = \mathbf{x}_{k-1} + \left[\frac{d\mathbf{x}(t)}{dt}\right]_{t=t_k}\Delta t + \frac{1}{2}\left[\frac{d^2\mathbf{x}(t)}{dt^2}\right]_{t=t_k}\left(\Delta t\right)^2 + \ldots \ , \qquad (4.64)$$

which is identical to expression (4.48) except for the time at which the derivatives are evaluated. Substitution of equation (3.8) in equation (4.64), and disregarding the terms higher than first order, leads to equation (4.62) again. This equation is known as an *implicit Euler scheme*, because $\mathbf{x}_k$ appears both at the left-hand and the right-hand side of the equation. It can be rewritten as

$$\left(\mathbf{I} - \Delta t\mathbf{A}_c\right)\mathbf{x}_k = \mathbf{x}_{k-1} + \Delta t\mathbf{B}_c\mathbf{u}_k, \qquad (4.65)$$

whereafter it can formally be solved for $\mathbf{x}_k$ as

$$\mathbf{x}_k = \left(\mathbf{I} - \Delta t\mathbf{A}_c\right)^{-1}\left(\mathbf{x}_{k-1} + \Delta t\mathbf{B}_c\mathbf{u}_k\right) \ , \qquad (4.66)$$

although in a numerical implementation it is, as always, more efficient to solve the linear system of equations (4.65) than to compute the inverse as in equation (4.66). Expression (4.66) can be rewritten in a form similar to equation (4.56) if we redefine $\mathbf{A}_d$ and $\mathbf{B}_d$ as

$$\mathbf{A}_d = \left(\mathbf{I} - \Delta t\mathbf{A}_c\right)^{-1}, \ \mathbf{B}_d = \left(\mathbf{I} - \Delta t\mathbf{A}_c\right)^{-1}\Delta t\mathbf{B}_c, \qquad (4.67, 4.68)$$

leading to the discrete state-space form

$$\mathbf{x}_k = \mathbf{A}_d\mathbf{x}_{k-1} + \mathbf{B}_d\mathbf{u}_k \ . \qquad (4.69)$$

Note that although equation (4.69) appears to be explicit in time again, the underlying implicit discretization scheme results in the need to solve a system of equations at each time step. An implicit Euler discretization of the nonlinear system of equations (3.6) can also be obtained by substitution in equation (4.64). Disregarding higher-order terms, this leads to

$$\mathbf{x}_k = \mathbf{x}_{k-1} + \Delta t\, \mathbf{f}_c\left(\mathbf{u}_k, \mathbf{x}_k\right) \ , \tag{4.70}$$

Or, with $\mathbf{f}_k = \mathbf{x}_{k-1} + \mathbf{f}_{c,k}\Delta t$, to

$$\mathbf{x}_k = \mathbf{f}_k\left(\mathbf{u}_k, \mathbf{x}_{k-1}, \mathbf{x}_k\right) \ . \tag{4.71}$$

It is, in general, not possible to invert the nonlinear function $\mathbf{f}_k$ and therefore we need an iterative procedure to solve equation (4.71) at every time step. The most simple procedure is *Picard iteration*, also known as *subsequent substitution* or *simple iteration*, in which we start by solving the equation with an initial guess $\mathbf{x}_k^0$ at the right-hand-side to obtain an improved estimate $\mathbf{x}_k^1$ at the left-hand side. The usual choice for $\mathbf{x}_k^0$ is simply the value $\mathbf{x}_{k-1}$ computed during the previous time step. Subsequent iterations steps can then be expressed as

$$\mathbf{x}_k^i = \mathbf{f}_k\left(\mathbf{u}_k, \mathbf{x}_{k-1}, \mathbf{x}_k^{i-1}\right) \ , \tag{4.72}$$

where the superscript i is the iteration counter. The iteration is terminated when a predefined convergence criterion is met. A typical criterion is given in terms of the two-norm

$$\left\|\mathbf{r}_k^i\right\|_2 \le \varepsilon \ , \tag{4.73}$$

where $\mathbf{r}_k^i = \mathbf{x}_k^i - \mathbf{x}_k^{i-1}$ is the *residual* of the iteration, and ε is a small number[†]. Another popular norm to specify convergence criteria is the infinity norm $\| \mathbf{r}_k^i \|_\infty$. Expression (4.73) is known as an *absolute* convergence criterion. An example of a *relative* criterion is

$$\frac{\left\|\mathbf{r}_k^i\right\|_2}{\left\|\mathbf{x}_k^i\right\|_2} \le \varepsilon \ . \tag{4.74}$$

In practice, it is often required that several convergence criteria are met simultaneously before an iteration may be terminated. An alternative to Picard iteration is *Newton-Raphson iteration*. The vectorial form of this iteration scheme can be expressed as the two-step procedure:

$$\begin{aligned} &\frac{\partial \mathbf{g}_k\left(\mathbf{u}_k, \mathbf{x}_{k-1}, \mathbf{x}_k^i\right)}{\partial \mathbf{x}_k^i}\mathbf{r}_k^i = \mathbf{g}_k\left(\mathbf{u}_k, \mathbf{x}_{k-1}, \mathbf{x}_k^i\right), \\ &\mathbf{x}_k^{i+1} = \mathbf{x}_k^i + \mathbf{r}_k^i, \end{aligned} \tag{4.75, 4.76}$$

where

$$\mathbf{g}_k\left(\mathbf{u}_k, \mathbf{x}_{k-1}, \mathbf{x}_k\right) \triangleq \mathbf{x}_k - \mathbf{f}_k\left(\mathbf{u}_k, \mathbf{x}_{k-1}, \mathbf{x}_k\right) = \mathbf{0} \tag{4.77}$$

is the implicit form of the system equations (4.71), and where

[†] For a definition of norms, see Section A.2 in Appendix B. The small number ε itself is often also called the convergence criterion. The state vector $\mathbf{x}$ may contain groups of elements with different physical dimensions (e.g. pressures and saturations), in which case the dimensions of ε are ill-defined. Moreover, the magnitudes of the pressures are usually much larger than those of the saturations (typically 10^6 - 10^7 versus 0 - 1) and therefore the pressure values determine whether or not the convergence criterion is met whereas the saturations have almost no influence. For multi-phase flow it is therefore required to specify separate convergence criteria for the pressures and the saturations, or to scale the variables such that they become dimensionless and of the same order of magnitude.

$$\frac{\partial \mathbf{g}_k\left(\mathbf{u}_k, \mathbf{x}_{k-1}, \mathbf{x}_k^i\right)}{\partial \mathbf{x}_k^i} = \begin{bmatrix} \frac{\partial g_1}{\partial x_1} & \frac{\partial g_1}{\partial x_2} & \cdots & \frac{\partial g_1}{\partial x_n} \\ \frac{\partial g_2}{\partial x_1} & \frac{\partial g_2}{\partial x_2} & \cdots & \frac{\partial g_2}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial g_n}{\partial x_1} & \frac{\partial g_n}{\partial x_2} & \cdots & \frac{\partial g_n}{\partial x_n} \end{bmatrix}_k^i , \tag{4.78}$$

is known as a *Jacobian* matrix. Equation (4.75) implies that we have to solve a system of linear equations to find $\mathbf{r}_k^i$ during each iteration step. Just as for Picard iteration, the convergence criterion for Newton-Raphson iteration can be specified in terms of the residual $\mathbf{r}_k^i$ in various ways. Newton-Raphson iteration generally converges faster than Picard iteration, especially in the close neighborhood of the root to which it converges. However, both methods may occasionally fail to converge in a reasonable number of iteration steps. Various ad-hoc measures to guide the iteration process, or to restart the process after failure, are therefore usually applied in numerical implementations.

For a reservoir-specific example consider the continuous-time state space representation for two-phase flow with or without well model as given in equations (3.142) and (3.144):

$$\dot{\mathbf{x}} = \mathbf{A}_c\left(\mathbf{x}\right)\mathbf{x} + \mathbf{B}_c\left(\mathbf{x}\right)\mathbf{u} \ . \tag{4.79}$$

Here we have, as before, added subscripts c to indicate that the secant matrices $\mathbf{A}_c$ and $\mathbf{B}_c$ represent a continuous- time formulation. Applying implicit Euler discretization results in

$$\mathbf{x}_k = \mathbf{x}_{k-1} + \Delta t \mathbf{A}_c\left(\mathbf{x}_k\right)\mathbf{x}_k + \Delta t \mathbf{B}_c\left(\mathbf{x}_k\right)\mathbf{u}_k \ , \tag{4.80}$$

or, formally,

$$\mathbf{x}_k = \mathbf{A}_d\left(\mathbf{x}_k\right)\mathbf{x}_{k-1} + \mathbf{B}_d\left(\mathbf{x}_k\right)\mathbf{u}_k \ , \tag{4.81}$$

where

$$\mathbf{A}_d\left(\mathbf{x}_k\right) = \left[\mathbf{I} - \Delta t \mathbf{A}_c\left(\mathbf{x}_k\right)\right]^{-1} , \quad \mathbf{B}_d\left(\mathbf{x}_k\right) = \Delta t\left[\mathbf{I} - \Delta t \mathbf{A}_c\left(\mathbf{x}_k\right)\right]^{-1}\mathbf{B}_c\left(\mathbf{x}_k\right) \ . \tag{4.82}$$

If we want to solve equation (4.81) using Newton-Raphson iteration, we could, formally, specify the implicit version of equation (4.81) in the form of a function $\mathbf{g}_k$ as

$$\mathbf{g}_k\left(\mathbf{u}_k, \mathbf{x}_{k-1}, \mathbf{x}_k\right) = \mathbf{x}_k - \mathbf{A}_d\left(\mathbf{x}_k\right)\mathbf{x}_{k-1} - \mathbf{B}_d\left(\mathbf{x}_k\right)\mathbf{u}_k \ , \tag{4.83}$$

and work out the Jacobian $\partial \mathbf{g}_k / \partial \mathbf{x}_k$. In practice, it will be more convenient to start from the version with continous-time matrices, as given in equation (4.80), such that $\mathbf{g}_k$ is expressed as:

$$\mathbf{g}_k\left(\mathbf{u}_k, \mathbf{x}_{k-1}, \mathbf{x}_k\right) = \left(\mathbf{I} - \Delta t \mathbf{A}_c\left(\mathbf{x}_k\right)\right)\mathbf{x}_k - \mathbf{x}_{k-1} - \Delta t \mathbf{B}_c\left(\mathbf{x}_k\right)\mathbf{u}_k \ . \tag{4.84}$$

Moreover, it is usually computationally more efficient to use the generalized state space formulation, which leads to

$$\mathbf{g}_k\left(\mathbf{u}_k, \mathbf{x}_{k-1}, \mathbf{x}_k\right) = \left(\hat{\mathbf{E}}_c\left(\mathbf{x}_k\right) - \Delta t \hat{\mathbf{A}}_c\left(\mathbf{x}_k\right)\right)\mathbf{x}_k - \hat{\mathbf{E}}_c\left(\mathbf{x}_k\right)\mathbf{x}_{k-1} - \Delta t \hat{\mathbf{B}}_c\left(\mathbf{x}_k\right)\mathbf{u}_k \ . \tag{4.85}$$

See Section A.6 in Appendix A for further details about the computation of derivatives involving matrix-vector products.

4.3.3 Stability

In Section 4.1.3 we addressed the stability of a continuous-time dynamical system, and we found that asymptotic stability requires that all eigenvalues of the system matrix **A** are smaller than zero. We also discussed, in Section 4.1.2, how a coupled system of n equations $\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t)$ can be transformed to a system of n uncoupled equations $\dot{\mathbf{z}}(t) = \mathbf{\Lambda}\mathbf{z}(t)$, where $\mathbf{z}(t) = \mathbf{M}^{-1}\mathbf{x}(t)$ and where $\mathbf{\Lambda} = \text{diag}\,(\lambda_1, \ldots, \lambda_n)$ is the diagonal matrix of eigenvalues of **A** with $\lambda_1 \le \lambda_2 \le \ldots \le \lambda_n$. The stability of the system is therefore governed by the stability of the individual differential equations of the uncoupled system:

$$\dot{z}_i(t) = \lambda_i z_i(t) \ . \tag{4.86}$$

In particular, if $\lambda_n < 0$, the system is asymptotically stable. If $\lambda_n = 0$, the system is marginally stable. To establish the stability properties of a time-discretized system of equations we will therefore consider the discretized version of equation (4.86). E.g. for the explicit Euler case, we can write:

$$z_{i,k} = (1 + \Delta t \lambda_i) z_{i,k-1} \ . \tag{4.87}$$

This recursive equation will asymptotically approach zero for large values of k if

$$-1 < (1 + \Delta t \lambda_i) < 1 \ . \tag{4.88}$$

This is equivalent to

$$0 < \Delta t < \frac{2}{-\lambda_i} . \tag{4.89}$$

If we restrict our attention to integration forward in time, i.e. to $\Delta t > 0$, inequalities (4.89) imply that the discretized system is only stable if

1. the underlying continuous-time system is stable, i.e. if $\lambda_i < 0$ for all i, otherwise Δt cannot fulfill both inequalities, and
2. the time step Δt is in between the bounds given by inequalities (4.89). This requirement is known as the *Courant-Friedrichs-Lewy (CFL) stability condition*, and the explicitly discretized system is therefore *conditionally* stable[†].

In case of explicit Euler discretization the stability is therefore governed by the negative eigenvalue with the largest absolute value, λ_1. In case of implicit Euler discretization we have

$$z_{i,k} = (1 - \Delta t \lambda_i)^{-1} z_{i,k-1} \ , \tag{4.90}$$

which will be asymptotically stable if

$$1 < |1 - \Delta t \lambda_i| \ . \tag{4.91}$$

If the underlying system is asymptotically stable, and if we take Δt positive, we find that this condition is always fulfilled. In other words, the implicitly discretized system is *unconditionally stable*. For nonlinear systems we may apply these concepts to the linearized

[†] The upper bound $\Delta t < -2/\lambda_1$ is also known as the *CFL limit*.

equations (3.24), i.e. to the tangent linear system, although the results are then only valid for a local neighborhood around each point along the state trajectory. The combined linearized pressure and saturation equations for porous media flow form a stiff system of equations, i.e. the ratio between the largest and smalles eigenvalues is very big. In particular, the absolute values of the eigenvalues corresponding to the pressure equations are much larger than those corresponding to the saturation equations. As a consequence the time step for explicit integration of the pressure equations becomes so small that for all practical purposes this is not an option. Through analysis of the linearized time-discetized saturation equations it can be shown that the stability limit for explicit integration is governed by grid blocks with the highest *throughput,* i.e. total flow rate per time step; see e.g. Aziz and Settari(1979) or Datta-Gupta and King (2007). In particular, for the case of one-dimensional incompressible two-phase flow without capillary forces the maximum time step for explicit Euler integration of the corresponding Buckley-Leverett equation is governed by the throughput condition:

$$\Delta t < \frac{\Delta x \phi \left(1 - S_{or} - S_{wc}\right)}{v_t v_D^*} \ , \tag{4.92}$$

where v_D^* is the dimensionless shock front velocity as defined in equation (2.91). Similar expressions can be obtained for more complicated cases. For all cases, higher total fluid velocities and smaller spatial grid block dimensions imply a smaller time step to maintain stability.

4.3.4 IMPES

Although the implicit formulation allows arbitrarily large time steps as far as stability is concerned, there is usually a time step restriction based on accuracy requirements. In particular, it is often required to restrict the saturation changes per time step such that they stay considerably below one. In that case the time step size is typically below the stability limit for the saturation equations, but above the limit for the pressure equations. A popular alternative for the time integration of equations for multiphase flow through porous media is therefore the *IMplicit Pressure – Explicit Saturation* (IMPES) scheme. In the IMPES scheme the equations are reorganized such that it is possible to solve for pressures and saturations separately, which allows for an implicit update of the pressures, and a stable explicit update of the saturations, using the same time step size. Since explicit updates do not involve the solving of equations, they are much faster than implicit updates, and therefore the IMPES scheme is computationally attractive. Alternatively, it is possible to use a large time step for the implicit pressure update (with a size above the stability limit for the explicit saturation updates), and use a smaller step size to perform multiple explicit saturation updates in between the pressure updates. To obtain the IMPES formulation, consider again the general continuous-time state space representation for two-phase flow with or without well model

$$\dot{\mathbf{x}} = \mathbf{A}_c\left(\mathbf{x}\right)\mathbf{x} + \mathbf{B}_c\left(\mathbf{x}\right)\mathbf{u} \ . \tag{4.93}$$

Recalling that the state vector $\mathbf{x}$ consists of the pressures $\mathbf{p}$ and the saturations $\mathbf{s}$, we may partition equation (4.93) as

$$\begin{bmatrix} \dot{\mathbf{p}} \\ \dot{\mathbf{s}} \end{bmatrix} = \begin{bmatrix} \mathbf{A}_p\left(\mathbf{s}\right) & \mathbf{0} \\ \mathbf{A}_s\left(\mathbf{s}\right) & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{p} \\ \mathbf{s} \end{bmatrix} + \begin{bmatrix} \mathbf{B}_p\left(\mathbf{s}\right) \\ \mathbf{B}_s\left(\mathbf{s}\right) \end{bmatrix} \mathbf{u} \ , \tag{4.94}$$

which can also be written as two separate systems of equations:

$$\dot{\mathbf{p}} = \mathbf{A}_p(\mathbf{s})\mathbf{p} + \mathbf{B}_p(\mathbf{s})\mathbf{u}\,, \qquad \dot{\mathbf{s}} = \mathbf{A}_s(\mathbf{s})\mathbf{p} + \mathbf{B}_s(\mathbf{s})\mathbf{u}\,. \tag{4.95, 4.96}$$

Equation (4.95) is a linear differential equation for $\mathbf{p}$ that can be solved implicitly as†:

$$\mathbf{p}_k = \left[\mathbf{I} - \Delta t \mathbf{A}_p(\mathbf{s}_{k-1})\right]^{-1}\left[\mathbf{p}_{k-1} + \Delta t \mathbf{B}_p(\mathbf{s}_{k-1})\mathbf{u}_k\right]\,, \tag{4.97}$$

or, computationally more efficiently, as

$$\left[\mathbf{I} - \Delta t \mathbf{A}_p(\mathbf{s}_{k-1})\right]\mathbf{p}_k = \mathbf{p}_{k-1} + \Delta t \mathbf{B}_p(\mathbf{s}_{k-1})\mathbf{u}_k\,. \tag{4.98}$$

Equation (4.96) is a nonlinear equation for $\mathbf{s}$, that can be solved explicitly as

$$\mathbf{s}_k = \mathbf{s}_{k-1} + \Delta t \mathbf{A}_s(\mathbf{s}_{k-1})\mathbf{p}_k + \Delta t \mathbf{B}_s(\mathbf{s}_{k-1})\mathbf{u}_k\,. \tag{4.99}$$

Note that it is possible to use the input at time k in equations (4.98) and (4.99) and the pressure at time k in equation (4.99), but that we have to use the saturation at time k-1 in equation (4.98). Of course it is possible to repeat the implicit pressure computation (4.98) with the new saturation vector $\mathbf{s}_k$ as obtained from equation (4.99) and repeat this process until convergence. Moreover, also the saturation update may be performed implicitly, in which case we obtain a scheme known as the *sequential solution method*. For an alternative, more traditional, derivation of the IMPES scheme, and for a detailed analysis of properties such as mass conservation, stability, and accuracy, consult Aziz and Settari (1979) or Chen et. al (2006). In the special case that the fluid and rock compressibilities can be taken as zero, i.e. in the case of *incompressible flow*, the IMPES equation (4.98) for the pressures reduces to an algebraic equation

$$\tilde{\mathbf{A}}_p(\mathbf{s}_{k-1})\mathbf{p}_k = \tilde{\mathbf{B}}_p(\mathbf{s}_{k-1})\mathbf{u}_k\,, \tag{4.100}$$

where

$$\tilde{\mathbf{A}}_p(\mathbf{s}_{k-1}) = \mathbf{T}_w(\mathbf{s}_{k-1}) + \mathbf{T}_o(\mathbf{s}_{k-1})\,, \quad \tilde{\mathbf{B}}_p(\mathbf{s}_{k-1}) = \left[\mathbf{F}_w(\mathbf{s}_{k-1}) + \mathbf{F}_o(\mathbf{s}_{k-1})\right]\mathbf{L}_{qu}\,, \tag{4.101, 4.102}$$

as has been derived using the material from Section 2.4.11. The incompressible IMPES equation for saturations still has the form of equation (4.99), but with modified system and input matrices†:

$$\mathbf{A}_s(\mathbf{s}_{k-1}) = \mathbf{V}_{ws}^{-1}\mathbf{T}_w(\mathbf{s}_{k-1})\,, \quad \mathbf{B}_s(\mathbf{s}_{k-1}) = \mathbf{V}_{ws}^{-1}\mathbf{F}_w(\mathbf{s}_{k-1})\mathbf{L}_{qu}\,. \tag{4.103, 4.104}$$

4.3.5 Stream line simulation*

The key element of the IMPES and sequential simulation schemes is the separate solution of two sets of equations, one for pressures and one for saturations, which are only mildly coupled through the coefficients. A further step can be made by redefining the saturation equations such that they can be expressed as a system of decoupled equations that can be solved independently from each other. The basis for this redefinition is the insight that the

† In the more general case that $\mathbf{A}_p$ is a continuous function of p, an iterative implicit solution using Picard or Newton iteration will be required.

† Here we have chosen, arbitrarily, to base the incompressible IMPES saturation equation on equation (2.132) which is expressed in terms of $\mathbf{V}_{ws}$, $\mathbf{T}_w$ and $\mathbf{F}_w$. We could just as well have used equation (2.133), which would have resulted in an expression in terms of $\mathbf{V}_{os}$, $\mathbf{T}_o$ and $\mathbf{F}_o$.

saturation equations are mainly convective, or, in other words, that the saturation changes in the reservoir are mainly driven by a velocity field[‡]. Here the velocity field refers to the total fluid velocity, i.e. the sum of oil and water velocities. As discussed in Section 3.4.5 the velocity field can be 'traced' to generate streamlines, i.e. trajectories of 'virtual' particles traveling through the reservoir. Assuming incompressible flow and a situation where the flow is entirely driven by injection and production wells, the streamlines all start at an injector and end at a producer. The time it takes a particle to travel from the injector to a certain point $\hat{s}$ along its streamline is known as the *time of flight* τ which can be expressed as

$$\tau = \int_0^{\hat{s}} \frac{\phi}{v_t} ds \ , \tag{4.105}$$

where $v_t = |\mathbf{v}_t|$ is the magnitude of the total Darcy velocity, ϕ is porosity and s is a coordinate along the streamline starting at the injector. Equation (4.105) can be differentiated with respect to s resulting in the relationship

$$\frac{d\tau}{ds} = \frac{\phi}{v_t} \ , \tag{4.106}$$

which can be used to convert expressions in terms of streamline coordinate s to equivalent expressions in terms of time of flight τ. In particular, consider the one-dimensional Buckley-Leverett equation (2.73) expressed in streamline coordinate s:

$$v_t \frac{\partial f_w}{\partial S_w} \frac{\partial S_w}{\partial s} + \phi \frac{\partial S_w}{\partial t} = 0 \ . \tag{4.107}$$

With the aid of equation (4.106) we can express the Buckley-Leverett equation (4.107) in terms of time of flight coordinate τ as

$$\frac{\partial f_w}{\partial S_w} \frac{\partial S_w}{\partial \tau} + \frac{\partial S_w}{\partial t} = 0 \ . \tag{4.108}$$

In analogy to the analytical solution for the Buckley-Leverett equation derived in Section (2.4.5), see equation (2.93), we can express the solution of equation (4.108) as

$$\tau\left(S_w, t\right) = \begin{cases} \dfrac{df_w}{dS_w} t, & S_w^* \le S_w \le 1 - S_{or} \\ v_D^* t, & S_{wc} \le S_w \le S_w^* \end{cases} \ , \tag{4.109}$$

where the dimesnionless shock velocity v_D^* is given by equation (2.91). Note that the time of flight, although expressed in units of time, has taken the role of the spatial coordinate. Expression (4.109) allows us to determine the saturation at a point along a streamline if the corresponding time of flight is known. This is a powerful relationship because it is easy to compute the time of flight along a streamline once the velocity field has been computed. This, in turn, is a simple step once the pressure field has been computed, as was discussed in Section 2.4.12; see equation (2.144). The streamlines can then be traced using the expressions given in Section 3.4.5, which include the computation of the grid block travel times $\Delta\tau$, see

[‡] In our case, where we neglected the diffusive effect of capillary pressures, the continuous form of the saturation equation is in fact completely convective. The spatial discretization brings back some numerical diffusion again.

equation (3.173), which can be summed to obtain the time of flight. In practice, equation (4.108) is usually solved with the aid of finite differences, especially when complicating effects such as compressiblity and gravity have to be accounted for. A major advantage of computing the saturations using finite differences along streamlines instead of using finite differences on a conventional spatial grid is an improved stability criterion for explicit time stepping. As discussed in Section 4.3.3, the stability condition for explicit Euler integration of the saturation equations is governed by grid blocks with the highest throughput, i.e. those with the smallest spatial dimensions and the highest total velocities. In the numerical simulation of saturations along streamlines it is possible to select the 'grid' size, in terms of time of flight increments, independently from the underlying spatial grid, such that much larger time steps can be accomodated. In addition, the stability condition can be determined independently for each streamline, such that for streamlines with low velocities a much larger time step can be chosen than for those with high velocities. Because of this reason, streamline simulation is popular for the fast computation of saturation fields. Streamline simulation becomes particularly attractive in situations where only infrequent updating of the pressure field is required, and therefore also only infrequent repetition of the stream line tracing procedure is needed. The most popular applications involve water flooding with small (or no) compressiblity, no or modestly nonlinear relative permeabilities, and fixed wells settings. However the application area is becoming much wider and we refer to Bratvedt, Gimse and Tegnander (1996), Batycky, Blunt and Thiele (1997), King and Datta Gupta (1998) and to the text book of Datta-Gupta and King (2007) for further reading.

4.3.6 Computational aspects

- The implicit formulation (4.85) has been implemented in `simsim`, with the option to solve the nonlinear system either with Picard iteration or with Newton-Raphson iteration. In addition, `simsim` can integrate the equations using an IMPES scheme. For completeness sake `simsim` can also perform explicit integration, but in practice the severe time step restrictions required to maintain stability make explicit integration of no value except for very small problems.
- In `simsim` the wells are represented using a well model, and the user can specify either the bottom hole pressure (BHP) or the total flow rate for each well. In addition it is possible to prescribe a maximum BHP for an injector with a prescribed rate, or a minimum BHP for a producer with a prescribed rate. Similarly, the user can prescribe maximum or minimum rates for wells with prescribed BHPs. Every time step the integration algorithm checks for violation of the constraints, and if so, recomputes the time step with a new prescribed well condition. This implies that a well that is controlled on BHP may become a rate-controlled well or vice versa. In fact the prescription of a rate or BHP can be considered a constraint itself, and the program simply checks that at any moment in time the *most constraining constraint* is active.
- For explicit integration the time step size is limited to maintain stability, as was discussed in Section 4.3.3. For implicit integration, which is mostly used in reservoir simulation, the time step limitation is governed by accuracy requirements for which there exist no hard rules. In `simsim` a variable step size algorithm has been implemented following Aziz and Settari (1979). Is aims at maintaining pressure and saturation changes at or below prescribed levels Δp_{target} and $\Delta S_{w,target}$ for each time step by adjusting the new time step based on the converged results from the previous time step accordingto:

$$\Delta t_{new} = \min\left[\Delta t_{old}\left(\frac{\Delta p_{target}}{\Delta p}\right), \Delta t_{old}\left(\frac{\Delta S_{w,target}}{\Delta S_w}\right)\right]. \qquad (4.110)$$

Because this involves extrapolation from the previous time step, the actual pressure and saturation changes will sometimes somewhat overshoot the target values. Therefore optional maximum allowed changes may be specified which, if exceeded, will trigger repetition of the integration step with a reduced step size. Moreover, the step size may be limited to stay below a maximum allowed value.

- Solution of the linear system of equations within each Newton-Raphson iteration can be performed in MATLAB using the backslash operator, which invokes a *direct solution method*, or using an *iterative solution method*, in conjunction with a *pre-conditioner*. Discussion of these numerical mathematics aspects is outside the scope of this text, and we refer to e.g. Chen *et al*. (2006) for a detailed discussion. In `simsim` the choice is between the backslash operator for relatively small systems, and a biconjugate-gradient iterative solver with incomplete LU decomposition as preconditioner for larger systems.
- The implicit simulation using Newton-Raphson iterations can often be accelerated by restricting the update of the Jacobian at each iteration to those elements that correspond to grid blocks where a certain minimum saturation change has occurred. Other numerical 'tuning' parameters, which may be either hard-coded or user defined are, for example, a maximum allowable number of iterations and corresponding shrinkage and growth factors for the time step size which are applied depending on whether or not the maximum is reached. Often, the values of such parameters are problem-dependent, and some trial and error is required to find their optimal values.
- Adaptive Implicit Methods (AIM); staged preconditioning (to be written).

4.4 Examples

4.4.1 Example 1 continued – Stability

Stability limit

We perform a numerical integration of the state equations for Example 1 as defined in Section 3.3.2. We choose the input vector as $\mathbf{u}^T = [0.01\ -0.01]$, which implies that the wells in grid blocks 1 and 6 inject and produce at a rate of 0.01 m^3/s (864 m^3/d, or 5434 bpd). Note that, in line with the assumptions in Appendix A, a negative flow rate implies production. If we integrate with an explicit Euler scheme from $t = 0$ until $t = 365 \times 24 \times 3600$ s (i.e. for one year) with a time step of 1 day we obtain the output depicted in Figure 4.1. The pressure increase at the injector is smaller than the pressure decrease at the producer because of the different permeabilities in the corresponding grid blocks. If we use a time step of 5.8 days we obtain the spurious result as depicted in Figure 4.2, because we exceeded the stability limit (the CFL condition) for explicit Euler integration.

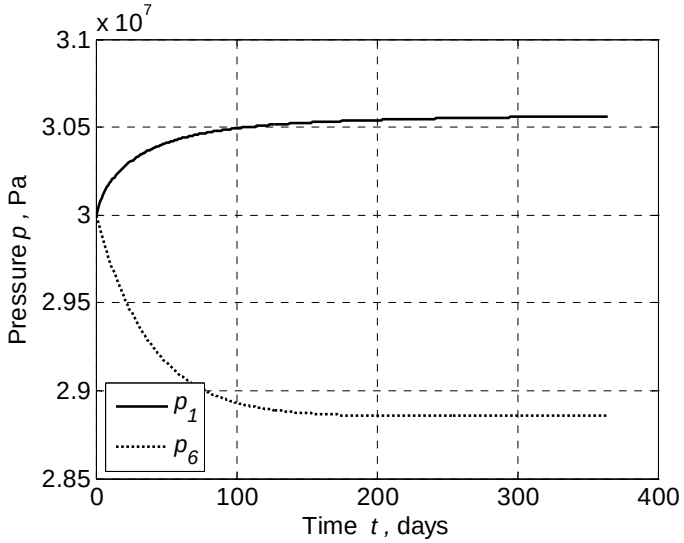


Figure 4.1: Numerical integration of Example 1 using explicit Euler integration with a time step of 1 day.

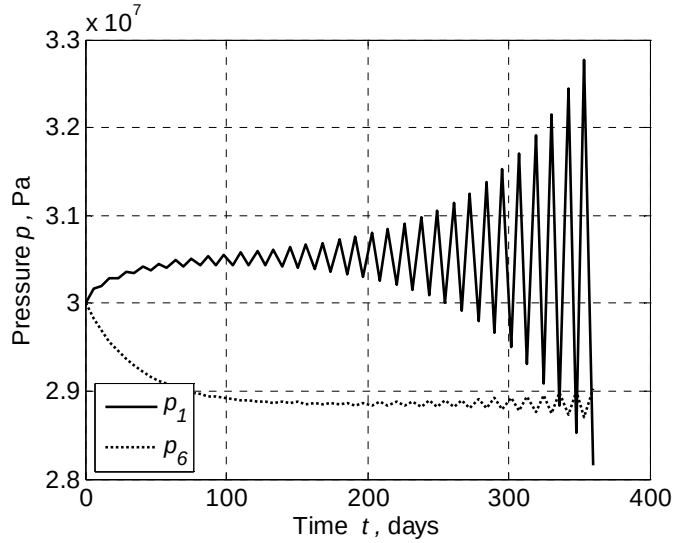


Figure 4.2: Numerical integration of Example 1 using explicit Euler integration with a time step of 5.8 days, displaying numerically unstable behavior.

Singular system matrix

Earlier we computed the eigen values of **A**, see equation (4.22), and we found a zero eigenvalue reflecting that **A** is singular with a rank deficiency of 1. In Section 4.2.3 it was shown that this implies the impossibility to compute the steady state solution. Here we mention another, numerical, effect. In our case the zero eigenvalue is equal to zero up to 14 significant digits. However, because of the finite precision of the numerical computations in MATLAB, a 'zero' eigenvalue may sometimes have a small positive or negative value. Because a positive eigenvalue corresponds to an exponentially growing response, this may introduce a solution that slowly drifts away from its correct steady-state value if the integration is pursued long enough. For our example the effect is not an issue, but, in general, time integration with a singular system matrix may cause problems for long integration periods.

Regular system matrix

We again integrate the system equations for Example 1. However, in this case the bottom hole pressure of the production well in grid block 6 has been prescribed as $p_{wf,6} = 26.00 \times 10^6$ Pa (3771 psi), while the injection rate in block 1 remains fixed at $q_1 = 0.01$ m^3/s (864 m^3/d, 5434 bpd). Figures (4.3) and (4.4) give a plot of the output variables versus time.

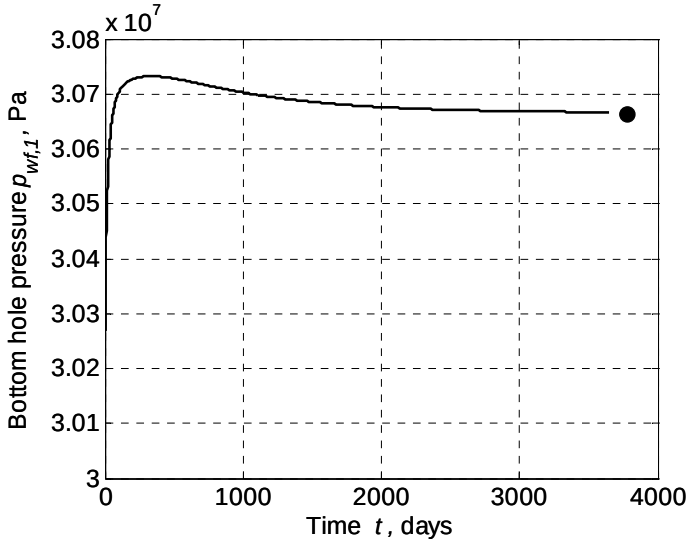


Figure 4.3: Numerical integration of Example 1 with a prescribed pressure in grid block 6. The figure shows the bottom hole pressure in the injection well in grid block 1. The dot represents the steady-state result.

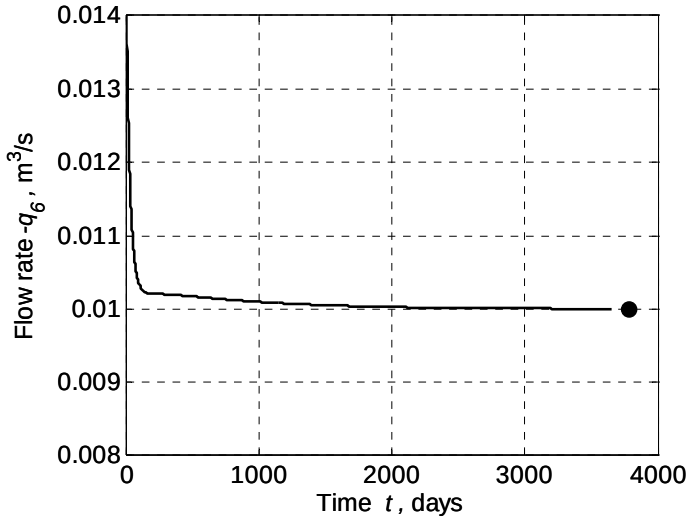


Figure 4.4: Numerical integration of Example 1 with a prescribed pressure in grid block 6. The figure shows the absolute value of the flow rate in the production well in grid block 6. The dot represents the steady-state result.

The solid dots represent the steady-state results computed with the aid of equations (4.40) and (4.41). Note that we needed to integrate for a period of around 3000 days before the pressure in the injection well approached the steady state result closely (3.06668 MPa after 10 years vs. 3.06655 MPa fully steady-state).

4.4.2 Example 2 continued – Mobility effects

For two-phase flow we present examples obtained with the `simsim` simulator, of which a brief description is given in Appendix C. We start with the forward simulation of Example 2, i.e. the same six-block model that was used to illustrate single-phase flow behavior, but with additional reservoir and fluid properties as given earlier in Section 2.4.4. We choose the operating conditions such that water is injected at a constant rate of 0.01 m^3/s in grid block 1, while liquid is produced at a constant well bore pressure of 20 MPa in grid block 6. Because of the very small size of the model we can use explicit Euler integration, and Figure 4.5 depicts the output for a simulation time of 10000 days (approximately 27 years). The total injected water volume is 8.64×10^6 m^3 which amounts to 2.4 times the total volume of moveable oil[†]. In the top-right figure it can be seen that water breakthrough in the producer occurs after about 3000 days.

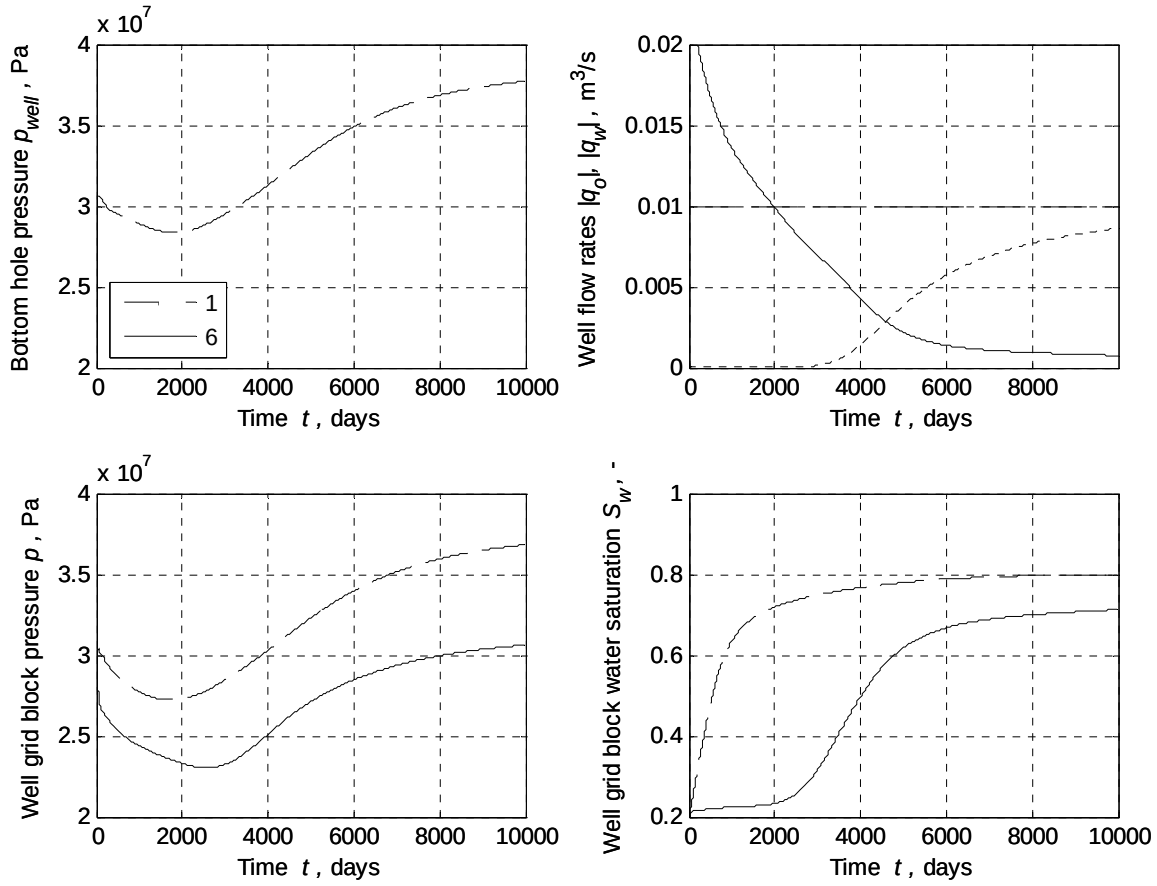


Figure 4.5: Results for numerical integration of Example 2 with a prescribed water injection flow rate in grid block 1 and a prescribed pressure in grid block 6. The solid and dotted lines in the top right figure represent the oil and water production rates in grid block 6; the dashed line represents the injection water rate in grid block 1. In the other three figures the dashed and solid lines refer to results for the injection and production wells in grid blocks 1 and 6 respectively.

[†] The moveable oil volume is equal to the pore volume times $(1 - S_{wc} - S_{or})$.

The bottom right figure displays the water saturations in the well grid blocks. Starting from connate water saturation (0.2) they approach a value of one minus residual oil saturation (0.8) more or less gradually. The relatively large grid blocks, compared to the total domain, cause a large amount of numerical diffusion. The two figures at the left illustrate that the well bore pressure in the injector, and the grid block pressure in both well grid blocks behave non-monotonously. Initially they drop because the prescribed pressure in the producer (20 MPa) is considerably below the initial reservoir pressure (30 MPa). However, when the oil-water front approaches the producer the total relative mobility in the producer grid block decreases because of the nonlinear saturation dependency of the relative permeabilities (see the dotted line in Figure 2.4) and because of the viscosity difference between oil and water (0.5×10^{-3} versus 1.0×10^{-3} Pa s respectively). The resulting increased pressure drop over the near well bore area of the producer, as represented in the well model, results in an increase in pressure in the entire reservoir. Note that if the injector had been operated at a prescribed pressure instead of at prescribed rate, this mobility effect would have resulted in a drop in total production rate instead of an increase in reservoir pressure.

4.4.3 Example 3 continued – Well constraints

Next we consider the forward simulation of Example 3 which was described in Section 2.4.9. The initial operating constraint for the injector is specified as a prescribed rate of 0.002 m^3/s (1087 bbl/d) with a maximum bottom hole pressure constraint equal to 35 MPa (5076 psi) which is 5 MPa (725 psi) above the initial reservoir pressure. The initial operating constraint for the producers is a prescribed pressure of 25 MPa (3626 psi), i.e. 5 MPa (725 psi) below the initial reservoir pressure, with a maximum flow rate per well equal to –0.001 m^3/s (–543 bbl/d). The initial water saturation is equal to the connate water saturation (0.2), and the total moveable oil volume is 16700 m^3 (1.05×10^6 bbl). We use implicit Euler integration with Newton-Raphson iteration, and a variable time step with maximum allowed pressure and saturation changes of 1×10^6 Pa and 0.2 repectively, target changes equal to 90% of these values, and a maximum time step of 30 days. Because of the relatively small problem size (882 states) the linear system of equations within each Newton-Raphson iteration is solved with the aid of the Matlab backslash operator (i.e. with a direct solver). Figure 4.6 displays the output for a simulation time of 1500 days (approximately 4.1 years). In the top left figure it can be seen that the bottom hole pressure in the producers stays at its prescribed pressure of 25 MPa (3626 psi) during the entire period and in the top right figure it can be verified that the production rates in the producers never exceed the maximum allowed flow rate of –0.001 m^3/s (–543 bbl/d). The injector, however runs against its pressure constraint after approximately 700 days: the top left figure shows that until that time the pressure stays below the constraint of 35 MPa (5076 psi) and, correspondingly, the top right figure shows a steady injection rate of 0.002 m^3/s (1087 bbl/d). After reaching the constraint the injector is effectively operating at a prescribed bottom hole pressure of 35 MPa (5076 psi) with a maximum rate constraint equal to 0.002 m^3/s (1087 bbl/d). This new constraint is not reached anymore in the remaining time, so no further constraint switches occur. The bottom right figure displays the water saturations in the well grid blocks an it can clearly be seen that there is a considerable difference in arrival time of the water front in the four producers. The same effect can be observed in the oil and water well flow rates depicted in the top right figure.

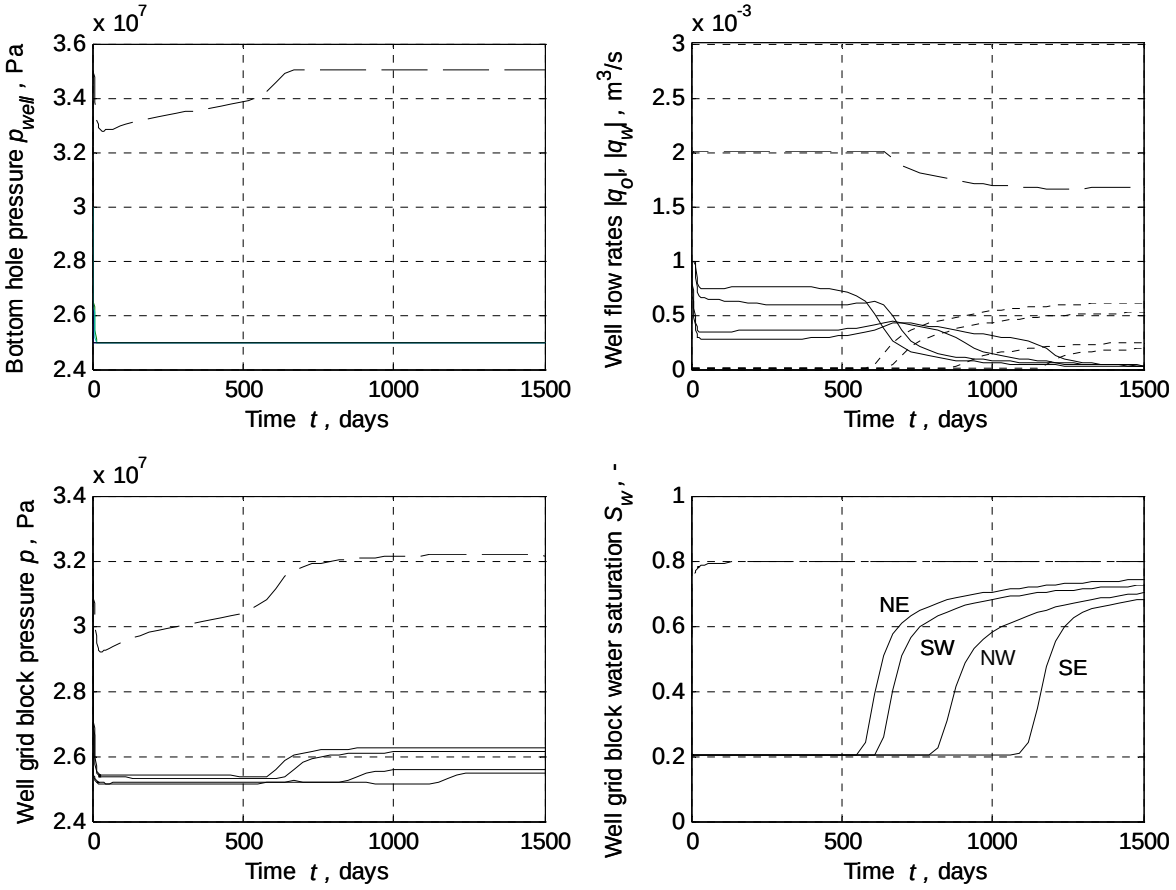


Figure 4.6: Well production data for Example 3. The solid and dotted lines in the top right figure represent the oil and water production rates in the producers; the dashed line represents the injection water rate. In the other three figures the four solid lines refer to the four producers, and the dashed line to the injector. The letters NE refer to the North-East (top right) producer, the letters SW to the South-West (bottom left producer), and so on.

The bottom left figure displays the grid blocks pressures, and, just as in Example 2, displays a clear mobility effect when the water front reaches the producers. The effect is an increase in pressure which rapidly spreads through the entire reservoir and which therefore results in the injector reaching its maximum bottom hole constraint as was described above. The corresponding injection and production field rates, i.e. the sums of the well rates, have been depicted in Figure 4.7. They show a typical oil production plateau followed by a rapid decline and a simultaneous increase in water production. The decline in oil production is not only caused by the increase in water cut in the producers but also by the inability of the injector to maintain its maximum rate because it has run into its pressure constraint. Figure 4.8 depicts 8 snapshots of the water saturation at different moments in time. The effect of the high permeable streak can clearly be seen: water breakthrough occurs in the North-East corner first, followed by the South-West, North-West and South-East corners, a sequence that is in line with the saturation curves in Figure 4.6 (bottom right).

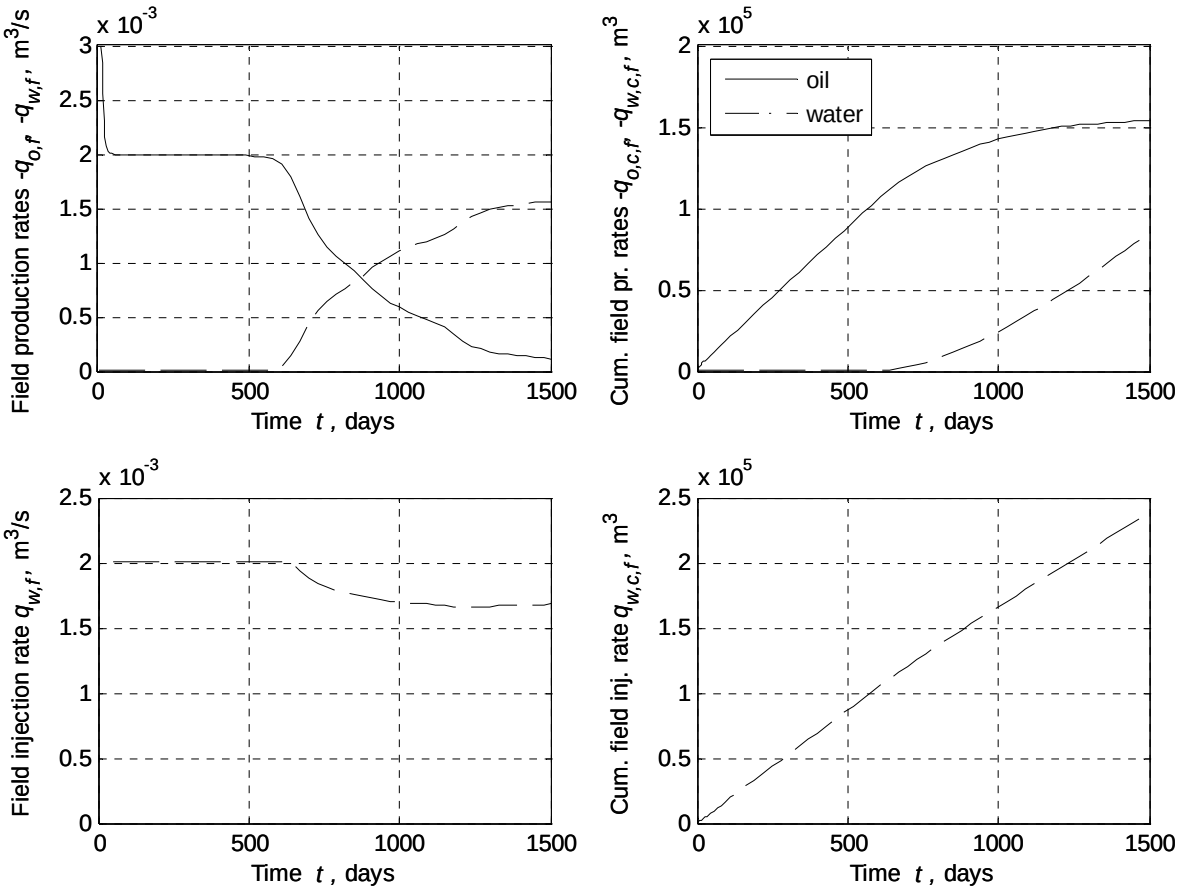


Figure 4.7: Field production data for Example 3. The solid and dotted lines represent oil and water rates respectively.

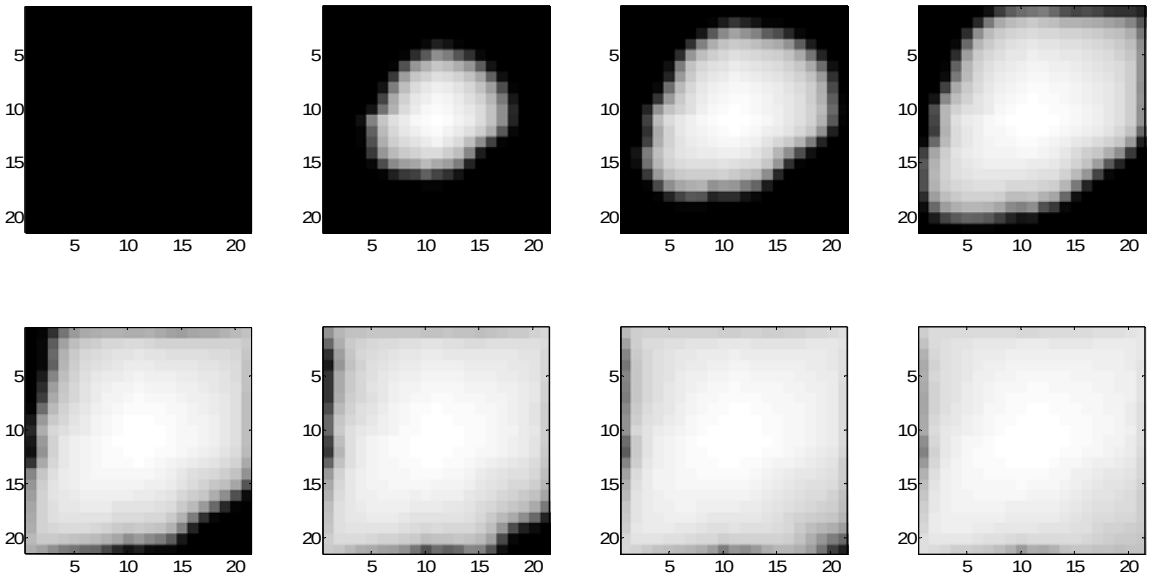


Figure 4.8: Snapshots of the water saturation field at time intervals of approximately 214 days (7 months) on a scale from connate water saturation (0.2, black) to one minus residual oil saturation (0.8, white). Initially (top left) the field is entirely at connate water saturation. After 1500 days (bottom right) the field is approaching residual oil saturation.

4.4.4 Example 3 continued – Time stepping statistics

Figures 4.9 displays several numerical parameters that give an indication of the functioning of the implicit variable-time step integration process. The top left graph displays the number of Newton iterations per time step, and it can be observed that around time steps 3 and 17 the convergence became somewhat problematic. As can be seen from the top right graph the early iteration problems are related to a high number of constraint violations per time step.

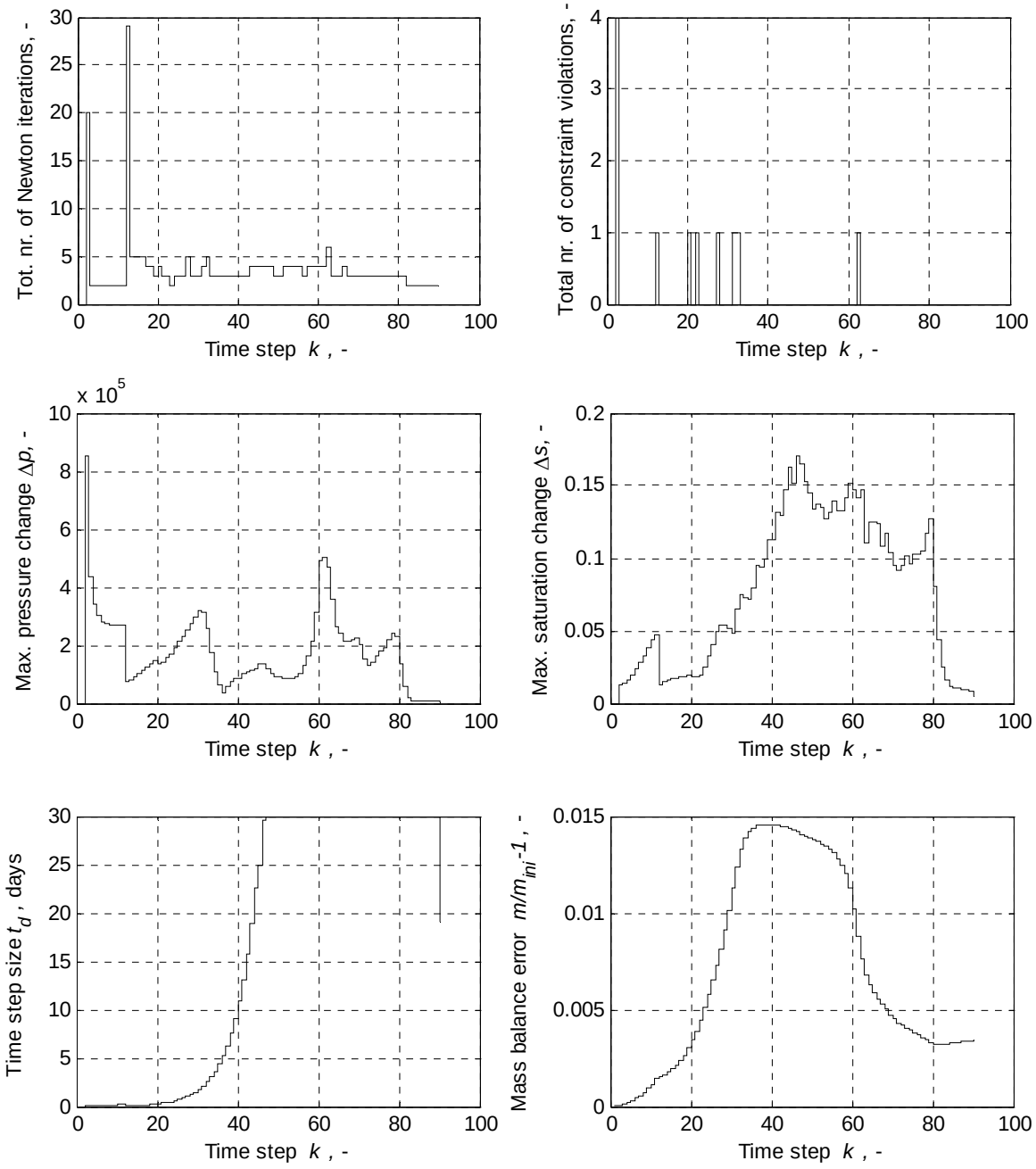


Figure 4.9: Numerical parameters. Total number of Newton-Raphson iterations per time step (top left), total number of constraint violations of bottom hole pressure or total well rate per time step (top right), maximum grid block pressure change per time step (middle left), maximum grid block saturation change per time step (middle right), time step size (bottom left), and mass balance error (bottom right).

In response, the time stepping algorithm repeats the Newton-Raphson iterations with progressively reduced time step size until convergence without constraint violation is reached. The targets for gridblock pressure and saturation changes were specified as 90% of 1×10^6 Pa and 0.2 respectively, with a maximum growth factor of 0.7 per time step, and a maximum time step size of 30 days. It can be seen from the two graphs in the middle row of Figure 4.9 that the target values were never met. This is because initially the time step size was occasionally reduced to obtain convergence, and as of time step 48 because the maximum allowed time step size was reached; see also the bottom left graph of Figure 4.9 which displays the size of each time step. The bottom right graph of Figure 4.9 displays the mass balance error ε_m during each time step k defined as

$$\varepsilon_{m,k} \triangleq \frac{m_k - m_0}{m_0} \equiv \frac{m_{1,k} + m_{2,k} + m_{3,k} + m_{4,k} - m_{5,k}}{m_{1,0} + m_{3,0}} - 1 \ , \tag{4.111}$$

where m_1 is the mass of oil in all grid blocks, m_2 the cumulative mass of produced oil, m_3 the mass of water in all grid blocks, m_4 the cumulative mass of produced water, and m_5 the cumulative mass of injected water. A small mass balance error develops during the simulation because we do not use a fully mass-conservative formulation, but the maximum error never exceeds 1.5% in this example, while at the end of the simulation it is less than 0.4%.

4.4.5 Example 3 continued – System energy*

Figure 4.10 illustrates the power balance for Example 3, where the various contributions have been computed in `simsim` with the aid of equation (3.176) of Section 3.4.6. The top left figure illustrates that during a brief initial period potential energy is released from the reservoir through oil flow, but that rapidly an equilibrium is established during which the total amount of potential energy stored stays nearly constant and close to zero. The top-right graph in Figure 4.10 displays the energy dissipation caused by oil and water flow through the grid block boundaries and in the near-well bore area. Note that the ratio between near-well bore and gridblock losses would become progressively smaller with decreasing grid size. Not surprisingly, the dissipation caused by oil flow reduces with time while the dissipation caused by water flow increases, corresponding to the increasing water-oil ratio of the produced reservoir fluids. The effects of relative permeabilities are visible in the small increase halfway the producing reservoir life when the water front reaches the producers. The bottom-left graph displays the power flow in the wells. It can be seen that there is an influx of energy through the injector and an outflow through the producers, resulting in a near-constant net influx of approximately 17 kW with a small peak to just above 20 kW.

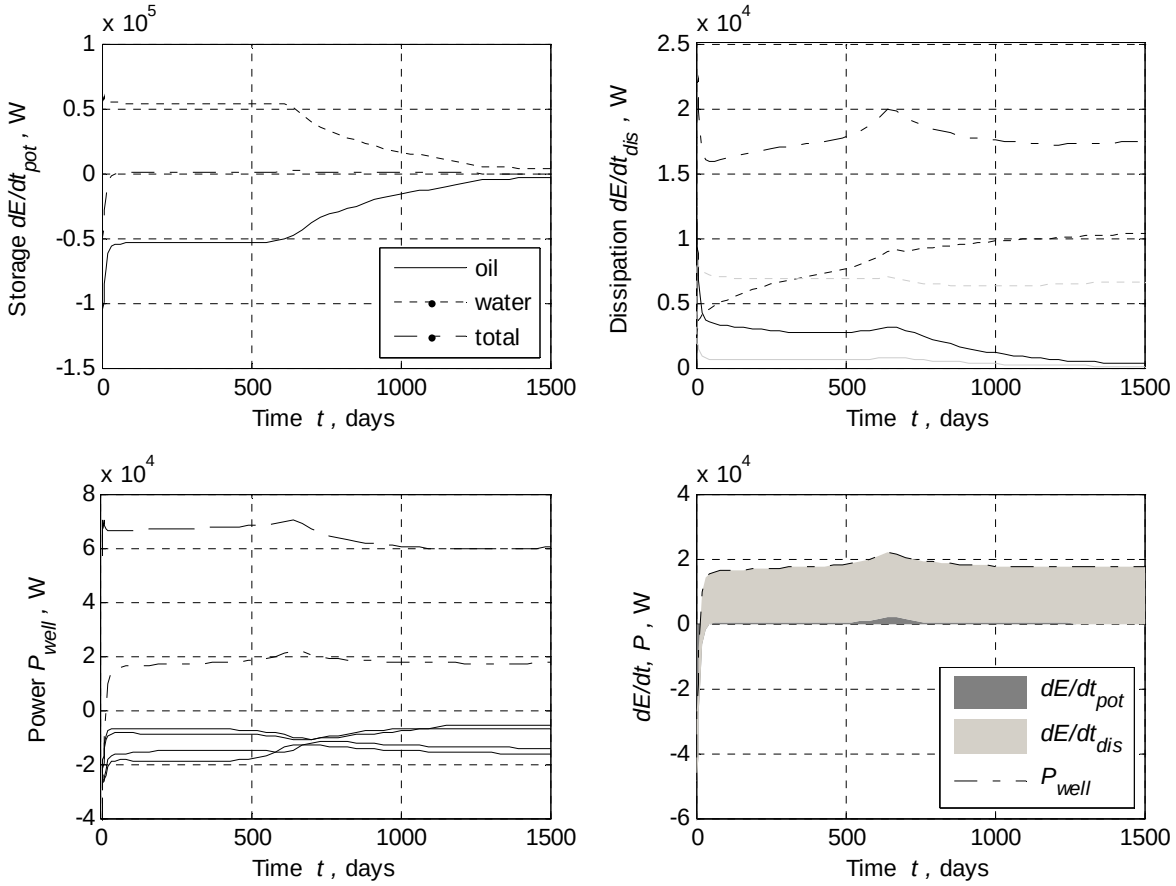


Figure 4.10: Power balance for Example 3. The solid, dotted and dash-dotted lines in the top left figure represent the oil, water and total potential energy storage rates respectively. The solid, dotted and dash-dotted lines in the top right figure represent the oil, water and total disspation rates respectively, where the black solid and dotted lines refer to dissipation through the grid block boundaries, and the light gray solid and dotted lines to dissipation in the near-well bore area. In the bottom-left figure the four solid lines refer to the power flow in four producers, the dashed line to the power flow in the injector, and the dash-dotted line to the total power flow in the wells. The bottom-right figure illustrates that the total power flow in the wells (dash-dotted line) equals the sum of the total potential energy storage rate (dark gray area; hardly visible) and the total dissipation rate (light gray area).

Note that these values do not take into account the effect of elevation-related potential energy, which would change the situation. E.g. if we would assume that the reservoir were located at a depth of $d = 3000$ meter, and were initially hydrostatically pressured with oil and water densities $\rho_w = 850$ kg/m^3 and $\rho_w = 1000$ kg/m^3 , and an acceleration of gravity $g = 9.81$ m/s^2 , then the elevation-related energy in a totally oil-filled well would be equal to

$$E_{lift} = (\rho_w - \rho_o) gd = (1000 - 850) \times 9.81 \times 3000 = 4{,}414{,}500 \text{ J} . \qquad (4.112)$$

Assuming a well bore diameter of radius $r_{well} = 0.114$ m , the total well volume would be

$$V_{well} = \pi r_{well}^2 d = 3.14 \times 0.114^2 \times 3000 = 122.4 \text{ m}^3 , \qquad (4.113)$$

such that with an average production rate of $q_o = 0.5$ m^3/s the well bore contents would be emptied in

$$t_{lift} = \frac{V_{well}}{q_o} = \frac{122.4}{0.5} = 245 \text{ s} \ . \tag{4.114}$$

The elevation-related lifting power of a completely oil filled-well would then be

$$P_{lift} = \frac{E_{lift}}{t_{lift}} = \frac{4{,}414{,}500}{245} = 18018 \text{ W} \ . \tag{4.115}$$

This simple analysis does not even take into account the additional lift effect of gas escaping from oil in the well bore. However, it should be noted that if we would not inject water, a very rapid reduction in reservoir pressure would occur and soon after start of production the wells would stop flowing.

4.5 References for Chapter 4

Aziz, K. and Settari, A., 1979: *Petroleum reservoir simulation*, Applied Science Publishers, London.

Batycky, R.P., Blunt, M.J. and Thiele, M.R. 1997: A 3D field-scale streamline-based reservoir simulator. *SPE Reservoir Engineering* **12** (4) 246-254. DOI: 10.2118/36726-PA.

Boyce, W. and Di Prima, R.C., 2005: *Elementary differential equations and boundary value problems, 8th ed.*, Wiley, New York.

Bratvedt, F., Gimse, T. and Tegnander, C. (1996), Streamline computations for porous media flow including gravity. *Transport in Porous Media* **25** (1) 63-78. DOI: 10.1007/BF00141262.

Chen, Z., Huan, G. and Ma, Y., 2006: *Computational methods for multiphase flows in porous media*, SIAM, Philadelphia.

Datta-Gupta, A. and King, M.J., 2007: *Streamline simulation: Theory and Practice*, SPE Textbook Series, **11**, SPE, Richardson.

King, M.J. and Datta-Gupta, A., 1998: Streamline simulation: a current perspective. *In Situ* **22** (1), 91-140.

Luenberger, D.G., 1979: *Introduction to dynamic systems*, Wiley, New York.

Moler, C. and Van Loan, C.: Nineteen dubious ways to compute the exponential of a matrix. *SIAM Review* **20** (4), 801-836.

5 System analysis

5.1 Alternative system representations

5.1.1 Triples and quadruples

In the systems theory literature various forms are used to represent LTI systems. For an overview, see Skogestad and Postlewaite (2005). Until now we have mainly used the time-continuous state space form for the system and output equations as in expression (3.8) and (3.11):

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) \ , \tag{5.1}$$

$$\mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{u}(t) \ . \tag{5.2}$$

Equations (5.1) and (5.2) may be combined in a compact notation as

$$\begin{bmatrix} \dot{\mathbf{x}}(t) \\ \mathbf{y}(t) \end{bmatrix} = \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix} \begin{bmatrix} \mathbf{x}(t) \\ \mathbf{u}(t) \end{bmatrix} , \tag{5.3}$$

Representation (5.3) is known as a *quadruple*, or, in case $\mathbf{D} = \mathbf{0}$, as a *triple.* Just like equations (5.1) and (5.2), it is a non-unique representation of the system, because under a similarity transformation the elements of the quadruple obtain different numerical values but still represent the same system. E.g. using relationship (4.10), the quadruple transforms as

$$\begin{bmatrix} \dot{\mathbf{z}}(t) \\ \mathbf{y}(t) \end{bmatrix} = \begin{bmatrix} \tilde{\mathbf{A}} & \tilde{\mathbf{B}} \\ \tilde{\mathbf{C}} & \mathbf{D} \end{bmatrix} \begin{bmatrix} \mathbf{z}(t) \\ \mathbf{u}(t) \end{bmatrix} , \tag{5.4}$$

where

$$\tilde{\mathbf{A}} = \mathbf{M}^{-1}\mathbf{A}\mathbf{M} , \tilde{\mathbf{B}} = \mathbf{M}^{-1}\mathbf{B} , \tilde{\mathbf{C}} = \mathbf{C}\mathbf{M} \ , \tag{5.5}$$

while **D** remains unchanged.

5.1.2 Impulse response representation

An *impulse* is a sudden input. An impulse occurring at time $\breve{t}$ can be represented as

$$\delta(t - \breve{t}) \triangleq \begin{cases} \infty, & t = \breve{t} \\ 0, & t \neq \breve{t} \end{cases} , \tag{5.6}$$

under the condition that

$$\lim_{\Delta t \to 0} \int_{\breve{t}}^{\breve{t}+\Delta t} \delta(t - \breve{t}) dt = 1 \ , \tag{5.7}$$

where the symbol δ is known as the *Dirac delta function*. Condition (5.7) implies that although the amplitude of the impulse function at $t = \breve{t}$ is infinite, its integral value is finite with a value equal to one[†]. For example, consider a time interval $t_1 \le t < t_2$ during which a system experiences a single impulsive input $u\delta(\breve{t} - t)$, where u represents a flow rate with dimensions m^3/s. The instantaneous flow rate during the infinitesimally small time interval

[†] An impulse represented by a Dirac delta function is therefore sometimes referred to as a *unit impulse*.

$$\breve{t} = \lim_{\Delta t \to 0} \int_{\bar{t}}^{\bar{t}+\Delta t} dt \tag{5.8}$$

is then infinitely large, but the total flow entering the system over the period $t_1 \le t < t_2$ is just u m^3. The *impulse response* of a scalar dynamic system is now defined as

$$x_{\delta(\bar{t})}(t) \triangleq \int_{\bar{t}}^{t} e^{a(t-\tau)} b\delta(\tau - \breve{t})\, d\tau = e^{a(t-\bar{t})} b\,. \tag{5.9}$$

Similarly the impulse response of a vector dynamic system to a single input can be written as

$$\mathbf{x}_{\delta_i(\bar{t})}(t) \triangleq \int_{\bar{t}}^{t} e^{\mathbf{A}(t-\tau)} \mathbf{B}\mathbf{i}_i \delta(\tau - \breve{t})\, d\tau = e^{\mathbf{A}(t-\bar{t})} \mathbf{B}\mathbf{i}_i\,, \tag{5.10}$$

where $\mathbf{i}_i$ is the i^{th} *canonical unit vector*, i.e. a unit vector with just the i^{th} element equal to one:

$$\mathbf{i}_i = \begin{bmatrix} \vdots \\ i_{i-1} \\ i_i \\ i_{i+1} \\ \vdots \end{bmatrix} = \begin{bmatrix} \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \end{bmatrix}. \tag{5.11}$$

Without loss of generality we may take the time at which the impulse occurs as $\breve{t} = 0$, in which case expression (5.10) reduces to

$$\mathbf{x}_{\delta_i}(t) \triangleq e^{\mathbf{A}t} \mathbf{B}\mathbf{i}_i\,. \tag{5.12}$$

The *impulse response matrix* $\mathbf{G}_{xu}$ is now defined as the relationship between an impulsive input and the state:

$$\mathbf{G}_{xu}(t) \triangleq \begin{cases} \mathbf{0}, & t < 0 \\ e^{\mathbf{A}t}\mathbf{B}, & t \ge 0 \end{cases}, \tag{5.13}$$

such that the ij^{th} element of $\mathbf{G}_{xu}$ corresponds to the response $x_i(t)$ of a system with zero initial condition to an impulsive input $u_j(t) = \delta(t-0)$. The response to an arbitrary input $\mathbf{u}(t)$ that is zero for $t < 0$ can then be expressed as

$$\mathbf{x}(t) = \int_0^t \mathbf{G}_{xu}(t-\tau)\mathbf{u}(\tau)\, d\tau\,. \tag{5.14}$$

The right-hand side of equation (5.14), which is known as a *convolution integral*, may be interpreted as an operator that maps the input **u** to the state **x**. Similarly, we can define the impulse response matrix $\mathbf{G}_{yu}$,

$$\mathbf{G}_{yu}(t) \triangleq \begin{cases} \mathbf{0}, & t < 0 \\ \mathbf{C}e^{\mathbf{A}t}\mathbf{B} + \mathbf{D}\delta(t) & t \ge 0 \end{cases}, \tag{5.15}$$

which allows us to directly map the input **u** to the output **y** according to

$$\mathbf{y}(t) = \int_0^t \mathbf{G}_{yu}(t-\tau)\mathbf{u}(\tau)\, d\tau\ . \tag{5.16}$$

A system representation like the one in equation (5.16) is sometimes referred to as an *external description*, because it links the input **u** to the output **y** without considering the internal variables **x**. Obviously, a representation like the one in equation (5.3) is then called an *internal description*.

5.1.3 Markov parameters

Consider an LTI system with $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{u} \in \mathbb{R}^m$ and $\mathbf{y} \in \mathbb{R}^p$ where $m \le n$ and $p \le n$. Starting from equations (5.15) and (5.16) we can express the external system description as

$$\begin{aligned}
\mathbf{y}(t) &= \int_0^t \mathbf{G}_{yu}(t-\tau)\mathbf{u}(\tau)\,d\tau \\
&= \int_0^t \left[\mathbf{C}e^{\mathbf{A}(t-\tau)}\mathbf{B} + \mathbf{D}\delta(t-\tau)\right]\mathbf{u}(\tau)\,d\tau \\
&= \int_0^t \left[\mathbf{C}\left\{\mathbf{I} + \mathbf{A}(t-\tau) + \frac{\left[\mathbf{A}(t-\tau)\right]^2}{2!} + \ldots\right\}\mathbf{B} + \mathbf{D}\delta(t-\tau)\right]\mathbf{u}(\tau)\,d\tau, \\
&= \mathbf{D}\int_0^t \delta(t-\tau)\mathbf{u}(\tau)\,d\tau + \mathbf{CB}\int_0^t \mathbf{u}(\tau)\,d\tau + \mathbf{CAB}\int_0^t (t-\tau)\mathbf{u}(\tau)\,d\tau + \mathbf{CA}^2\mathbf{B}\int_0^t \frac{(t-\tau)^2}{2!}\mathbf{u}(\tau)\,d\tau + \ldots,
\end{aligned} \tag{5.17}$$

where we used a Taylor expansion for the matrix exponential just like in equation (4.18). All the integrals in the last term of equation (5.33), when evaluated, become *m*-dimensional input vectors[†], and we can therefore write $\mathbf{y}(t)$ as

$$\mathbf{y}(t) = \underbrace{\begin{bmatrix}\mathbf{D} & \mathbf{CB} & \mathbf{CAB} & \mathbf{CA}^2\mathbf{B} & \ldots & \mathbf{CA}^k\mathbf{B} & \ldots\end{bmatrix}}_{\mathcal{H}} \int_0^t \begin{bmatrix} \delta(t-\tau) \\ 1 \\ (t-\tau) \\ \frac{(t-\tau)^2}{2!} \\ \vdots \\ \frac{(t-\tau)^k}{k!} \\ \vdots \end{bmatrix} \mathbf{u}(\tau)\,d\tau. \tag{5.18}$$

The matrix $\mathcal{H}$ is known as the *Hankel matrix*, and its elements as the *Markov parameters* of the dynamical system.

to be continued

5.1.4 Transfer function representation*

A frequently used representation of LTI systems makes use of a *transfer function* defined as either the *Laplace transform* of the impulse response:

$$\mathbf{G}_{yu}(s) = \int_0^\infty \mathbf{G}_{yu}(t)e^{-st}dt\ , \tag{5.19}$$

[†] More precisely, the evaluated integrals are vector-valued functions of time.

where s is the Laplace variable, or, alternatively, by using the Laplace transforms of the state and output equations (5.1) and (5.2):

$$s\mathbf{x}(s) = \mathbf{A}\mathbf{x}(s) + \mathbf{B}\mathbf{u}(s) \ , \tag{5.20}$$

$$\mathbf{y}(s) = \mathbf{C}\mathbf{x}(s) + \mathbf{D}\mathbf{u}(s) \ , \tag{5.21}$$

leading to

$$\mathbf{x}(s) = (s\mathbf{I} + \mathbf{A})^{-1}\mathbf{B}\mathbf{u}(s) \ , \tag{5.22}$$

$$\mathbf{y}(s) = \underbrace{\left[\mathbf{C}(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B} + \mathbf{D}\right]}_{\mathbf{G}_{yu}(s)}\mathbf{u}(s) \ . \tag{5.23}$$

An advantage of the transfer function representation is that it is unique, unlike the state-space representation. Also, it allows for the description of a wider class of dynamic systems than the state space representation, such as for example systems containing time delays. Moreover, there exists a well-developed theory of Laplace transforms which enables a relatively simple analysis of complex LTI systems. Finally, it is possible to perform the analysis in the *frequency domain*, by simply replacing the Laplace variable s by the complex frequency $i\omega$. However, transfer function and frequency domain representations are not very well suited to analyze nonlinear systems, as opposed to state space representations. Because eventually we aim to analyze the behavior of multi-phase nonlinear reservoir systems we will mainly use state space representations in the remainder of this text.

*5.2 The state transition matrix**

5.2.1 Linear time-varying systems

Until now, the theory in this chapter was limited to LTI systems. In this section we will extend the analysis to linear time-varying (LTV) systems. They are represented in state-space notation by the equations

$$\dot{\mathbf{x}}(t) = \mathbf{A}(t)\mathbf{x}(t) + \mathbf{B}(t)\mathbf{u}(t) \ , \tag{5.24}$$

$$\mathbf{y}(t) = \mathbf{C}(t)\mathbf{x}(t) + \mathbf{D}(t)\mathbf{u}(t) \ , \tag{5.25}$$

which are identical to equations (3.7) and (3.11) except for the addition of the direct throughput matrix **D**. The general solution to differential equation (5.24) is no longer given by equation (4.34), although it is still possible to derive a result that has a similar structure, i.e. that is the sum of a transient response and a forced response:

$$\mathbf{x}(t) = \mathbf{\Phi}(t,\breve{t})\breve{\mathbf{x}} + \int_{\breve{t}}^{t} \mathbf{\Phi}(t,\tau)\mathbf{B}(\tau)\mathbf{u}(\tau)\,d\tau \ . \tag{5.26}$$

Here the matrix $\mathbf{\Phi}(t,\breve{t}) \in \mathbb{R}^{n\times n}$ is the *state transition matrix*, which, as the name implies, represents the transition from an initial state $\breve{\mathbf{x}}(\breve{t})$ to a state $\mathbf{x}(t)$. As before, we usually restrict the analysis to cases where $t > \breve{t}$ to honor the causality of the underlying physical process. The exponential $e^{\mathbf{A}(t-\breve{t})}$ in equation (4.34) is in fact a particular form of the state transition matrix, with the special property that it is invariant for a shift in time, i.e. it only depends on the time difference $t - \breve{t}$ and not on the specific values of t and $\breve{t}$.

5.2.2 Properties

For LTV systems it is in general not possible to derive a closed-form expression for the state transition matrix $\mathbf{\Phi}$, except for some simple cases. However, it is possible to derive a large number of properties of $\mathbf{\Phi}$ that are of importance for the analysis of dynamics of time-varying systems. Here we will mention just a few of them; for a wider overview and detailed proofs consult the texts mentioned in Section 1.1. First of all, for $\breve{t} = t$ we find that

$$\mathbf{x}(t) = \mathbf{\Phi}(t,t)\mathbf{x}(t) \ , \tag{5.27}$$

which implies that

$$\mathbf{\Phi}(t,t) = \mathbf{I} \ . \tag{5.28}$$

Furthermore, if we consider the states of a system at three moments in time, t_1, t_2 and t_3, we can write

$$\begin{aligned} \mathbf{x}(t_3) &= \mathbf{\Phi}(t_3,t_2)\mathbf{x}(t_2) \\ &= \mathbf{\Phi}(t_3,t_2)\mathbf{\Phi}(t_2,t_1)\mathbf{x}(t_1) , \end{aligned} \tag{5.29}$$

which implies that

$$\mathbf{\Phi}(t_3,t_1) = \mathbf{\Phi}(t_3,t_2)\mathbf{\Phi}(t_2,t_1) \ . \tag{5.30}$$

This implies that it makes no difference whether we go from state $\mathbf{x}(t_1)$ to state $\mathbf{x}(t_3)$ directly or via an intermediate state. This property follows from the fact that linear differential equations have unique solutions, as is proved in any textbook on differential equations; see e.g. Boyce and Di Prima (2005). Using this property and equation (5.28) we can now write

$$\mathbf{\Phi}(t,t) = \mathbf{\Phi}(t,\breve{t})\mathbf{\Phi}(\breve{t},t) = \mathbf{I} \ , \tag{5.31}$$

which implies that $\mathbf{\Phi}(t,\breve{t})$ is non-singular for all t and $\breve{t}$, and its inverse is given by[†]

$$\mathbf{\Phi}^{-1}(t,\breve{t}) = \mathbf{\Phi}(\breve{t},t) \ . \tag{5.32}$$

5.3 Controllability and observability – continuous systems

5.3.1 Controllability

Controllability matrix

We start with the following definition:

An LTI system $\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t)$ *is controllable if it is possible to drive it from a zero initial state* $\mathbf{x}(0) = \mathbf{0}$ *to a nonzero state* $\mathbf{x}(\breve{t}) = \breve{\mathbf{x}}$*, using an input* $\mathbf{u}(t)$*, within a finite time period* $0 \le t \le \breve{t}$.[‡]

[†] Here we made use of the fact that mathematically there is no objection against integration 'backwards' in time.

[‡] In many texts the necessary and sufficient conditions to reach a state $\mathbf{x}(\breve{t}) = \breve{\mathbf{x}}$ given the initial state $\mathbf{x}(0) = \mathbf{0}$ are referred to as *reachability* conditions, and the associated matrix as the *reachability matrix*. The conditions for the reverse situation, i.e. to reach a state $\mathbf{x}(\breve{t}) = \mathbf{0}$ given the initial state $\mathbf{x}(0) = \hat{\mathbf{x}}$ are then referred to as the controllability conditions. For nonlinear systems and for discrete-time linear systems there can be a difference between reachability and controllability, but for continuous-time linear systems, they are identical; see e.g. Kailath (1980) or Antsaklis and Michel (2006).

To determine the conditions that govern controllability, consider an LTI system with $\mathbf{x} \in \mathbb{R}^n$ and $\mathbf{u} \in \mathbb{R}^m$ where $m \le n$. If we start from a given initial state $\mathbf{x}(0) = \mathbf{0}$ and attempt to reach a state $\mathbf{x}(\breve{t}) = \breve{\mathbf{x}}$ through prescribing the components of the input vector $\mathbf{u}(t), 0 \le t \le \breve{t}$, we can express this, using equation (5.14), as

$$\begin{aligned}
\breve{\mathbf{x}} &= \int_0^{\breve{t}} \mathbf{G}_{xu}(\breve{t}-\tau)\mathbf{u}(\tau)\,d\tau \\
&= \int_0^{\breve{t}} e^{\mathbf{A}(\breve{t}-\tau)}\mathbf{B}\mathbf{u}(\tau)\,d\tau \\
&= \int_0^{\breve{t}} \left\{ \mathbf{I} + \mathbf{A}(\breve{t}-\tau) + \frac{\left[\mathbf{A}(\breve{t}-\tau)\right]^2}{2!} + \ldots \right\} \mathbf{B}\mathbf{u}(\tau)\,d\tau, \\
&= \mathbf{B}\int_0^{\breve{t}} \mathbf{u}(\tau)\,d\tau + \mathbf{A}\mathbf{B}\int_0^{\breve{t}} (\breve{t}-\tau)\mathbf{u}(\tau)\,d\tau + \mathbf{A}^2\mathbf{B}\int_0^{\breve{t}} \frac{(\breve{t}-\tau)^2}{2!}\mathbf{u}(\tau)\,d\tau + \ldots .
\end{aligned} \tag{5.33}$$

where we used again a Taylor expansion for the matrix exponential just like in equation (4.18). All the integrals in the last term of equation (5.33), when evaluated, become *m*-dimensional input vectors, and we can therefore write the desired state $\breve{\mathbf{x}}$ as a linear combination of columns of the matrices $\mathbf{B}, \mathbf{AB}, \mathbf{A}^2\mathbf{B}, \ldots$ as determined by these vectors. If we consider only the first *n* terms of this expansion, this leads to

$$\breve{\mathbf{x}} = \begin{bmatrix} \mathbf{B} & \mathbf{AB} & \mathbf{A}^2\mathbf{B} & \ldots & \mathbf{A}^{n-1}\mathbf{B} \end{bmatrix} \begin{bmatrix} \breve{\mathbf{u}}_n \\ \breve{\mathbf{u}}_{n-1} \\ \breve{\mathbf{u}}_{n-2} \\ \vdots \\ \breve{\mathbf{u}}_1 \end{bmatrix}. \tag{5.34}$$

where we have numbered the input vectors $\breve{\mathbf{u}}_n, \breve{\mathbf{u}}_{n-1}, \cdots, \breve{\mathbf{u}}_1$ in reversed order to reflect that those with the lowest indices correspond to the oldest inputs. The matrix

$$\mathcal{C} \triangleq \begin{bmatrix} \mathbf{B} & \mathbf{AB} & \mathbf{A}^2\mathbf{B} & \ldots & \mathbf{A}^{n-1}\mathbf{B} \end{bmatrix} \in \mathbb{R}^{n \times nm} \tag{5.35}$$

is known as the *controllability matrix*. For the special case that the input vector $\mathbf{u}$ and therefore also the vectors $\breve{\mathbf{u}}_n, \breve{\mathbf{u}}_{n-1}, \cdots, \breve{\mathbf{u}}_1$ are scalars, i.e. for a single-input system, the $n \times m$ matrix $\mathbf{B}$ becomes an $n \times 1$ vector and the $n \times nm$ matrix $\mathcal{C}$ becomes an $n \times n$ matrix. In that case, we can find a unique set of input values $\breve{u}_n, \ldots, \breve{u}_1$ that produce the desired state $\breve{\mathbf{x}} \in \mathbb{R}^n$ if the square matrix $\mathcal{C}$ is invertible, i.e. if it has rank *n*. In the multiple-input case we have $nm > n$, and therefore there may exist an infinitely large number of control sequences that lead to the same desired state. However also for a multiple-input system the rank of $\mathcal{C}$ will have to be equal to *n*, because otherwise it will not be possible to find a linearly independent set of vectors $\breve{\mathbf{u}}_n, \breve{\mathbf{u}}_{n-1}, \cdots, \breve{\mathbf{u}}_1$ that produce the desired state $\breve{\mathbf{x}} \in \mathbb{R}^n$. Note that the rank of $\mathcal{C}$ can never be higher than its number of rows, which is *n*. Therefore a necessary and sufficient condition for controllability is that $\mathcal{C}$ has rank *n*. Another way of stating the same condition is by requiring that $\breve{\mathbf{x}} \in \mathrm{col}(\mathcal{C})$, i.e. that $\breve{\mathbf{x}}$ is in the column space of $\mathcal{C}$; see also Section B.1.2 of Appendix B.

Controllability Gramian

An alternative way to express the controllability condition is through the use of a matrix known as a *Gramian matrix*, or, in short, a *Gramian*, which is a symmetric matrix defined as

$$\mathbf{W}(t) \triangleq \int_0^t \mathbf{G}(\tau)\mathbf{G}^T(\tau)d\tau \ , \tag{5.36}$$

for a certain matrix $\mathbf{G}(t)$. More specifically, we consider the *controllability Gramian* $\mathbf{W}_c \in \mathbb{R}^{n\times n}$, defined as

$$\mathbf{W}_c \triangleq \int_0^{\breve{t}} \mathbf{G}_{xu}(\tau)\mathbf{G}_{xu}^T(\tau)d\tau = \int_0^{\breve{t}} e^{\mathbf{A}\tau}\mathbf{B}\mathbf{B}^T e^{\mathbf{A}^T\tau}d\tau \ , \tag{5.37}$$

which, since the upper limit in the integral is finite, is also known as the *finite-time controllability Gramian*, as opposed to the *infinite time controllability Gramian*, which is, indeed, obtained by changing the upper limit to ∞. The latter is only defined for systems that are asymptotically stable. The necessary and sufficient condition for controllability is now that the controllability Gramian is non-singular, or, equivalently, that it has full rank. To demonstrate the proof, which goes back to the work of Kalman et al. (1963), it is convenient to rewrite the controllability Gramian as

$$\mathbf{W}_c \triangleq \int_0^{\breve{t}} \mathbf{G}_{xu}(\breve{t}-\tau)\mathbf{G}_{xu}^T(\breve{t}-\tau)d\tau = \int_0^{\breve{t}} e^{\mathbf{A}(\breve{t}-\tau)}\mathbf{B}\mathbf{B}^T e^{\mathbf{A}^T(\breve{t}-\tau)}d\tau \ , \tag{5.38}$$

which is numerically identical to equation (5.37) but in its form more directly related to the input vector **u** as represented in equation (5.33). First we consider the sufficient condition, which states that a non-singular Gramian implies the ability to transfer a state $\mathbf{x}(0)=\mathbf{0}$ to $\mathbf{x}(\breve{t})=\breve{\mathbf{x}}$, and of which the proof is as follows: Assume that $\mathbf{W}_c$ is non-singular such that its inverse exists. Next consider an input vector

$$\mathbf{u}(t) = \mathbf{B}^T e^{\mathbf{A}^T(\breve{t}-\tau)}\mathbf{W}_c^{-1}\breve{\mathbf{x}} \ . \tag{5.39}$$

It can now be verified that this input vector indeed forces the state from $\mathbf{x}(0)=\mathbf{0}$ to $\mathbf{x}(\breve{t})=\breve{\mathbf{x}},$ by substitution of equation (5.39) in equation (5.14) which leads to

$$\begin{aligned}
\breve{\mathbf{x}} &= \int_0^{\breve{t}} \mathbf{G}_{xu}(\breve{t}-\tau)\mathbf{u}(\tau)d\tau \\
&= \int_0^{\breve{t}} e^{\mathbf{A}(\breve{t}-\tau)}\mathbf{B}\mathbf{u}(\tau)d\tau \\
&= \int_0^{\breve{t}} e^{\mathbf{A}(\breve{t}-\tau)}\mathbf{B}\mathbf{B}^T e^{\mathbf{A}^T(\breve{t}-\tau)}\mathbf{W}_c^{-1}\breve{\mathbf{x}}\,d\tau \ . \\
&= \mathbf{W}_c\mathbf{W}_c^{-1}\breve{\mathbf{x}} \\
&= \breve{\mathbf{x}}.
\end{aligned} \tag{5.40}$$

Next we consider the necessary condition, which states that a non-singular Gramian is required to guarantee controllability, and of which the proof consists of 2 steps: 1) Consider some time $\breve{t}$ for which $\mathbf{W}_c$ is singular. In that case there must exist some non-zero vector **v** such that

$$\mathbf{v}^T\mathbf{W}_c\mathbf{v} = \int_0^{\breve{t}} \mathbf{v}^T e^{\mathbf{A}(\breve{t}-\tau)}\mathbf{B}\mathbf{B}^T e^{\mathbf{A}^T(\breve{t}-\tau)}\mathbf{v}\,d\tau = 0\,. \tag{5.41}$$

The integrand in equation (5.41) can be written as

$$\mathbf{v}^T e^{\mathbf{A}(\breve{t}-\tau)}\mathbf{B}\mathbf{B}^T e^{\mathbf{A}^T(\breve{t}-\tau)}\mathbf{v} = \tilde{\mathbf{v}}^T\tilde{\mathbf{v}}\ , \tag{5.42}$$

which means that it is always nonnegative. For the integral in (5.41) to become zero it is therefore necessary that the integrand is equal to zero itself at the entire interval $0 < t < \breve{t}$, i.e. for a singular Gramian we have a non-zero vector **v** such that

$$\mathbf{v}^T e^{\mathbf{A}(\breve{t}-\tau)}\mathbf{B} = \tilde{\mathbf{v}}^T = \mathbf{0} \text{ for } 0 < \tau < \breve{t}\ . \tag{5.43}$$

2) Next we need to prove that in this case there are uncontrollable states. In particular we will prove that it is impossible to force the state in the direction of **v**, i.e. to force the state from $\mathbf{x}(0) = \mathbf{0}$ to $\mathbf{x}(\breve{t}) = \breve{\mathbf{x}} = \alpha\mathbf{v}$, where α is an arbitrary non-zero scalar. Suppose that it would be possible to do so, then there should be an input **u** such that we could write, using equation (5.14),

$$\mathbf{x}(\breve{t}) = \alpha\mathbf{v} = \int_0^{\breve{t}} e^{\mathbf{A}(\breve{t}-\tau)}\mathbf{B}\mathbf{u}(\tau)\mathbf{u}(\tau)\,d\tau\ , \tag{5.44}$$

and therefore also

$$\mathbf{v}^T\alpha\mathbf{v} = \int_0^{\breve{t}} \mathbf{v}^T e^{\mathbf{A}(\breve{t}-\tau)}\mathbf{B}\mathbf{u}(\tau)\mathbf{u}(\tau)\,d\tau\ . \tag{5.45}$$

The left-hand side of equation is clearly unequal to zero because both α and **v** are unequal to zero. However, the right-hand side must be zero according to equation (5.43), which is a contradiction. This contradiction completes the proof, because it demonstrates that there are uncontrollable states if $\mathbf{W}_c$ is singular in the entire interval $0 < t < \breve{t}$.

Lyapunov equations

Because reservoir systems are nearly always stable, we may, in theory, assess controllability of a linear reservoir system through verifying the rank of the infinite time controllability Gramian. For small systems this can efficiently be done through solving the following matrix differential equation known as a *Lyapunov equation*:

$$\mathbf{A}\mathbf{W}_c + \mathbf{W}_c\mathbf{A}^T = \mathbf{B}\mathbf{B}^T\ . \tag{5.46}$$

It can be verified that $\mathbf{W}_c$ is indeed a solution of equation (5.46) by observing that

$$\mathbf{A}\mathbf{W}_c + \mathbf{W}_c\mathbf{A}^T = \int_0^{\infty} \frac{d}{d\tau}\left(e^{\mathbf{A}\tau}\mathbf{B}\mathbf{B}^T e^{\mathbf{A}^T\tau}\right)d\tau = e^{\mathbf{A}\tau}\mathbf{B}\mathbf{B}^T e^{\mathbf{A}^T\tau}\Big|_0^{\infty} = \mathbf{B}\mathbf{B}^T\ , \tag{5.47}$$

where we have used the property that the system is stable, i.e. that $e^{\mathbf{A}\tau}$ approaches zero for τ approaching infinity. For an overview of numerical techniques to compute the solutions of Lyapunov equations see e.g. Antoulas (2005). Unfortunately, solving Lyapunov equations for realistic reservoir models is computationally infeasible, and we will have to use approximate methods to assess the controllability of large systems, as will be discussed in more detail in later chapters.

5.3.2 Duality

The *dual* system of the linear system (5.3) is defined as the linear system

$$\begin{bmatrix} \dot{\mathbf{x}}'(t) \\ \mathbf{u}(t) \end{bmatrix} = \begin{bmatrix} -\mathbf{A}^T & -\mathbf{C}^T \\ \mathbf{B}^T & \mathbf{D}^T \end{bmatrix} \begin{bmatrix} \mathbf{x}'(t) \\ \mathbf{y}(t) \end{bmatrix}, \tag{5.48}$$

where we use a prime to indicate that the states $\mathbf{x}'$ correspond to the dual system. In the dual system the role of inputs and outputs have been reversed, and therefore its response can be interpreted as the response of the original system, but with time running backwards, and with causality reversed. In particular, consider a nonhomogeneous original system with input $\mathbf{u}(t)$ and output $\mathbf{y}(t)$ over time interval $t_1 \le t \le t_2$, such that according to equation (4.33) we have

$$\mathbf{y}(t) = \mathbf{C}\int_{t_1}^{t} e^{\mathbf{A}(t-\tau)}\mathbf{B}\mathbf{u}(\tau)\,d\tau + \mathbf{D}\mathbf{u}(t). \tag{5.49}$$

The response of the dual system can then be expressed as

$$\mathbf{u}(t) = -\mathbf{B}^T\int_{t_2}^{t} e^{-\mathbf{A}(t-\tau)}\mathbf{C}^T\mathbf{y}(\tau)\,d\tau + \mathbf{D}^T\mathbf{y}(t). \tag{5.50}$$

5.3.3 Observability

Observability matrix

We start with the following definition:

An LTI system $\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t)$, $\mathbf{y}(t) = \mathbf{C}\mathbf{x}(t)$, *is observable if it is possible to determine a non-zero initial state* $\mathbf{x}(0) = \widehat{\mathbf{x}}$ *from an output* $\mathbf{y}(t)$, *within a finite time period* $0 \le t \le \breve{t}$.[†]

Using the concept of duality as introduced in Section 5.3.2, this definition may be rephrased as

An LTI system $\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t)$, $\mathbf{y}(t) = \mathbf{C}\mathbf{x}(t)$ *is observable if it is possible to drive its dual system from a zero final state* $\mathbf{x}'(\breve{t}) = \mathbf{0}$ *to a non-zero state* $\mathbf{x}'(0) = \widehat{\mathbf{x}}'$, *using an output* $\mathbf{y}(t)$, *within a finite time period* $0 \le t \le \breve{t}$.

To determine the conditions that govern observability, we can now apply all the arguments that were used to derive the conditions for controllability, but then applied to the dual system. In particular, the matrix

$$\mathscr{O} \triangleq \begin{bmatrix} \mathbf{C}^T & \mathbf{A}^T\mathbf{C}^T & \left(\mathbf{A}^2\right)^T\mathbf{C}^T & \dots & \left(\mathbf{A}^{n-1}\right)^T\mathbf{C}^T \end{bmatrix}^T \in \mathbb{R}^{np\times n} \tag{5.51}$$

is known as the *observability matrix*. A necessary and sufficient condition for observability is that $\mathscr{O}$ has rank *n*. Similarly, the finite-time *observability Gramian* is defined as

$$\mathbf{W}_o \triangleq \int_{\breve{t}}^{0} \mathbf{G}_{xy}(\tau)\mathbf{G}_{xy}^T(\tau)\,d\tau = \int_{\breve{t}}^{0} e^{-\mathbf{A}^T\tau}\mathbf{C}^T\mathbf{C}e^{-\mathbf{A}\tau}\,d\tau = \int_{0}^{\breve{t}} e^{\mathbf{A}^T\tau}\mathbf{C}^T\mathbf{C}e^{\mathbf{A}\tau}\,d\tau\ , \tag{5.52}$$

[†] In many texts term observability is restricted to the necessary and sufficient conditions to determine a state $\mathbf{x}(\breve{t})$ of a homogeneous system given the outputs $\mathbf{y}(t), t > \breve{t}$. The conditions to determine a state $\mathbf{x}(\breve{t})$ of a homogeneous system given the outputs $\mathbf{y}(t), t < \breve{t}$, i.e. from earlier measurements, are then referred to as the *constructability* conditions. For nonlinear systems and for discrete-time linear systems there may be a difference between constructability and observability, but for continuous-time linear systems, they are identical; see e.g. Kailath (1980) or Antsaklis and Michel (2006).

where the impulse response function $\mathbf{G}_{xy}$ is the dual of $\mathbf{G}_{xu}$. The necessary and sufficient condition for observability is now that the observability Gramian is non-singular, or, equivalently, that it has full rank. The corresponding Lyapunov equation is given by

$$\mathbf{A}^T\mathbf{W}_o + \mathbf{W}_o\mathbf{A} = \mathbf{C}^T\mathbf{C} \ , \tag{5.53}$$

which may be used to compute the observability Gramian for not-too-large systems.

5.3.4 Notes

- In some situations it may happen that not all individual states of a system are controllable and/or observable. In that case there may be a *controllable subspace* and/or an *observable subspace* consisting of those states that are controllable and/or observable respectively, as will be discussed in more detail in the following section. The terms *system controllability* and *system observability* refer to the situation where all states of a system can be controlled and/or observed respectively.
- If we bring the system in diagonal form (4.36), it follows that system controllability also implies that all the modes can be controlled. Conversely, if we find that one of the modes is not connected to an input, i.e. that one of the rows of matrix $\mathbf{M}^{-1}\mathbf{B}$ is consisting of zeros only, the system is not controllable. Similarly, if we find that one of the columns of the matrix $\mathbf{CM}$ is consisting entirely of zeros, the system is not observable.
- The concepts of controllability and observability as discussed above are sometimes referred to as *state controllability* and *state observability* as opposed to other, more pragmatic, concepts known as *input-output controllability/observability* see Skogestad and Postlethwaite (2005). We will not address these alternative definitions but we mention two limitations of the state controllability/observability concepts: 1) They are restricted to controllability/observability at instantaneous moments in time, and do not consider the conditions required to control or observe a state for a longer period. 2) They do not take into account any bounds on the inputs or the outputs. This implies that although a state may, in theory, be controllable, this might require very large inputs outside of what is practically achievable. Similarly a state may be, in theory, observable, but only with an extremely sensitive sensor and a very low noise level. Verifying the rank of the controllability/observability matrix, or, equivalently, the regularity of the controllability/observability Gramian, thus provides a theoretical, qualitative (yes/no) answer rather than a practical, quantitative answer about the controllability/observability of a system.
- In the next section we will discuss an extension to the concepts of controllability and observability. In particular it will be shown that through the use of a singular value decomposition (SVD) of the controllability/observability matrix it is possible to determine a quantitative measure of controllability/observability. This is of major relevance to reservoir simulation models, where we often find that all states are controllable/observable qualitatively, i.e. under the assumption of unbounded inputs/outputs, whereas in practice the limited available input energy and the limited sensor sensitivity restrict the controllability and observability to narrow regions around the wells.
- The theory of controllability and observability has been developed in the measurement and control community, focusing on systems with a small number of state variables, say below 10^2. For larger systems, as e.g. our numerical reservoir models, computation of the

controllability and observability matrices becomes a numerically ill-conditioned problem. Also computation of the Lyapunov equations to obtain the controllability and observability Gramians is not feasible for realistic reservoir models which may have up to millions of state variables. In Chapter xxx we will address these issues again and present techniques that allow for the computation of approximate controllability/ observability matrices and Gramians for very large systems.

5.3.5 Observable and controllable subspaces

It was a discussed above that system controllability, i.e. controllability of all states, requires that the controllability matrix $\mathcal{C}$ has full rank. To be continued.

5.3.6 Linear time-varying systems*

The concept of controllability and observability Gramians may be extended to linear time-varying (LTV) systems. For controllability the definition becomes

$$\mathbf{W}_c(\breve{t},t) \triangleq \int_t^{\bar{t}} \mathbf{G}_{xu}(\breve{t},\tau)\mathbf{G}_{xu}^T(\breve{t},\tau)d\tau = \int_t^{\bar{t}} \mathbf{\Phi}(\breve{t},\tau)\mathbf{B}(\tau)\mathbf{B}^T(\tau)\mathbf{\Phi}(\breve{t},\tau)d\tau \ . \qquad (5.54)$$

Note that in this case the lower bound of the integral cannot be taken arbitrarily as zero, because the time-variance of the system implies that the results depend on the choices for t and $\breve{t}$. This leads to the use of the term *complete controllability*[†], for systems that can be forced from any state at a certain finite time t to any other state at another finite time $\breve{t}$. The proof that regularity of the controllability Gramian is a necessary and sufficient requirement for controllability of an LTV system is very similar to the proof for the time-invariant case; see e.g. Friedland (1986). Using the duality principle, the LTV version of the observability Gramian follows as

$$\mathbf{W}_o(\breve{t},t) \triangleq \int_{\breve{t}}^{t} \mathbf{G}_{xy}(\tau,\breve{t})\mathbf{G}_{xy}^T(\tau,\breve{t})d\tau = \int_t^{\bar{t}} \mathbf{\Phi}(\breve{t},\tau)\mathbf{C}(\tau)\mathbf{C}^T(\tau)\mathbf{\Phi}(\breve{t},\tau)d\tau \ , \qquad (5.55)$$

with a corresponding self-explanatory definition of *complete observability*.

5.3.7 Identifiability

In the previous sub-sections we addressed to what extent it is possible to control and observe the *states* of a system. In addition, we may be interested to obtain information about the *parameters* of the system[†]. Generally it will not be possible to influence the parameters of a system, so at first sight the concept of *parameter identifiability* is more closely related to observability than to controllability. However, as will be shown later on, in order to identify a parameter from an output, we often need to actively disturb the input, which implies that controllability has to play a role as well. Parameter estimation will be addressed in quite some detail in Chapter 8, where we will also return to the concept of identifiability.

[†] Sometimes, the term "complete controllability" is used to indicate what we called "system controllability", i.e. the property that all states are controllable.

[†] Recall that the states are the dependent variables in the governing differential equations, whereas the parameters appear in the coefficients of the equations. E.g. for two-phase reservoir flow considered in the examples, the state variables are the pressures and saturations, while the parameters are the permeabilities, porosities, relative permeabilities, fluid properties, etc.

5.4 Controllability and observability – discrete systems

In Chapter 4 we considered various system-theoretical concepts for continuous-time linear systems. Nearly all these concepts may also be defined for discrete-time linear systems, and we will discuss several of them, and some new ones, in this chapter. For a systematic overview of the analogy between continuous-time and discrete-time system aspects see e.g. Luenberger (1979).

5.4.1 Controllability

Consider the discrete-time LTI formulation (4.56)

$$\mathbf{x}_k = \mathbf{A}\mathbf{x}_{k-1} + \mathbf{B}\mathbf{u}_{k-1}\,. \tag{5.56}$$

where $\mathbf{x}_k \in \mathbb{R}^n$ and $\mathbf{u}_k \in \mathbb{R}^m$, and where we have dropped the subscripts *d*. Starting from initial condition

$$\mathbf{x}_0 = \breve{\mathbf{x}}_0\,, \tag{5.57}$$

at discrete time $k = 0$, we can apply equation (5.56) recursively to obtain the state vectors at later times as

$$\mathbf{x}_1 = \mathbf{A}\mathbf{x}_0 + \mathbf{B}\mathbf{u}_1\,. \tag{5.58}$$

$$\mathbf{x}_2 = \mathbf{A}\mathbf{x}_1 + \mathbf{B}\mathbf{u}_2 = \mathbf{A}^2\mathbf{x}_0 + \mathbf{A}\mathbf{B}\mathbf{u}_1 + \mathbf{B}\mathbf{u}_2\,. \tag{5.59}$$

$$\mathbf{x}_3 = \mathbf{A}\mathbf{x}_2 + \mathbf{B}\mathbf{u}_3 = \mathbf{A}^3\mathbf{x}_0 + \mathbf{A}^2\mathbf{B}\mathbf{u}_1 + \mathbf{A}\mathbf{B}\mathbf{u}_2 + \mathbf{B}\mathbf{u}_3\,. \tag{5.60}$$

and so on, which can be written in matrix form as

$$\begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \mathbf{x}_3 \\ \vdots \end{bmatrix} = \begin{bmatrix} \mathbf{A} \\ \mathbf{A}^2 \\ \mathbf{A}^3 \\ \vdots \end{bmatrix} \mathbf{x}_0 + \begin{bmatrix} \mathbf{B} & \mathbf{0} & \mathbf{0} & \cdots \\ \mathbf{A}\mathbf{B} & \mathbf{B} & \mathbf{0} & \cdots \\ \mathbf{A}^2\mathbf{B} & \mathbf{A}\mathbf{B} & \mathbf{B} & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix} \begin{bmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \\ \mathbf{u}_3 \\ \vdots \end{bmatrix}. \tag{5.61}$$

If we consider the n^{th} term of this matrix equation, where *n* is equal to the number of states, i.e. to the dimension of $\mathbf{x}_k$, and take the initial condition $\mathbf{x}_0 = \mathbf{0}$, we find

$$\mathbf{x}_n = \begin{bmatrix} \mathbf{A}^{n-1}\mathbf{B} & \cdots & \mathbf{A}^2\mathbf{B} & \mathbf{A}\mathbf{B} & \mathbf{B} \end{bmatrix} \begin{bmatrix} \mathbf{u}_1 \\ \vdots \\ \mathbf{u}_{n-2} \\ \mathbf{u}_{n-1} \\ \mathbf{u}_n \end{bmatrix}. \tag{5.62}$$

This may also be written as

$$\mathbf{x}_n = \underbrace{\begin{bmatrix} \mathbf{B} & \mathbf{A}\mathbf{B} & \mathbf{A}^2\mathbf{B} & \ldots & \mathbf{A}^{n-1}\mathbf{B} \end{bmatrix}}_{\mathcal{C}} \begin{bmatrix} \mathbf{u}_n \\ \mathbf{u}_{n-1} \\ \mathbf{u}_{n-2} \\ \vdots \\ \mathbf{u}_1 \end{bmatrix}, \tag{5.63}$$

which is nearly identical to continuous-time expression (5.34). Just as in the continuous-time case we can now define the controllability matrix $\mathcal{C}$ as given in equation (5.51) and prove that the system is controllable if $\mathcal{C}$ has rank n as was discussed in Section 5.3.1.[†]

5.4.2 Observability

Similarly we may consider the discrete-time LTI output equation (4.57)

$$\mathbf{y}_k = \mathbf{C}\mathbf{x}_k + \mathbf{D}\mathbf{u}_k, \tag{5.64}$$

and apply it recursively to obtain

$$\mathbf{y}_1 = \mathbf{C}\mathbf{x}_1 + \mathbf{D}\mathbf{u}_1 = \mathbf{C}\mathbf{A}\mathbf{x}_0 + \left(\mathbf{C}\mathbf{B} + \mathbf{D}\right)\mathbf{u}_1, \tag{5.65}$$

$$\mathbf{y}_2 = \mathbf{C}\mathbf{x}_2 + \mathbf{D}\mathbf{u}_2 = \mathbf{C}\mathbf{A}^2\mathbf{x}_0 + \mathbf{C}\mathbf{A}\mathbf{B}\mathbf{u}_1 + \left(\mathbf{C}\mathbf{B} + \mathbf{D}\right)\mathbf{u}_2, \tag{5.66}$$

$$\mathbf{y}_3 = \mathbf{C}\mathbf{x}_3 + \mathbf{D}\mathbf{u}_3 = \mathbf{C}\mathbf{A}^3\mathbf{x}_0 + \mathbf{C}\mathbf{A}^2\mathbf{B}\mathbf{u}_1 + \mathbf{C}\mathbf{A}\mathbf{B}\mathbf{u}_2 + \left(\mathbf{C}\mathbf{B} + \mathbf{D}\right)\mathbf{u}_3, \tag{5.67}$$

and so on, which can also be written in matrix form as

$$\begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \\ \mathbf{y}_3 \\ \vdots \end{bmatrix} = \begin{bmatrix} \mathbf{C}\mathbf{A} \\ \mathbf{C}\mathbf{A}^2 \\ \mathbf{C}\mathbf{A}^3 \\ \vdots \end{bmatrix} \mathbf{x}_0 + \begin{bmatrix} \mathbf{C}\mathbf{B}+\mathbf{D} & \mathbf{0} & \mathbf{0} & \cdots \\ \mathbf{C}\mathbf{A}\mathbf{B} & \mathbf{C}\mathbf{B}+\mathbf{D} & \mathbf{0} & \cdots \\ \mathbf{C}\mathbf{A}^2\mathbf{B} & \mathbf{C}\mathbf{A}\mathbf{B} & \mathbf{C}\mathbf{B}+\mathbf{D} & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix} \begin{bmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \\ \mathbf{u}_3 \\ \vdots \end{bmatrix}. \tag{5.68}$$

If we consider the first n terms of this matrix equation, and take the inputs $\mathbf{u}_1, \mathbf{u}_2, \cdots, \mathbf{u}_n$ equal to zero, we find

$$\begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \\ \mathbf{y}_3 \\ \vdots \\ \mathbf{y}_n \end{bmatrix} = \underbrace{\begin{bmatrix} \mathbf{C} \\ \mathbf{C}\mathbf{A} \\ \mathbf{C}\mathbf{A}^2 \\ \vdots \\ \mathbf{C}\mathbf{A}^{n-1} \end{bmatrix}}_{\mathcal{O}} \mathbf{A}\mathbf{x}_0. \tag{5.69}$$

which leads to the observability matrix $\mathcal{O}$ as defined for the continuous case in equation (5.51)[‡]. As before, observability requires that the matrix $\mathcal{O}$ has rank n. Note that to derive the observability matrix we considered a system with zero initial conditions and finite inputs, whereas to derive the controllability matrix we considered a finite initial condition and no inputs.

5.4.3 Duality

To be completed.

[†] See the footnote at page 112 where it was explained that this matrix is often referred to as the reachability matrix, but that in the continuous-time case there is no difference between observability and reachability. Also in the discrete-time case there is no difference, as long as **A** is non-singular; see e.g. Kailath (1980) or Antsaklis and Michel (2006).

[‡] See the footnote at page 115 where it was explained that sometimes a distinction is made between observability and constructability, but that in the continuous-time case there is no difference between these concepts. Also in the discrete-time case there is no difference, as long as **A** is non-singular; see e.g. Kailath (1980) or Antsaklis and Michel (2006).

5.5 Model reduction

To be completed.

5.6 References for Chapter 5

Antsaklis, P.J. and Michel, A.N., 2006: *Linear systems*, Birkhäuser, Boston.

Antoulas, A.C., 2005: *Approximation of large-scale dynamical systems*, SIAM, Philadelphia.

Boyce, W. and Di Prima, R.C., 2005: *Elementary differential equations and boundary value problems, 8th ed.*, Wiley, New York.

Kailath, T., 1980: *Linear systems*, Prentice-Hall, Englewood Cliffs.

Kalman, R.E., Ho, Y.C. and Narendra, K.S., 1963: Controllability of linear dynamic systems. *Contributions to Differential Equations* **1** (2) 189-213.

Luenberger, D.G., 1979: *Introduction to dynamic systems*, Wiley, New York.

Skogestad, S. and Postlewaite, I., 2005: *Multivariable feedback control 2nd ed.*, Wiley, Chichester.

6 Optimization theory

6.1 Introduction

In Chapter 7 we will discuss techniques for reservoir *flooding optimization*, or *life-cycle optimization*, i.e. for the optimization of the recovery factor of a reservoir, or another economic objective, through manipulation of input variables such as water injection rates or bottom hole pressures over the producing life of the reservoir. Thereafter, in Chapter 8, we will discuss techniques for *data assimilation*, or *computer-assisted history matching*, i.e. for the updating of reservoir model parameters through minimizing the difference between model-predicted and actual measurements of production variables. As a precursor we will review some aspects of optimization theory in this chapter. Many numerical techniques are available to solve optimization problems. An important distinction is between methods that attempt to find a global optimum, and those that can find a local optimum only. For realistic problems, all optimization techniques involve some form of iteration, and the 'local methods' will produce answers that are dependent on the initial guess used as starting point for the iteration. Another distinction is between *gradient-based* and *gradient-free* methods. Gradient-based methods make use of gradients, i.e. of derivatives of the optimization objective with respect to the input variables, to guide the iteration process. Gradients of a function have the property that they point in the direction of maximum increase of the function value, which explains their significance to find the maximum (or the minimum) of a function.[‡] A disadvantage of gradient-based methods is that they usually converge to a local optimum, as opposed to some gradient-free techniques that can search for the global optimum. However, gradient-free methods require many more function evaluations (i.e. reservoir simulations) than gradient-based methods to find an optimum, which makes them unattractive for our purpose. In the following we will discuss some concepts from optimization theory. In particular we will state the optimality conditions for constrained and unconstrained optimization of a function of multiple variables, and we will review the method of Lagrange multipliers. There is a large amount of literature available treating optimization problems at widely varying levels of mathematical complexity. Accessible texts aimed at practical applications are Gill et al. (1986), Fletcher (1987), Rao (1996), and Luenberger and Ye (2010). Numerical aspects are covered extensively in Bonnans et al. (2003) and Nocedal and Wright (2006), which also provide an in-depth treatment of the theoretical aspects. For the theory covered in the current section on optimization theory we have in particular relied on Gill et al. (1986), and we refer to that text for further details and proofs.

6.2 Unconstrained optimization

6.2.1 Optimality conditions

Consider the unconstrained optimization problem

$$\min_u \mathcal{J}(u) \ , \tag{6.1}$$

or in words "minimize the *objective function*[†] $\mathcal{J}(u)$ by changing the control variable u". Here $\mathcal{J}$ is a *univariate* function, i.e. a function of a single variable which is u in this case.

[‡] Finding the minimum of a function is identical to finding the maximum of minus one times that function.

[†] The objective function is sometimes referred to as a *performance function*, or a *cost function*, which is then to be maximized or minimized respectively.

The conditions for a minimum $u = u^0$ are taught in any basic calculus course and we will briefly review them as a precursor to more complex problems. The first-order and second-order *necessary conditions* for a minimum are given by*

$$\left.\frac{\partial \mathcal{J}}{\partial u}\right|_{u=u^0} = 0 \ , \tag{6.2}$$

$$\left.\frac{\partial^2 \mathcal{J}}{\partial u^2}\right|_{u=u^0} \geq 0 \ , \tag{6.3}$$

while the first-order and second-order *sufficient conditions* are given by condition (6.2) together with

$$\left.\frac{\partial^2 \mathcal{J}}{\partial u^2}\right|_{u=u^0} > 0 \ . \tag{6.4}$$

More in general, a point $u = u^0$ that satisfies the necessary condition (6.2) is called a *stationary point* or a *critical point*‡. It is a minimum if the sign of the second derivative is positive (as in equation (6.4)), and a maximum if the sign is negative. In both cases, the point is called an *extreme* or an *optimal point*, and the necessary and sufficient conditions for an optimal point are therefore also known as *optimality conditions*. If the second derivative is equal to zero, u^0 is either an *inflection point* or an extreme depending on the sign of the higher derivatives. Conditions (6.2) to (6.4) can be derived through approximating $\mathcal{J}$ in the neighborhood of u^0 using a Taylor expansion:

$$\mathcal{J}(u) = \mathcal{J}\left(u^0\right) + \left[\frac{\partial \mathcal{J}(u)}{\partial u}\right]_{u=u^0}\left(u-u^0\right) + \frac{1}{2}\left[\frac{\partial^2 \mathcal{J}(u)}{\partial u^2}\right]_{u=u^0}\left(u-u^0\right)^2 + \ldots \ . \tag{6.5}$$

The necessary condition (6.2) can be obtained by considering the first two terms of this expansion: for any non-zero value of the first-order derivative it would be possible to find a smaller value for $\mathcal{J}$ than $\mathcal{J}(u^0)$. Next, by considering also the second-order term, and assuming that the first necessary condition is fulfilled, it follows that the second necessary condition is given by inequality (6.3) because for any other choice of the second-order derivative the function value $\mathcal{J}(u)$ would increase in the neighborhood of u^0. Because inequality (6.3) still allows for the possibility that $\mathcal{J}(u)$ either decreases or stays constant, we need the strict inequality (6.4) to obtain the sufficient condition: for any other choice the function value $\mathcal{J}(u)$ would not decrease in the neighborhood of u^0. A similar set of conditions can be obtained for the case where $\mathcal{J}$ is a *multivariate* function of variables $u_1, u_2, \ldots, u_m$, corresponding to an optimization problem

$$\min_{\mathbf{u}} \mathcal{J}(\mathbf{u}), \ \mathbf{u} = \begin{bmatrix} u_1 & u_2 & \cdots & u_m \end{bmatrix}^T \ . \tag{6.6}$$

The first-order necessary condition then becomes

$$\left.\frac{\partial \mathcal{J}}{\partial \mathbf{u}}\right|_{\mathbf{u}=\mathbf{u}^0} = \left[\frac{\partial \mathcal{J}}{\partial u_1} \quad \frac{\partial \mathcal{J}}{\partial u_2} \quad \cdots \quad \frac{\partial \mathcal{J}}{\partial u_m}\right]_{\mathbf{u}=\mathbf{u}^0} = \mathbf{0}^T \ , \tag{6.7}$$

* We tacitly assume that $\mathcal{J}$ is continuous and at least twice differentiable.

‡ We use superscripts 0 to indicate stationary points.

while the second-order necessary condition is the requirement that the *Hessian matrix*

$$\left.\frac{\partial^2 \mathcal{J}}{\partial \mathbf{u}^2}\right|_{\mathbf{u}=\mathbf{u}^0} = \left[\begin{array}{cccc} \dfrac{\partial^2 \mathcal{J}}{\partial u_1^2} & \dfrac{\partial^2 \mathcal{J}}{\partial u_1 \partial u_2} & \cdots & \dfrac{\partial^2 \mathcal{J}}{\partial u_1 \partial u_m} \\ \dfrac{\partial^2 \mathcal{J}}{\partial u_2 \partial u_1} & \dfrac{\partial^2 \mathcal{J}}{\partial u_2^2} & \cdots & \dfrac{\partial^2 \mathcal{J}}{\partial u_2 \partial u_m} \\ \vdots & \vdots & \ddots & \vdots \\ \dfrac{\partial^2 \mathcal{J}}{\partial u_m \partial u_1} & \dfrac{\partial^2 \mathcal{J}}{\partial u_m \partial u_2} & \cdots & \dfrac{\partial^2 \mathcal{J}}{\partial u_m^2} \end{array}\right]_{\mathbf{u}=\mathbf{u}^0} \geq 0 \ , \tag{6.8}$$

where the inequality sign implies that the matrix is positive semi-definite, i.e. that all its eigenvalues are larger than or equal to zero. The sufficient conditions are given by condition (6.7) together with the requirement that

$$\left.\frac{\partial^2 \mathcal{J}}{\partial \mathbf{u}^2}\right|_{\mathbf{u}=\mathbf{u}^0} > 0 \tag{6.9}$$

where the strict inequality sign now implies that the matrix is positive definite, i.e. that all its eigenvalues are larger than zero. In parallel to the univariate case, a point $\mathbf{u} = \mathbf{u}^0$ that satisfies the first-order necessary condition (6.7) is called a *stationary point*. Depending on the sign of the eigenvalues of the Hessian (6.9) it can be a minimum (all eigenvalues larger than zero), a maximum (all eigenvalues smaller than zero) or a *saddle point* (some eigenvalues larger, some smaller than zero). If one or more of the eigenvalues are equal to zero, and the remaining ones have an equal sign, it will be required to consider the higher derivatives to determine the character of the stationary point.

6.2.2 Convexity

In some circumstances the necessary conditions are also sufficient conditions. In particular this is often the case if the optimization problem is *convex*. A *convex set* $\mathcal{S}$ is defined as a collection of points such that a line connecting any two points of the set is entirely within the set. If the set consists of points $\mathbf{s} \in \mathbb{R}^n$ this can be written as

$$\text{if } \mathbf{s}_1, \mathbf{s}_2 \in \mathcal{S} \text{ then } \mathbf{s} \in \mathcal{S}, \tag{6.10}$$

where

$$\mathbf{s} = \beta \mathbf{s}_1 + (\beta - 1)\mathbf{s}_2 \in \mathcal{S}, 0 \leq \beta \leq 1. \tag{6.11}$$

Figure 6.11 gives some examples of two-dimensional convex and non-convex sets. A *convex function* $\mathcal{J}(\mathbf{u}) : \mathbb{R}^n \to \mathbb{R}$ is defined as a function for which the *epigraph*, i.e. all points $\mathbf{s} \in \mathbb{R}^n \times \mathbb{R}$ above the graph of $\mathcal{J}(\mathbf{u})$, form a convex set. For smooth, twice differentable functions $\mathcal{J}(u) : \mathbb{R} \to \mathbb{R}$, convexity is equal to the requirement that the second derivative $d^2\mathcal{J}/du^2$, which is also known as the *curvature*, is positive everywhere. The concepts of convexity and positive curvature can be extended to smooth functions $\mathcal{J}(\mathbf{u}) : \mathbb{R}^n \to \mathbb{R}$, in which case it is required that the Hessian matrix $\partial^2 \mathcal{J}/\partial \mathbf{u}^2$ is positive definite for all values of $\mathbf{u}$. Figure 6.12 gives some examples of one-dimensional convex and non-convex functions, and illustrates that for smooth convex problems the necessary conditions for an optimum are often also sufficient conditions. An unconstrained optimization problem is called convex if the epigraph of the objective function is a convex set. In constrained optimization problems

the constraints typically restrict the domain of the optimization variables and thus also the epigraph. Constraints can therefore make an optimization problem nonconvex, despite convexity of the objective function. Practical optimization problems in reservoir engineering are seldom convex. However, the concept of convexity is an essential element of optimization theory and forms an important ingredient in the derivation of many optimization algorithms.

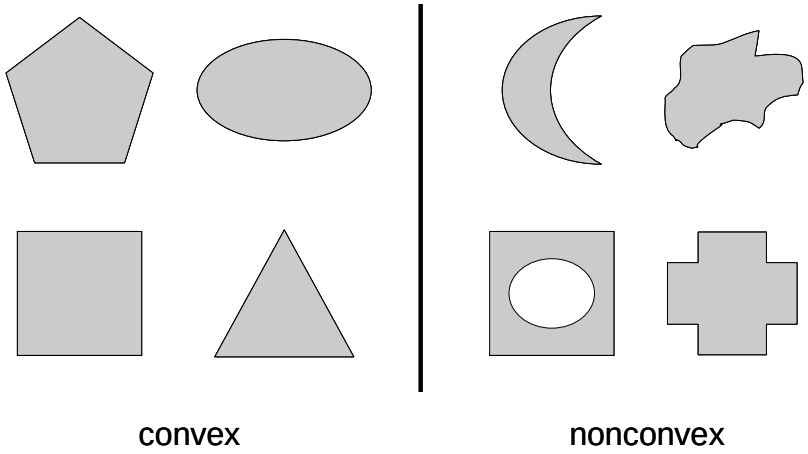


Figure 6.11: Convex and nonconvex sets of two-dimensional points.

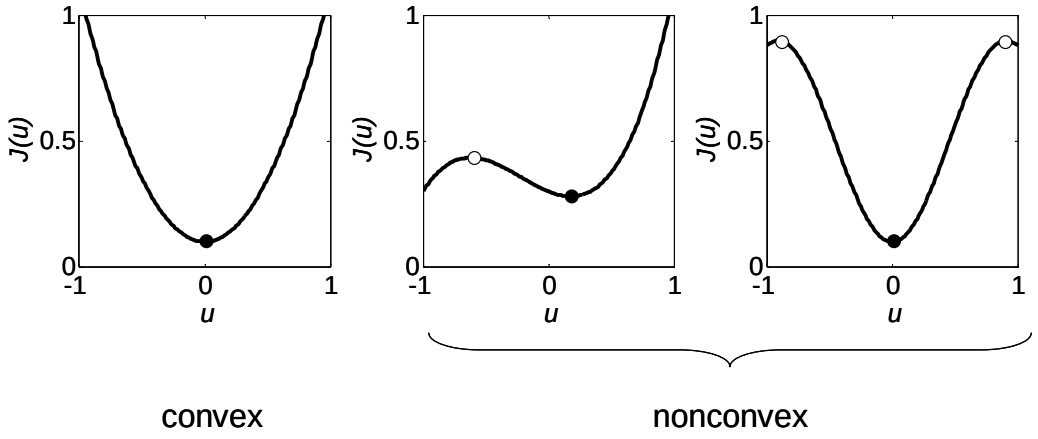


Figure 6.12: Convex and nonconvex functions $\mathcal{J}(u):\mathbb{R}\to\mathbb{R}$. The figures show only parts of the graphs for $u\in[0,1]$ and $\mathcal{J}(u)\in[0,1]$. The dots represent stationary points. The solid dots represent minima (i.e. optimal points for a minimization problem). For the convex function $\mathcal{J}(u)=0.1+u^2$ (left figure) the single stationary point is automatically the optimal point.

6.3 Constrained optimization

6.3.1 Single equality constraint

Simple example

As an introduction to constrained optimization we will discuss a simple example involving a single equality constraint. First, consider the unconstrained optimization problem

$$\min_{\mathbf{u}} \mathcal{J}(\mathbf{u}),\ \mathbf{u}=\begin{bmatrix}u_1 & u_2\end{bmatrix}^T, \tag{6.12}$$

where

$$\mathcal{J}(\mathbf{u}) = 2\left(u_1^2 + u_2^2\right) \ . \tag{6.13}$$

Equation (6.13) represents a paraboloide, and it can be seen immediately that the minimum is given by $\mathbf{u}^0 = \mathbf{0}$; see Figure 6.13 (top left). Formally we find this result by setting the derivatives of $\mathcal{J}$ with respect to u_1 and u_2 equal to zero, i.e.

$$\frac{\partial \mathcal{J}}{\partial \mathbf{u}} \equiv \left[\frac{\partial \mathcal{J}}{\partial u_1} \quad \frac{\partial \mathcal{J}}{\partial u_2}\right] \equiv \begin{bmatrix} 4u_1 & 4u_2 \end{bmatrix} = \mathbf{0}^T \ , \tag{6.14}$$

and solving for the stationary point $\mathbf{u}^0$ from the two resulting equations $4u_1 = 0$ and $4u_2 = 0$. In that case we have automatically satisfied the first-order necessary condition (6.7). Next we take the second derivatives, to check if they fulfill the second-order necessary condition (6.8), or, more interestingly, the second-order sufficient condition (6.9). The first step of this formal procedure can also be expressed as setting the total differential $\delta\mathcal{J}$ of the function $\mathcal{J}$ equal to zero:

$$\delta\mathcal{J} \equiv \frac{\partial \mathcal{J}}{\partial \mathbf{u}} \delta\mathbf{u} \equiv \begin{bmatrix} 4u_1 & 4u_2 \end{bmatrix} \begin{bmatrix} \delta u_1 \\ \delta u_2 \end{bmatrix} = 0 \ . \tag{6.15}$$

The differentials $\delta\mathcal{J}$ and $\delta\mathbf{u}$ are often referred to as the *variations* of $\mathcal{J}$ and $\mathbf{u}$ respectively[†]. The two equations for u_1 and u_2 follow by realizing that equation (6.15) should hold for arbitrary values of the variations δu_1 and δu_2 and therefore that $4u_1 = 0$ and $4u_2 = 0$. The second-order conditions can be verified similarly by taking the second variation:

$$\delta^2\mathcal{J} \equiv \delta\mathbf{u}^T \frac{\partial^2 \mathcal{J}}{\partial \mathbf{u}^2} \delta\mathbf{u} \equiv \begin{bmatrix} \delta u_1 & \delta u_2 \end{bmatrix} \begin{bmatrix} 4 & 0 \\ 0 & 4 \end{bmatrix} \begin{bmatrix} \delta u_1 \\ \delta u_2 \end{bmatrix} \ . \tag{6.16}$$

Because the Hessian at the right-hand side of equation (6.16) is diagonal its eigenvalues are equal to the diagonal terms, and because these are both larger than zero the matrix is positive definite and $\mathbf{u}^0$ is indeed a minimum.

Elimination of the constraint

Next, consider a constrained optimization example, given by the same problem statement (6.12), but now under the equality constraint

$$c(\mathbf{u}) \equiv u_1 + u_2 - 0.6 = 0 \ . \tag{6.17}$$

Equation (6.17) represents a line in the $u_1 - u_2$ plane passing through $(0, 0.6)$ and $(0.6, 0)$; see Figure 6.13 (top right and bottom right). The solution to this constrained optimization problem can be obtained by solving for u_1 or u_2 from equation (6.17), substituting the result in equation (6.13), and then proceeding as in the unconstrained case, but now taking derivatives only to the one remaining variable:

[†] Sometimes the use of the word *variation* is restricted to differentials of a *functional*, i.e. of a function of a function.

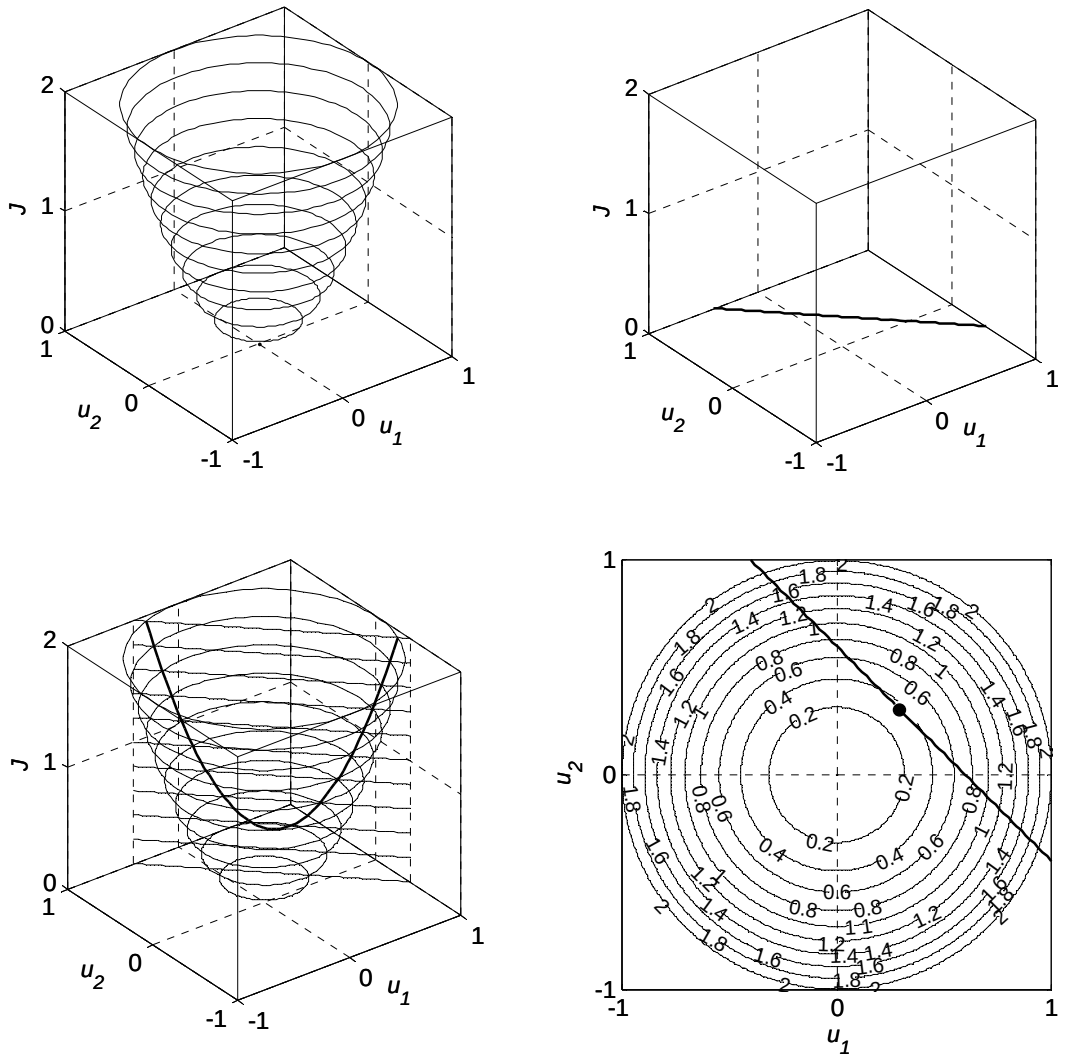


Figure 6.13: Constrained optimization. Top left: contour lines of the objective function $\mathcal{J}(\mathbf{u})$ *. Top right: constraint* $c(\mathbf{u})$ *in the* (u_1,u_2) *plane. Bottom left: Contour lines of the objective function intersected by a vertical plane (represented by a set of horizontal lines) through the constraint. The constrained minimum is at the bottom of the parabola in this plane. Bottom right: top view displaying the projected contour lines of the objective function together with the constraint. The dot indicates the minimum, and coincides with the point where the constraint curve is tangent to the contour lines.*

$$u_2 = 0.6 - u_1 \ , \tag{6.18}$$

$$\mathcal{J} = 4u_1^2 - 2.4u_1 + 0.72 \ , \tag{6.19}$$

$$\delta\mathcal{J} \equiv \frac{\partial \mathcal{J}}{\partial u_1}\delta u_1 = \left(8u_1 - 2.4\right)\delta u_1 \ . \tag{6.20}$$

The first-order necessary condition for a minimum is obtained by requiring that $\delta\mathcal{J} = 0$ for arbitrary values of δu_1, which results in the solution $u_1^0 = u_2^0 = 0.3$, corresponding to a value of the objective function $\mathcal{J}^0 = 0.36$. The second-order sufficient condition is obtained by requiring that $\delta^2\mathcal{J} > 0$ for arbitrary values of $\delta^2 u_1$, and because

$$\delta^2\mathcal{J} \equiv \frac{\partial^2 \mathcal{J}}{\partial u_1^2}\delta u_1^2 = 8\delta u_1^2 \ , \tag{6.21}$$

the solution is indeed a minimum. It is the lowest point of the parabola given by equation (6.19), as can also be seen in Figure 6.13 (bottom left).

6.3.2 Lagrange multipliers

Modified objective function

For more complex constraints it may not be possible to explicitly solve for one of the input variables as we could do in this simple example. In that case, the classic way of solving the constrained problem is through the use of the *Lagrange multiplier* technique. This involves the definition of a *modified objective function*[†]:

$$\bar{\mathcal{J}}\left(\mathbf{u}, \lambda\right) \triangleq \mathcal{J}\left(\mathbf{u}\right) + \lambda c\left(\mathbf{u}\right) \equiv 2\left(u_1^2 + u_2^2\right) + \lambda\left(u_1 + u_2 - 0.6\right), \tag{6.22}$$

where the constraint equation has been added to the original objective function after multiplication with the Lagrange multiplier λ. Note that if the constraint equation is satisfied, the term containing λ becomes zero, such that the value of the modified objective $\bar{\mathcal{J}}$ becomes equal to value of the original objective $\mathcal{J}$. We will discuss the meaning of the Lagrange multiplier later, and for the time being we just consider λ as an additional variable such that $\bar{\mathcal{J}}$ is now a function of three variables: u_1, u_2 and λ. The first step of the minimization procedure, taking the total differential and setting the result equal to zero, then gives

$$\begin{aligned} \delta\bar{\mathcal{J}} &\equiv \frac{\partial \bar{\mathcal{J}}}{\partial \mathbf{u}}\delta\mathbf{u} + \frac{\partial \bar{\mathcal{J}}}{\partial \lambda}\delta\lambda \\ &\equiv \begin{bmatrix} 4u_1 + \lambda & 4u_2 + \lambda \end{bmatrix}\begin{bmatrix} \delta u_1 \\ \delta u_2 \end{bmatrix} + \left(u_1 + u_2 - 0.6\right)\delta\lambda = 0 \, . \end{aligned} \tag{6.23}$$

The first-order necessary condition for a minimum requires that equation (6.23) should hold for arbitrary values of the variations δu_1, δu_2, and $\delta\lambda$, which results in three equations

$$\frac{\partial \bar{\mathcal{J}}}{\partial u_1} \equiv 4u_1 + \lambda = 0 \ , \tag{6.24}$$

[†] In texts on constrained optimization the modified objective function is often referred to as the *Lagrangian*. Here we will restrict the use of the term Lagrangian for a quantity that plays a role in dynamic optimization, as will be discussed in Chapter 7.

$$\frac{\partial \bar{\mathcal{J}}}{\partial u_2} \equiv 4u_2 + \lambda = 0 \ , \tag{6.25}$$

$$\frac{\partial \bar{\mathcal{J}}}{\partial \lambda} \equiv u_1 + u_2 - 0.6 = 0 \ , \tag{6.26}$$

from which we can solve for the three variables to obtain

$$\begin{bmatrix} \mathbf{u}^0 \\ \lambda^0 \end{bmatrix} = \begin{bmatrix} 0.3 \\ 0.3 \\ -1.2 \end{bmatrix} , \tag{6.27}$$

which is indeed the same result for $\mathbf{u}^0$ as was found before.

Directional derivative

As discussed above, and as illustrated in Figure 6.13 (bottom left), the minimum of the constrained problem occurs at the bottom of the parabola that is formed by only considering those values of the objective function corresponding to points (u_1,u_2) that obey the constraint equation (i.e. those values that are exactly above the constraint 'curve' (which here is a line) in the (u_1,u_2) plane)[†]. We could therefore also search for the minimum by considering the *directional derivative* of the objective function along the constraint which is defined as the length of the vector formed by projecting the derivative $\partial\mathcal{J}/\partial\mathbf{u}$ on the tangent to the constraint curve:

$$\left.\frac{\partial \mathcal{J}}{\partial \mathbf{u}}\right|_{\mathbf{v}} \triangleq \frac{\partial \mathcal{J}}{\partial \mathbf{u}}\mathbf{v} \equiv \begin{bmatrix} \dfrac{\partial \mathcal{J}}{\partial u_1} & \dfrac{\partial \mathcal{J}}{\partial u_2} \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} , \tag{6.28}$$

where $\mathbf{v} \equiv \mathbf{v}(\mathbf{u})$ is the unit vector[§] of the tangent in a point (u_1,u_2). Filling in the numbers from the numerical example and equating the result to zero gives

$$\left.\frac{\partial \mathcal{J}}{\partial \mathbf{u}}\right|_{\mathbf{v}} \equiv \begin{bmatrix} 4u_1 & 4u_2 \end{bmatrix} \begin{bmatrix} 1 \\ -1 \end{bmatrix} \equiv 4u_1 - 4u_2 = 0 \ , \tag{6.29}$$

which, together with constraint equation (6.17) results in the same optimum that we found before. Note that the directional derivative[*] becomes zero when $\mathbf{v}$ is perpendicular to $\partial\mathcal{J}/\partial\mathbf{u}$. In that case $\mathbf{v}$ is not only tangent to the constraint but also tangent to the contour lines of the objective function; see Figure 6.13 (bottom right). At the same time $\partial\mathcal{J}/\partial\mathbf{u}$ will then have the same or the opposite direction as the derivative $\partial c/\partial\mathbf{u}$[†]. Therefore we find that at the minimum we have

---

[†] Points (u_1,u_2) obeying the constraints are known as *feasible points*, and the set of all feasible points as the *feasible set*.

[§] In our simple example the components $[1 \quad -1]^T$ of $\mathbf{v}$ are constant for all values of u_1 and u_2; see Figure 6.13 (top right and bottom right). However, in the general case $\mathbf{v}$ is a function of $\mathbf{u}$.

[*] The directional derivative is the transpose of the *projected gradient* of $\mathcal{J}$, defined as $\nabla\mathcal{J}\,|_{\mathbf{v}} \triangleq \mathbf{v}^T\nabla\mathcal{J}$.

[†] The constraint equation $c(\mathbf{u}) = 0$ can be interpreted as a single contour line of the function $c(\mathbf{u})$. Just like for the objective function, the steepest descent direction, i.e. the maximum value of $\partial c/\partial\mathbf{u}$, is perpendicular to the contour lines, and therefore to the constraint curve in the (u_1,u_2) plane.

$$\left.\frac{\partial \mathcal{J}}{\partial \mathbf{u}}\right|_{\mathbf{u}=\mathbf{u}^0} = -\lambda \left.\frac{\partial c}{\partial \mathbf{u}}\right|_{\mathbf{u}=\mathbf{u}^0}, \tag{6.30}$$

where λ is an arbitrary constant. Equation (6.30) implies that the two derivative vectors have the same or opposite directions, but different magnitudes with a ratio that is equal to the arbitrary constant λ. The reason to choose λ to be preceded by a minus sign becomes clear if we differentiate definition (6.22) with respect to $\mathbf{u}$ which results in

$$\frac{\partial \bar{\mathcal{J}}}{\partial \mathbf{u}} = \frac{\partial \mathcal{J}}{\partial \mathbf{u}} + \lambda \frac{\partial c}{\partial \mathbf{u}}, \tag{6.31}$$

and because we have $\partial \bar{\mathcal{J}} / \partial \mathbf{u}|_{\mathbf{u}=\mathbf{u}^0} = \mathbf{0}$ we find that for the optimal value $\mathbf{u}^0$ equations (6.30) and (6.31) are now identical. The arbitrary constant λ is therefore just the Lagrange multiplier that was introduced in a more ad-hoc way before. The use of the minus sign serves only to remain consistent with our earlier definition of the Lagrange multiplier[‡]. Returning to equation (6.28) we can simply verify that the first-order necessary condition is satisfied, i.e. that the directional derivative

$$\begin{bmatrix} \dfrac{\partial \mathcal{J}}{\partial u_1} & \dfrac{\partial \mathcal{J}}{\partial u_2} \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \begin{bmatrix} 4u_1 & 4u_2 \end{bmatrix} \begin{bmatrix} 1 \\ -1 \end{bmatrix}, \tag{6.32}$$

is indeed equal to zero in the stationary point $\mathbf{u}^0 = [0.3 \quad 0.3]^T$.

Implicit differentiation

An alternative route to introduce the Lagrange multiplier method is as follows. A key step in the elimination of the constraint as initially considered in the simple example above was the possibility to write u_2 as an explicit function of u_1; see equation (6.18). For more complex constraints it will usually not be possible to compute such an explicit relationship. However, it will often be possible to compute the derivative $\partial u_2 / \partial u_1$ and in that case we can simply write

$$\delta \mathcal{J} = \left(\frac{\partial \mathcal{J}}{\partial u_1} + \frac{\partial \mathcal{J}}{\partial u_2} \frac{\partial u_2}{\partial u_1} \right) \delta u_1 . \tag{6.33}$$

The key step in this procedure is the computation of the derivative $\partial u_2 / \partial u_1$. Usually this can not be done explicitly, in which case we can use the technique of *implicit differentiation* as follows. Starting from the implicit constraint equation

$$c(u_1, u_2) = 0, \tag{6.34}$$

we find that

$$\delta c = \frac{\partial c}{\partial u_1} \delta u_1 + \frac{\partial c}{\partial u_2} \delta u_2 = 0, \tag{6.35}$$

which leads to

[‡] Alternatively, some texts define the modified objective function as $\bar{\mathcal{J}}(\mathbf{u}, \lambda) \triangleq \mathcal{J}(\mathbf{u}) - \lambda c(\mathbf{u})$, i.e. with a minus sign in front of the Lagrange multiplier λ.

$$\delta u_2 = -\left(\frac{\partial c}{\partial u_2}\right)^{-1} \frac{\partial c}{\partial u_1} \delta u_1 , \tag{6.36}$$

and thus to the implicit derivative

$$\frac{\partial u_2}{\partial u_1} = -\left(\frac{\partial c}{\partial u_2}\right)^{-1} \frac{\partial c}{\partial u_1} . \tag{6.37}$$

Substitution of this expression in equation (6.33) results in

$$\delta \mathcal{J} = \left[\frac{\partial \mathcal{J}}{\partial u_1} - \frac{\partial \mathcal{J}}{\partial u_2}\left(\frac{\partial c}{\partial u_2}\right)^{-1} \frac{\partial c}{\partial u_1}\right] \delta u_1 . \tag{6.38}$$

In a stationary point this expression should be zero for arbitrary values of δu_1 which means that

$$\frac{\partial \mathcal{J}}{\partial u_1} + \underbrace{-\frac{\partial \mathcal{J}}{\partial u_2}\left(\frac{\partial c}{\partial u_2}\right)^{-1}}_{\lambda_1} \frac{\partial c}{\partial u_1} = 0 , \tag{6.39}$$

where we have now introduced λ_1 as a short-cut notation for $-\partial \mathcal{J}/\partial u_2 (\partial c/\partial u_2)^{-1}$. Following a simular reasoning we can derive that

$$\frac{\partial \mathcal{J}}{\partial u_2} + \underbrace{-\frac{\partial \mathcal{J}}{\partial u_1}\left(\frac{\partial c}{\partial u_1}\right)^{-1}}_{\lambda_2} \frac{\partial c}{\partial u_2} = 0 , \tag{6.40}$$

where λ_2 is a short-cut notation for $-\partial \mathcal{J}/\partial u_1 (\partial c/\partial u_1)^{-1}$. Now, if $\lambda_1 = \lambda_2$, i.e. if

$$\frac{\partial \mathcal{J}}{\partial u_2}\left(\frac{\partial c}{\partial u_2}\right)^{-1} = \frac{\partial \mathcal{J}}{\partial u_1}\left(\frac{\partial c}{\partial u_1}\right)^{-1} , \tag{6.41}$$

we can combine equations (6.39) and (6.40) into

$$\frac{\partial \mathcal{J}}{\partial \mathbf{u}} + \lambda \frac{\partial c}{\partial \mathbf{u}} = \mathbf{0}^T . \tag{6.42}$$

Condition (6.41) implies that the components of $\partial \mathcal{J}/\partial \mathbf{u}$ and $\partial c/\partial \mathbf{u}$ are pair-wise proportional which is just the case in a stationary point. Comparison of equation (6.42) with equation (6.30) shows that in this way we again find λ as an arbitrary constant multiplying the derivative of the constraint. Filling in the numerical values of the example in equation (6.39) and in the constraint (6.17) we obtain

$$\left.\begin{aligned} 4u_1^0 + \underbrace{-4u_2^0 \times (1)^{-1}}_{\lambda} \times 1 = 0 \\ u_1^0 + u_2^0 - 0.6 = 0 \end{aligned}\right\} \rightarrow \begin{bmatrix} \mathbf{u}^0 \\ \lambda^0 \end{bmatrix} = \begin{bmatrix} 0.3 \\ 0.3 \\ -1.2 \end{bmatrix} , \tag{6.43}$$

which is indeed again the result derived before, and which illustrates that the use of Lagrange multipliers to compute the derivatives of a constrained objective function can also be

interpreted as a form of implicit differentiation. An essential role in this derivation is played by equation (6.35) which defines the *admissible variations*, i.e. those combinations of variations δu_1 and δu_2 that keep the constraint condition fulfilled.

6.3.3 Multiple equality constraints

Feasible arcs

Until now we considered the minimization of a function with a single equality constraint, but the method of Lagrange multipliers can be generalized to cope with multiple constraints which may be equalities, inequalities or a combination of both. We will discuss inequality constraints later, while we will now address the extension to multiple equality constraints $c_i = 0, i = 1, \ldots, p,$ which can be stacked in a constraint vector $\mathbf{c}$ as

$$\mathbf{c}(\mathbf{u}) = \mathbf{0}, \mathbf{c} \in \mathbb{R}^p, \mathbf{u} \in \mathbb{R}^m. \tag{6.44}$$

In general, the number of constraints, p, should be less than or equal to the number of control variables, m, otherwise the problem is over-constrained and there will be no solution except for special cases. This requirement can be made more precise in case of multiple linear constraints, which can be expressed as

$$\mathbf{c}(\mathbf{u}) \equiv \mathbf{A}\mathbf{u} + \mathbf{b} = \mathbf{0}. \tag{6.45}$$

where $\mathbf{A} \in \mathbb{R}^{p \times m}$ and $\mathbf{b} \in \mathbb{R}^p$. If the rows of $\mathbf{A}$ are independent, each row of the matrix-vector equation (6.45) removes a degree of freedom from the optimization problem. The number of independent rows of $\mathbf{A}$, i.e. its rank, therefore determines the number of constraints. In the extreme case of $\text{rank}(\mathbf{A}) = m$, we have as many constraints as input variables $u_i, i = 1, \ldots, m,$ and there is no freedom left for optimization. Another property that follows from equation (6.45) is the condition for admissible variations. Because

$$\frac{\partial \mathbf{c}}{\partial \mathbf{u}} = \mathbf{A}, \tag{6.46}$$

we find, in analogy to equation (6.35), that admissible variations in case of multiple linear constraints have to obey

$$\mathbf{A}\delta\mathbf{u} = \mathbf{0}. \tag{6.47}$$

I.e., the admissible variations are in the null space of the matrix $\mathbf{A}$. Points $\mathbf{u}$ that obey the constraint equations are called *feasible points*. The set of all feasible points $\mathbf{u}$ is called the *feasible set*, and the admissible variations $\delta\mathbf{u}$ the *feasible directions*. Note that any movement in a feasible direction will remain on the constraint. In case of nonlinear constraints we can use a Taylor expansion around a feasible point $\mathbf{u}^*$,

$$\mathbf{c}(\mathbf{u}) = \mathbf{c}(\mathbf{u}^*) + \left.\frac{\partial \mathbf{c}}{\partial \mathbf{u}}\right|_{\mathbf{u}=\mathbf{u}^*} (\mathbf{u} - \mathbf{u}^*) + \ldots, \tag{6.48}$$

such that $\mathbf{A}$ can be interpreted as

$$\mathbf{A} \equiv \mathbf{A}(\mathbf{u}^*) \triangleq \left.\frac{\partial \mathbf{c}}{\partial \mathbf{u}}\right|_{\mathbf{u}=\mathbf{u}^*}. \tag{6.49}$$

In this nonlinear case it is not sufficient to just consider feasible directions, i.e. straight lines along which we can move without violating the constraints, but we have to consider *feasible*

arcs, i.e. curved lines. Equation (6.47) is therefore still a necessary condition for admissible variations, but no longer a sufficient one. For example, consider a three-dimensional optimization problem with input vector $\mathbf{u} = [u_1 \quad u_2 \quad u_3]^T$ and two nonlinear constraints

$$c_1 \equiv u_3 - \left(u_1^2 + u_2^2\right) = 0 \quad \text{and} \quad c_2 \equiv u_3 + \left(u_1^2 + u_2^2\right) = 0 \ . \tag{6.50, 6.51}$$

The constraints are paraboloids that just touch each other in the origin which is therefore the only feasible point. The matrix **A** in the feasible point follows as

$$\mathbf{A} \equiv \left.\frac{\partial \mathbf{c}}{\partial \mathbf{u}}\right|_{\mathbf{u}=\mathbf{0}} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix}, \tag{6.52}$$

such that any vector $\mathbf{u} = [\delta u_1 \quad \delta u_2 \quad 0]^T$ fulfills equation (6.47), i.e. the null space is identical to the u_1-u_2 plane. However, because the origin is the only feasible point, feasible arcs simply don't exist in this case. The additional requirements that are necessary to specify admissible variations in case of nonlinear constraints are known as *constraint qualifications*. These exist in various forms, see e.g. Bonnans et al. (2003) or Nocedal and Wright (2006), but most of them are outside the scope of our text. An exception is the simple constraint qualification given by the requirement that **A** has full row rank, i.e. that it has independent rows. Consider a three-dimensional optimization problem again, but this time with the constraints

$$c_1 \equiv u_3 - u_1 = 0 \quad \text{and} \quad c_2 \equiv u_3 - u_1^2 = 0 \ . \tag{6.53, 6.54}$$

These constraints are a plane and a parabolic cylinder (a 'parabolic tunnel') respectively that intersect each other at two lines: one for which $u_1 = u_3 = 0$ (i.e. the u_2-axis) and a parallel one for which $u_1 = u_3 = 1$. The matrices **A** now become

$$\mathbf{A}_1 \equiv \left.\frac{\partial \mathbf{c}}{\partial \mathbf{u}}\right|_{\mathbf{u}=[0 \quad u_2 \quad 0]^T} = \begin{bmatrix} -1 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix}, \tag{6.55}$$

$$\mathbf{A}_2 \equiv \left.\frac{\partial \mathbf{c}}{\partial \mathbf{u}}\right|_{\mathbf{u}=[1 \quad u_2 \quad 1]^T} = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 1 \end{bmatrix}, \tag{6.56}$$

which both have independent rows and therefore meet the constraint qualification. The admissible variations are in the null spaces of $\mathbf{A}_{1,2}$ and are given by $\delta\mathbf{u} = [0 \quad \delta u_2 \quad 0]^T$. In other words the feasible arcs are in this case lines with feasible directions parallel to the u_2-axis.

Necessary optimality conditions

Under the assumption that **A** has full row rank, i.e. that the constraint qualifications are met, we can now derive the first-order necessary conditions for a minimum in case of multiple constraints by considering the modified objective function (c.f. equation (6.22))

$$\bar{\mathcal{J}}(\mathbf{u}, \boldsymbol{\lambda}) \triangleq \mathcal{J}(\mathbf{u}) + \boldsymbol{\lambda}^T \mathbf{c}(\mathbf{u}), \tag{6.57}$$

where $\boldsymbol{\lambda} \in \mathbb{R}^p$ is a vector of Lagrange multipliers. Just as for the single-constraint case, stationarity of the modified objective function provides the first-order necessary conditions for a constrained minimum. The complete set of necessary first-order optimality conditions can therefore be written as

$$\frac{\partial \bar{\mathcal{J}}}{\partial \mathbf{u}} = \mathbf{0}^T , \tag{6.58}$$

$$\frac{\partial \bar{\mathcal{J}}}{\partial \boldsymbol{\lambda}} = \mathbf{0}^T , \tag{6.59}$$

where equation (6.59) is just a restatement of the equality constraint (6.44). Conditions (6.58) and (6.59) are sometimes referred to as the *Euler-Lagrange equations*. As before we may, alternatively, consider the directional derivatives along the constraints, and, in analogy to equation (6.28), write

$$\left.\frac{\partial \mathcal{J}}{\partial \mathbf{u}}\right|_{\mathrm{V}} \triangleq \frac{\partial \mathcal{J}}{\partial \mathbf{u}}\mathbf{V} \equiv \frac{\partial \mathcal{J}}{\partial \mathbf{u}}\begin{bmatrix}\mathbf{v}_1 & \mathbf{v}_2 & \cdots & \mathbf{v}_p\end{bmatrix} , \tag{6.60}$$

where $\mathbf{v}_i \equiv \mathbf{v}(\mathbf{u})_i , i = 1,\ldots,p$ are the unit vectors of feasible directions, tangent to the constraints, in a feasible point $\mathbf{u}$. In fact, because the admissible variations along the constraints obey equation (6.47), the columns of $\mathbf{V}$ may be any set of vectors that form a basis for the null space of $\mathbf{A}$. Following the same reasoning as for the single-constraint case we note that we obtain $\partial \mathcal{J}/\partial \mathbf{u}\,|_{\mathrm{V}} = \mathbf{0}^T$ when all columns $\mathbf{v}^i$ of $\mathbf{V}$ are perpendicular to $\partial \mathcal{J}/\partial \mathbf{u}$. In that case the columns $\mathbf{v}^i$ are not only tangent to the corresponding constraints c_i but also tangent to the contour lines of the objective function, and the derivatives $\partial c_i/\partial \mathbf{u}$ will then have the same or the opposite direction as $\partial \mathcal{J}/\partial \mathbf{u}$:

$$\left.\frac{\partial \mathcal{J}}{\partial \mathbf{u}}\right|_{\mathbf{u}=\mathbf{u}^0} = -\boldsymbol{\lambda}^T \left.\frac{\partial \mathbf{c}}{\partial \mathbf{u}}\right|_{\mathbf{u}=\mathbf{u}^0} \equiv -\boldsymbol{\lambda}^T \mathbf{A}^0 , \tag{6.61}$$

which implies that in a stationary feasible point $\mathbf{u}^0$ the derivative of the objective function, $\partial \mathcal{J}/\partial \mathbf{u}$, is a linear combination, with coefficients λ_i, of the constraint derivatives $\partial c_i/\partial \mathbf{u}, i = 1,\ldots,p$. Note that, just as in equation (6.30) we added a minus sign in front of λ to remain consistent with our earlier definition of the Lagrange multipliers.

6.3.4 Interpretation of the Lagrange multipliers

The magnitude of the Lagrange multipliers can be interpreted as a measure of the effect of perturbing the constraints on the value of the objective function. This can be seen by applying definition (6.57) to a perturbed constraint[†] $\tilde{\mathbf{c}}(\mathbf{u}) = \mathbf{c}(\mathbf{u}) + \delta\mathbf{c}$:

$$\tilde{\bar{\mathcal{J}}}(\mathbf{u},\boldsymbol{\lambda}) = \mathcal{J}(\mathbf{u}) + \boldsymbol{\lambda}^T\tilde{\mathbf{c}}(\mathbf{u}) \equiv \mathcal{J}(\mathbf{u}) + \boldsymbol{\lambda}^T\left[\mathbf{c}(\mathbf{u}) + \delta\mathbf{c}\right] . \tag{6.62}$$

The difference in the modified objective function value between a perturbed and an unperturbed constraint then follows as

$$\delta\bar{\mathcal{J}} \equiv \tilde{\bar{\mathcal{J}}}(\mathbf{u},\boldsymbol{\lambda}) - \bar{\mathcal{J}}(\mathbf{u},\boldsymbol{\lambda}) = \boldsymbol{\lambda}^T\delta\mathbf{c} . \tag{6.63}$$

In the perturbed and unperturbed optima we have $\bar{\mathcal{J}}(\mathbf{u}^0,\boldsymbol{\lambda}^0) = \mathcal{J}(\mathbf{u}^0)$ and $\tilde{\bar{\mathcal{J}}}(\tilde{\mathbf{u}}^0,\tilde{\boldsymbol{\lambda}}^0) = \tilde{\mathcal{J}}(\tilde{\mathbf{u}}^0)$, and therefore we can also write

$$\delta\mathcal{J} \equiv \tilde{\mathcal{J}}\left(\tilde{\mathbf{u}}^0\right) - \mathcal{J}\left(\mathbf{u}^0\right) = \boldsymbol{\lambda}^T\delta\mathbf{c} . \tag{6.64}$$

[†] The perturbation $\delta\mathbf{c}$ as defined here can also be interpreted as the *residual* in the constraint, i.e. its deviation from $\mathbf{0}$.

As an illustration, consider a spherical objective function in three-dimensional input space with its center in $(u_1, u_2, u_3) = (2, 0, 0)$:

$$\mathcal{J}(\mathbf{u}) = (u_1 - 2)^2 + u_2^2 + u_3^2 \ , \tag{6.65}$$

and with constraints given by equations (6.53) and (6.54) such that the modified objective function becomes

$$\bar{\mathcal{J}}(\mathbf{u}, \boldsymbol{\lambda}) \equiv \mathcal{J}(\mathbf{u}) + \boldsymbol{\lambda}^T \mathbf{c} = (u_1 - 2)^2 + u_2^2 + u_3^2 + \begin{bmatrix} \lambda_1 & \lambda_2 \end{bmatrix} \begin{bmatrix} u_3 - u_1 \\ u_3 - u_1^2 \end{bmatrix} . \tag{6.66}$$

Proceeding in the usual fashion, i.e. taking the first variation and setting the coefficients multiplying δu_1 up to $\delta\lambda_2$ equal to zero, we obtain the set of five equations

$$\frac{\partial \bar{\mathcal{J}}}{\partial u_1} \equiv 2(1 - \lambda_2) u_1 - (4 + \lambda_1) = 0 \ , \tag{6.67}$$

$$\frac{\partial \bar{\mathcal{J}}}{\partial u_2} \equiv 2u_2 = 0 \ , \tag{6.68}$$

$$\frac{\partial \bar{\mathcal{J}}}{\partial u_3} \equiv 2u_3 + \lambda_1 + \lambda_2 = 0 \ , \tag{6.69}$$

$$\frac{\partial \bar{\mathcal{J}}}{\partial \lambda_1} \equiv u_3 - u_1 = 0 \ , \tag{6.70}$$

$$\frac{\partial \bar{\mathcal{J}}}{\partial \lambda_2} \equiv u_3 - u_1^2 = 0 \ , \tag{6.71}$$

from which the constrained minimum and the corresponding objective function value can be computed as either

$$\begin{bmatrix} u_1^0 & u_2^0 & u_3^0 & \lambda_1^0 & \lambda_2^0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & -4 & 4 \end{bmatrix} \text{ with } \mathcal{J}(\mathbf{u}^0) = 4 \ , \tag{6.72, 6.73}$$

or

$$\begin{bmatrix} u_1^0 & u_2^0 & u_3^0 & \lambda_1^0 & \lambda_2^0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 & -2 & 0 \end{bmatrix} \text{ with } \mathcal{J}(\mathbf{u}^0) = 2 \ , \tag{6.74, 6.75}$$

such that clearly the latter solution corresponds to the minimum; see also Figure 6.14 (left). If we specify the perturbed constraints

$$\tilde{c}_1 = u_3 - u_1 + \delta c_1 \ \text{ and } \ \tilde{c}_2 = u_3 - u_1^2 + \delta c_2 \ . \tag{6.76, 6.77}$$

we can repeat the computation for the minimum which leads to

$$\tilde{\mathbf{u}}^0 \equiv \begin{bmatrix} \tilde{u}_1^0 & \tilde{u}_2^0 & \tilde{u}_3^0 \end{bmatrix}^T \approx \begin{bmatrix} 1 - \delta c_1 + \delta c_2 & 0 & 1 - 2\delta c_1 + \delta c_2 \end{bmatrix}^T \ , \tag{6.78}$$

and

$$\mathcal{J}(\tilde{\mathbf{u}}^0) \approx 2 - 2\delta c_1 \ , \tag{6.79}$$

where we used the Taylor expansion $\sqrt{1+\varepsilon} = 1 + \tfrac{1}{2}\varepsilon - \ldots$ and neglected the quadratic terms in δc_1 and δc_2; see Figure 6.14 (right). Clearly, the magnitude of the Lagrange multipliers (

$\lambda_1^0 = -2$ and $\lambda_2^0 = 0$) is a first-order measure of the effect on $\mathcal{J}$ of perturbing the constraints with δc_1 and δc_2. In fact perturbing c_2 while staying on c_1 does not have any effect at all (at least to first order) because the corresponding Lagrange multiplier λ_2^0 is equal to zero. This can also be understood by considering Figure 6.14: staying in the plane of the paper and moving away from the optimal point $\mathbf{u}^0$ (the solid dot) along the linear constraint means staying tangent to the circles, at least up to first order. In other words, in this case we do not need the second constraint to arrive at the constrained minimum $\mathbf{u}^0$.

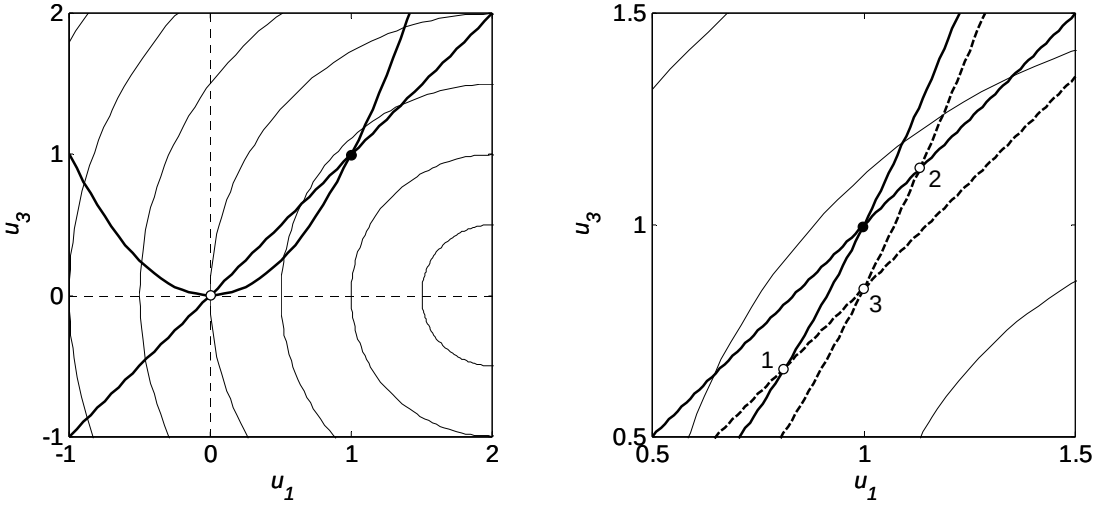


Figure 6.14: Spherical objective function with two constraints. Left: Cross-section in the u_1-u_3 plane. The circles are cross-sections through the contour spheres of the objective function. The line and the parabola are cross-sections through the plane and the parabolic cylinder that form the constraints c_1 and c_2. The two dots are cross-sections through the two lines that together form the feasible set. The dots also indicate the constrained stationary points where these two feasible lines (which are perpendicular to the paper) are tangent to the contour spheres of the objective function. The solid dot corresponds to the minimum. Right: Detail showing the effect of perturbing the constraints. The solid dot corresponds to the minimum in the unperturbed case. Open dots nr. 1 and nr. 2 correspond to the minima when perturbing constraint c_1 and c_2 with $\delta c_1 = 0.15$ and $\delta c_2 = 0.15$ respectively, while open dot nr. 3 corresponds to the minimum when perturbing both constraints.

6.3.5 Inequality constraints

Constraint activity

Once we allow the possibility of *inequality constraints*, we need to make a distinction between feasible points that are *on* a constraint, and those that are not. The constraints related to these two categories of feasible points are called *active constraints* and *inactive constraints* respectively. Both categories of feasible points are said to *satisfy* the constraints, whereas infeasible points *violate* the constraints. Inactive constraints do not restrict the feasibility of perturbations from a feasible point, i.e., even if a feasible point $\mathbf{u}$ is very close to a constraint, there is, in theory, always[‡] room for a small feasible change $\delta\mathbf{u}$. Active constraints, however, restrict the feasible perturbations. Two types of feasible perturbations can now be

[‡] In numerical computations, this may be an impractically small value such that we can consider the constraint active for practical purposes, but for the time being we will restrict our attention to the theoretical case.

distinguished: those that keep the constraint active, known as *binding perturbations*, and those that make it inactive by moving away from it, known as *non-binding perturbations*. To derive the first- and second-order optimality conditions we have to consider the active constraints. Starting with a set of linear inequality constraints

$$\mathbf{d}(\mathbf{u}) \equiv \mathbf{A}\mathbf{u} + \mathbf{b} \le \mathbf{0} \ , \tag{6.80}$$

we can partition them as

$$\begin{bmatrix} \hat{\mathbf{d}} \\ \not{\mathbf{d}} \end{bmatrix} \equiv \begin{bmatrix} \hat{\mathbf{A}} \\ \not{\mathbf{A}} \end{bmatrix} \mathbf{u} + \begin{bmatrix} \hat{\mathbf{b}} \\ \not{\mathbf{b}} \end{bmatrix} \le \mathbf{0} \ , \tag{6.81}$$

where the hatted and the striked-through coefficients $\mathbf{A}$ and $\mathbf{b}$ correspond to the active and inactive constraints $\hat{\mathbf{d}}$ and $\not{\mathbf{d}}$ respectively. In analogy to the equality-constrained case we can now derive that admissible variations in the form of binding perturbations are in the null space of $\hat{\mathbf{A}}$:

$$\hat{\mathbf{A}}\delta\mathbf{u} = \mathbf{0} \, . \tag{6.82}$$

Assuming that the rows of $\hat{\mathbf{A}}$ are linearly independent, i.e. that the constraint qualification is met, a first-order necessary condition for optimality of a feasible point with active linear constraints $\hat{\mathbf{d}}$ is therefore given by (c.f. equation (6.61))

$$\left.\frac{\partial \mathcal{J}}{\partial \mathbf{u}}\right|_{\mathbf{u}=\mathbf{u}^0} = -\hat{\boldsymbol{\lambda}}^T \left.\frac{\partial \hat{\mathbf{d}}}{\partial \mathbf{u}}\right|_{\mathbf{u}=\mathbf{u}^0} \equiv -\hat{\boldsymbol{\lambda}}^T \hat{\mathbf{A}}^0 \ , \tag{6.83}$$

where $\hat{\boldsymbol{\lambda}}$ is the vector of Lagrange multipliers corresponding to the active constraints. In analogy to the case with pure equality constraints, equation (6.83) implies that in a stationary feasible point $\mathbf{u}^0$ the derivative of the objective function, $\partial \mathcal{J} / \partial \mathbf{u}$, is a linear combination, with coefficients $\hat{\lambda}_i$, of the active constraint derivatives $\partial \hat{d}_i / \partial \mathbf{u}$, $i = 1, \ldots, p$. However, we should also consider the possibility of variations in the form of non-binding perturbations, in which case it holds for at least one row $\hat{\mathbf{a}}^i$ of $\hat{\mathbf{A}}$ (corresponding to active constraint *i*) that

$$\hat{\mathbf{a}}^i \delta\mathbf{u} < 0 \, , \tag{6.84}$$

or that

$$\hat{\mathbf{a}}^i \delta\mathbf{u} > 0 \, , \tag{6.85}$$

Inequality (6.84) corresponds to a variation $\delta\mathbf{u}$ around the feasible point $\mathbf{u}$ that results in moving off the constraint into a feasible direction because in that case

$$\hat{\mathbf{a}}^i \left(\mathbf{u} + \delta\mathbf{u}\right) + b_i < \hat{\mathbf{a}}^i \mathbf{u} + b_i = 0 \, , \tag{6.86}$$

such that inequality (6.80) remains valid, but with a constraint that now becomes inactive. Opposedly, inequality (6.85) corresponds to moving off the constraint into an infeasible direction, i.e. to violating the constraint. In order for a feasible point $\mathbf{u}$ that obeys equation (6.83) to be a minimum, all non-binding perturbations $\delta\mathbf{u}$ should first of all be in a feasible direction, i.e. they should obey equation (6.84). For all perturbations we therefore require that

$$\hat{\mathbf{A}}^0 \delta\mathbf{u} \le \mathbf{0} \, . \tag{6.87}$$

Moreover, the non-binding perturbations should result in an increase of $\mathcal{J}$, because if at least one of them would result in a decrease, clearly $\mathbf{u}$ would not be a minimum. In other words we require that for any $\delta\mathbf{u}$ obeying equation (6.87) we also have

$$\left.\frac{\partial \mathcal{J}}{\partial \mathbf{u}}\right|_{\mathbf{u}=\mathbf{u}^0} \delta\mathbf{u} \geq 0\,. \tag{6.88}$$

With the aid of equation (6.83) this can be rewritten as

$$\hat{\boldsymbol{\lambda}}^T \hat{\mathbf{A}}^0 \delta\mathbf{u} \leq 0\ , \tag{6.89}$$

and therefore, since $\delta\mathbf{u}$ obeys equation (6.87), we find the additional necessary condition

$$\hat{\boldsymbol{\lambda}} \geq \mathbf{0}\ . \tag{6.90}$$

A similar reasoning holds for the case of nonlinear inequality constraints

$$\mathbf{d}\left(\mathbf{u}\right) \leq \mathbf{0}\ . \tag{6.91}$$

Using a Taylor expansion around a feasible point $\mathbf{u}^*$,

$$\mathbf{d}\left(\mathbf{u}\right) = \mathbf{d}\left(\mathbf{u}^*\right) + \left.\frac{\partial \mathbf{d}}{\partial \mathbf{u}}\right|_{\mathbf{u}=\mathbf{u}^*} \left(\mathbf{u}-\mathbf{u}^*\right) + \ldots, \tag{6.92}$$

and introducing the partitioning $\mathbf{d} = [\hat{\mathbf{d}}^T \ \not{\mathbf{d}}^T]^T$ in active and inactive constraints as before, the matrix $\hat{\mathbf{A}}$ now becomes a function of $\mathbf{u}^*$:

$$\hat{\mathbf{A}} \equiv \hat{\mathbf{A}}\left(\mathbf{u}^*\right) \triangleq \left.\frac{\partial \hat{\mathbf{d}}}{\partial \mathbf{u}}\right|_{\mathbf{u}=\mathbf{u}^*}\ . \tag{6.93}$$

As discussed in the previous section, the magnitude of the Lagrange multipliers is a first-order measure of the effect of perturbing the constraints. The special case of equation (6.90) in which at least one multipliers is zero implies a situation where the corresponding constraint, although active, is actually not functioning (at least to first order) because slightly moving the constraint does not change the objective function value; see also the example in Figure 6.14 (right). Such an active constraint with a zero Lagrange multiplier is said to be *weakly active*, as opposed to an active constraint with a positive multiplier which is called *strongly active*. Most texts on optimization also introduce Lagrange multipliers corresponding to inactive constraints which are then given a zero value by definition.

Necessary optimality conditions

Just as for the equality-constrained case we can now define the modified objective function (c.f. equation (6.57))

$$\bar{\mathcal{J}}\left(\mathbf{u},\boldsymbol{\lambda}\right) \triangleq \mathcal{J}\left(\mathbf{u}\right) + \boldsymbol{\lambda}^T \mathbf{d}\left(\mathbf{u}\right), \tag{6.94}$$

and express the first-order necessary conditions for a minimum as

$$\frac{\partial \bar{\mathcal{J}}}{\partial \mathbf{u}} = \mathbf{0}^T\ , \tag{6.95}$$

$$\frac{\partial \bar{\mathcal{J}}}{\partial \boldsymbol{\lambda}} \leq \mathbf{0}^T \ , \tag{6.96}$$

$$\boldsymbol{\lambda} \geq \mathbf{0}^T \ , \tag{6.97}$$

$$\boldsymbol{\lambda}^T \mathbf{d} = 0 \ . \tag{6.98}$$

Equations (6.95) to (6.98) are referred to as the *Karush-Kuhn-Tucker (KKT) conditions*, or sometimes just the *Kuhn-Tucker conditions*, and they can be intepreted as a special form of the Euler-Lagrange equations, adapted to inequality constraints. Note that equation (6.96) is just the constraint condition (6.91). Equation (6.98) is called the *complementarity condition*. For the case where either λ_i or d_i are zero for each of the constraints $i = 1, 2, \ldots, p$, but never both, the term *strict complementarity* is used. As an example of an inequality-constrained optimization problem, consider the three-dimensional case with a spherical objective function (6.65) that was considered above, but now with a single inequality constraint

$$d \equiv u_1^2 - u_3 \leq 0 \ . \tag{6.99}$$

Assuming that the constraint is active, the modified objective function is

$$\bar{\mathcal{J}}(\mathbf{u}) \equiv \mathcal{J}(\mathbf{u}) + \lambda d = (u_1 - 2)^2 + u_2^2 + u_3^2 + \lambda\left(u_1^2 - u_3\right) \ . \tag{6.100}$$

Taking the derivative with respect to **u** leads to the four equations

$$\frac{\partial \bar{\mathcal{J}}}{\partial u_1} \equiv 2(1+\lambda)u_1 - 4 = 0 \ , \tag{6.101}$$

$$\frac{\partial \bar{\mathcal{J}}}{\partial u_2} \equiv 2u_2 = 0 \ , \tag{6.102}$$

$$\frac{\partial \bar{\mathcal{J}}}{\partial u_3} \equiv 2u_3 - \lambda = 0 \ , \tag{6.103}$$

$$\frac{\partial \bar{\mathcal{J}}}{\partial \lambda} \equiv u_1^2 - u_3 = 0 \ , \tag{6.104}$$

which can be solved[‡] to give

$$\left[u_1^0 \quad u_2^0 \quad u_3^0 \quad \lambda^0\right]^T \approx \left[0.84 \quad 0 \quad 0.70 \quad 1.39\right]^T \text{ and } \mathcal{J}(\mathbf{u}^0) \approx 1.84 \ . \tag{6.105, 6.106}$$

The Lagrange multiplier is nonzero and positive, which implies that the inequality constraint is indeed strongly active; see also Figure 6.15. In the more general case of multiple inequality constraints it is usually necessary to perform a trial and error procedure to establish which contraints are active, and many search strategies have been developed for that purpose, for a description of which we refer to the literature listed in Section 6.1.

[‡] The solution for u_2^0 follows trivially from the second equation. The other three equations can be combined to give a cubic equation for u_1^0 which is most easily solved numerically, e.g. using the MATLAB command `roots`.

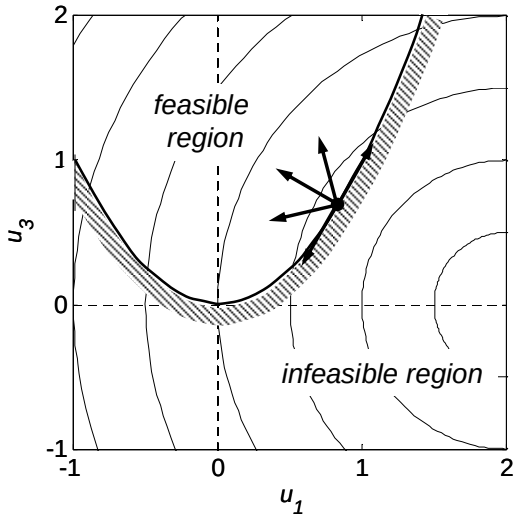


Figure 6.15: Cross section through a spherical objective function with a single inequality constraint in the form of a parabolic cyclinder perpendicular to the plane of the paper. The solid dot indicates the stationary point, and the arrows indicate examples of feasible perturbation directions in the plane of the paper. The two arrows tangent to the constraint correspond to binding perturbations (up to first order); the other three arrows to non-binding perturbations.

*6.4 Constrained optimization – optional topics**

6.4.1 Sufficient optimality conditions*

Practical relevance

For realistic optimization problems in reservoir engineering it is usually not computationally feasible to establish the second-order optimality conditions which are required to prove optimality of a stationary point. However, for completeness sake and to further illustrate the theoretical aspects of flooding optimization we will briefly discuss the sufficient conditions for constrained optimization.

*Single linear equality constraint**

In case of a single linear equality constraint, as in our simple example in Sections 6.3 and 6.3.2 above, the second-order sufficient condition can be obtained by considering the second-order directional derivative (also known as the *projected Hessian*) of the objective function,

$$\left.\frac{\partial^2 \mathcal{J}}{\partial \mathbf{u}^2}\right|_{\mathbf{v}} \triangleq \mathbf{v}^T \frac{\partial^2 \mathcal{J}}{\partial \mathbf{u}^2}\mathbf{v} \equiv \begin{bmatrix} v_1 & v_2 \end{bmatrix} \begin{bmatrix} \dfrac{\partial^2 \mathcal{J}}{\partial u_1^2} & \dfrac{\partial^2 \mathcal{J}}{\partial u_1 \partial u_2} \\ \dfrac{\partial^2 \mathcal{J}}{\partial u_2 \partial u_1} & \dfrac{\partial^2 \mathcal{J}}{\partial u_2^2} \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}, \tag{6.107}$$

which should be positive definite in the stationary point. Filling in the appropriate values gives

$$\begin{bmatrix} v_1^0 & v_2^0 \end{bmatrix} \begin{bmatrix} \dfrac{\partial^2 \mathcal{J}}{\partial u_1^2} & \dfrac{\partial^2 \mathcal{J}}{\partial u_1 \partial u_2} \\ \dfrac{\partial^2 \mathcal{J}}{\partial u_2 \partial u_1} & \dfrac{\partial^2 \mathcal{J}}{\partial u_2^2} \end{bmatrix}_{\mathbf{u}=\mathbf{u}^0} \begin{bmatrix} v_1^0 \\ v_2^0 \end{bmatrix} = \begin{bmatrix} 1 & -1 \end{bmatrix} \begin{bmatrix} 4 & 0 \\ 0 & 4 \end{bmatrix} \begin{bmatrix} 1 \\ -1 \end{bmatrix} = 8 \ , \tag{6.108}$$

which is indeed positive, confirming again that the stationary point is a minimum[§]. In this example the Hessian in the minimum is also positive definite. However, in general this is not necessarily the case and the projected Hessian may be positive definite whereas the Hessian itself is indefinite.

*Single nonlinear equality constraint**

In case of a nonlinear equality constraint the situation becomes more complex and we have to consider the *projected Hessian of the modified objective function*[†]:

$$\left.\frac{\partial^2 \bar{\mathcal{J}}}{\partial \mathbf{u}^2}\right|_{\mathbf{v}} \triangleq \mathbf{v}^T \frac{\partial^2 \bar{\mathcal{J}}}{\partial \mathbf{u}^2} \mathbf{v} \equiv \begin{bmatrix} v_1 & v_2 \end{bmatrix} \begin{bmatrix} \dfrac{\partial^2 \bar{\mathcal{J}}}{\partial u_1^2} & \dfrac{\partial^2 \bar{\mathcal{J}}}{\partial u_1 \partial u_2} \\ \dfrac{\partial^2 \bar{\mathcal{J}}}{\partial u_2 \partial u_1} & \dfrac{\partial^2 \bar{\mathcal{J}}}{\partial u_2^2} \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} . \tag{6.109}$$

Consider e.g. the simple two-dimensional minimization problem of Figure 6.13, but now with a nonlinear constraint defined as

$$c(\mathbf{u}) \triangleq 2u_1^2 + u_2 - 0.5 = 0 \ , \tag{6.110}$$

which is a parabola with its apex in $(u_1, u_2) = (0, 0.5)$; see Figure 6.16 (top left). The modified objective function is now

$$\bar{\mathcal{J}}(\mathbf{u}, \lambda) \triangleq \mathcal{J}(\mathbf{u}) + \lambda c(\mathbf{u}) = 2\left(u_1^2 + u_2^2\right) + \lambda\left(2u_1^2 + u_2 - 0.5\right), \tag{6.111}$$

from which we obtain the necessary conditions

$$\frac{\partial \bar{\mathcal{J}}}{\partial u_1} \equiv 4(1+\lambda)u_1 = 0 \tag{6.112}$$

$$\frac{\partial \bar{\mathcal{J}}}{\partial u_2} \equiv 4u_2 + \lambda = 0 \tag{6.113}$$

$$\frac{\partial \bar{\mathcal{J}}}{\partial \lambda} \equiv 2u_1^2 + u_2 - 0.5 = 0 \tag{6.114}$$

which results in three stationary points:

[§] In this simple example with a single constraint the projected Hessian becomes a scalar, and therefore "positive definite" is equal to "positive". In the case of multiple constraints the projected Hessian becomes a matrix, as will be discussed below.

[†] Often called the *(projected) Hessian of the Lagrangian*; see also the footnote on p. 128.

$$\begin{bmatrix} \mathbf{u}_1^0 \\ \lambda_1^0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0.5 \\ -2 \end{bmatrix}, \quad \begin{bmatrix} \mathbf{u}_{2,3}^0 \\ \lambda_{2,3}^0 \end{bmatrix} = \begin{bmatrix} \frac{\pm 1}{2\sqrt{2}} \\ \frac{1}{4} \\ -1 \\ \end{bmatrix} \approx \begin{bmatrix} \pm 0.35 \\ 0.25 \\ -1 \end{bmatrix}, \tag{6.115, 6.116}$$

with corresponding objective function values

$$\mathcal{J}\left(\mathbf{u}_1^0\right) = \frac{1}{2} \text{ and } \mathcal{J}\left(\mathbf{u}_{2,3}^0\right) = \frac{3}{8}. \tag{6.117, 6.118}$$

To compute the second-order optimality conditions we need the unit vector **v** tangent to the constraint which is perpendicular to gradient of the constraint function defined as

$$\nabla c \triangleq \begin{bmatrix} \frac{\partial c}{\partial u_1} \\ \frac{\partial c}{\partial u_2} \end{bmatrix} = \begin{bmatrix} 4u_1 \\ 1 \end{bmatrix}. \tag{6.119}$$

where the equality is specific to our example. In the three stationary points we find

$$\left(\nabla c\right)_1^0 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \left(\nabla c\right)_{2,3}^0 = \begin{bmatrix} \pm\sqrt{2} \\ 1 \end{bmatrix}. \tag{6.120, 6.121}$$

Using the relationship

$$\left(\nabla c\right)^T \mathbf{v} = 0 \ , \tag{6.122}$$

and the constraint

$$\left|\mathbf{v}\right| = 1 \ , \tag{6.123}$$

we obtain the unit tangent vectors **v** as

$$\mathbf{v} = \begin{bmatrix} 1 \\ -4u_1 \end{bmatrix} \frac{1}{\sqrt{1+16u_1^2}} \ , \tag{6.124}$$

which gives us three vectors $\mathbf{v}^0$ in the three stationary points:

$$\mathbf{v}_1^0 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \mathbf{v}_{2,3}^0 = \begin{bmatrix} \frac{1}{\sqrt{3}} \\ \mp\sqrt{\frac{2}{3}} \end{bmatrix}. \tag{6.125, 6.126}$$

The Hessian $\bar{\mathbf{H}}$ of the modified objective function is given by

$$\bar{\mathbf{H}} \triangleq \frac{\partial^2 \bar{\mathcal{J}}}{\partial \mathbf{u}^2} = \begin{bmatrix} 4\left(1+\lambda\right) & 0 \\ 0 & 4 \end{bmatrix}, \tag{6.127}$$

such that we find

$$\bar{\mathbf{H}}_1^0 = \begin{bmatrix} -4 & 0 \\ 0 & 4 \end{bmatrix} \text{ and } \bar{\mathbf{H}}_{2,3}^0 = \begin{bmatrix} 0 & 0 \\ 0 & 4 \end{bmatrix} \tag{6.128, 6.129}$$

for the three stationary points respectively. The corresponding projected Hessians are therefore given by

$$\left(\mathbf{v}_1^0\right)^T \bar{\mathbf{H}}_1^0 \mathbf{v}_1^0 = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} -4 & 0 \\ 0 & 4 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = -4 \ , \tag{6.130}$$

$$\left(\mathbf{v}_2^0\right)^T \bar{\mathbf{H}}_2^0 \mathbf{v}_2^0 = \begin{bmatrix} \frac{1}{\sqrt{3}} & -\sqrt{\frac{2}{3}} \end{bmatrix} \begin{bmatrix} 0 & 0 \\ 0 & 4 \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{3}} \\ -\sqrt{\frac{2}{3}} \end{bmatrix} = \frac{8}{3} \ , \tag{6.131}$$

$$\left(\mathbf{v}_3^0\right)^T \bar{\mathbf{H}}_3^0 \mathbf{v}_3^0 = \begin{bmatrix} \frac{1}{\sqrt{3}} & \sqrt{\frac{2}{3}} \end{bmatrix} \begin{bmatrix} 0 & 0 \\ 0 & 4 \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{3}} \\ \sqrt{\frac{2}{3}} \end{bmatrix} = \frac{8}{3} \ , \tag{6.132}$$

from which we conclude that the two symmetric stationary points $(u_1^0, u_2^0)_{2,3} = (\pm 0.35, 0.25)$ are indeed mininima, whereas the point $(u_1^0, u_2^0)_1 = (0, 0.5)$ is a maximum. However, the Hessian $\mathbf{H}$ of the original objective function is given by

$$\mathbf{H} \triangleq \frac{\partial^2 \mathcal{J}}{\partial \mathbf{u}^2} = \begin{bmatrix} 4 & 0 \\ 0 & 4 \end{bmatrix} , \tag{6.133}$$

such that we find for the projected values[‡]:

$$\left(\mathbf{v}_1^0\right)^T \mathbf{H}_1^0 \mathbf{v}_1^0 = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} 4 & 0 \\ 0 & 4 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = 4 \ , \tag{6.134}$$

$$\left(\mathbf{v}_2^0\right)^T \mathbf{H}_2^0 \mathbf{v}_2^0 = \begin{bmatrix} \frac{1}{\sqrt{3}} & -\sqrt{\frac{2}{3}} \end{bmatrix} \begin{bmatrix} 4 & 0 \\ 0 & 4 \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{3}} \\ -\sqrt{\frac{2}{3}} \end{bmatrix} = 4 \ , \tag{6.135}$$

$$\left(\mathbf{v}_3^0\right)^T \mathbf{H}_3^0 \mathbf{v}_3^0 = \begin{bmatrix} \frac{1}{\sqrt{3}} & \sqrt{\frac{2}{3}} \end{bmatrix} \begin{bmatrix} 4 & 0 \\ 0 & 4 \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{3}} \\ \sqrt{\frac{2}{3}} \end{bmatrix} = 4 \ , \tag{6.136}$$

[‡] In fact, the projected Hessian is equal to 4 along the *entire* constraint.

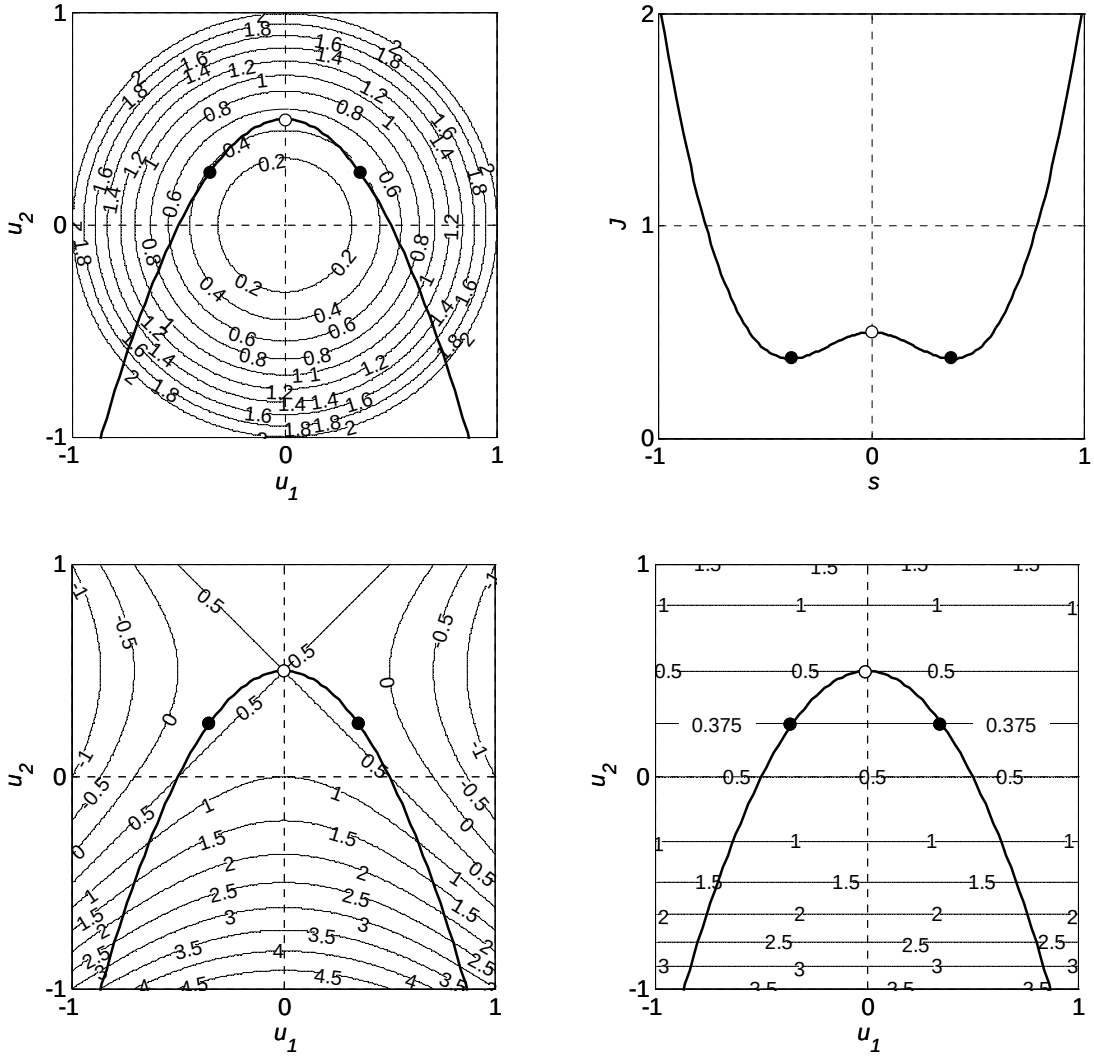


Figure 6.16: Nonlinear constrained optimization. Top left: Contour lines of nonlinear objective function (6.13), together with nonlinear constraint (6.110). The three dots indicate the constrained stationary points where the constraint curve is tangent to the contour lines. Only the solid dots correspond to a minimum. Top right: Magnitude of the objective function $\mathcal{J}$ as a function of position s along the constraint. Bottom left: Contour lines of modified objective function (6.111) for $\lambda = -2$, together with nonlinear constraint (6.110). Bottom right: Contour lines of modified objective function (6.111) for $\lambda = -1$, together with nonlinear constraint (6.110).

which, incorrectly, suggests that all three stationary point are a minimum. The correct result can be simply verified by inspection of the contour lines crossed by the constraint in Figure 6.16 (top left): starting at the left-bottom corner and following the constraint we first descend until reaching the first minimum at the left solid dot, then climb towards the maximum at the open dot, descend again to the second minimum at the right solid dot, and then finally climb back up again towards the right bottom corner. This behaviour is visualized in Figure 6.16 (top right) in which the objective function value is plotted as a function of coordinate *s* along

the constraint[†]. Note that in all three stationary points the contour lines are tangent to the constraint, in line with the Lagrange multiplier definition in equation (6.30). Figure 6.16 (bottom left and bottom right) also displays the modified objective function $\bar{\mathcal{J}}(\mathbf{u},\lambda)$ for values of λ equal to −2 and −1 respectively. It can be seen that the values of the modified objective function along the constraint, and therefore also in the three stationary points, are identical to those of the original objective function $\mathcal{J}(\mathbf{u})$ (top left) but that they are otherwise completely different. Interpretation of the curvature of the constraints as represented by the projected Hessians of (modified) objective functions is not trivial, and should be based on evaluating expressions (6.107) and (6.109) rather than on inspection of the graphs.

*Multiple nonlinear equality constraints**

In case of multiple nonlinear equality constraints we can write the equivalent version of equation (6.109) as

$$\left.\frac{\partial^2 \bar{\mathcal{J}}}{\partial \mathbf{u}^2}\right|_{\mathrm{V}} \triangleq \mathbf{V}^T \frac{\partial^2 \bar{\mathcal{J}}}{\partial \mathbf{u}^2} \mathbf{V} \ , \tag{6.137}$$

where the columns of **V** are the unit vectors of the tangents at the feasible arcs[‡], just as in equation (6.60). A second-order sufficient condition for a minimum can be obtained if the constraint qualifications are met, which is the case if e.g. the constraint tangent matrix **A**, as defined in equation (6.46), has full row rank. The sufficient condition is then the same as was discussed for the single-constraint case, i.e. the requirement that the projected Hessian of the modified objective function, as defined in equation (6.137), is positive definite in the stationary point.

*Nonlinear inequality constraints**

Finally, we arrive at the sufficient conditions for the case of one or more linear or nonlinear inequality constraints. In this case we need to consider the constraint tangent matrix $\hat{\mathbf{A}}$ for the active constraints as defined in equation (6.93). Considering again the case that the constraint qualifications are met, it is required, as was the case for (nonlinear) equality constraints, that the projected Hessian of the (modified) objective function is positive definite. Moreover, just as for the first-order necessary conditions which were considered in Section (6.3.5), we need additional restrictions on the sign of the Lagrange multipliers. However, unlike for the first-order necessary conditions it is not sufficient to just require that the Lagrange multipliers for the active constraints are non-negative. That is because a weakly active constraint, which therefore has a zero Lagrange multiplier, may be present in a stationary point where the projected Hessian is positive definite, but where the objective function has a negative curvature. In that case it is still possible to define feasible perturbations that lead to a decrease in the objective function. This can be illustrated by considering the non-convex objective function

$$\mathcal{J}(\mathbf{u}) = u_1^2 - u_2^2 \ , \tag{6.138}$$

with an inequality constraint

[†] The coordinate s is given by $s(\hat{u}_1) \triangleq \int_0^{\hat{u}_1} \sqrt{[1+(du_2/du_1)^2]}du_1 = \hat{u}_1\sqrt{0.25+\hat{u}_1^2} + 0.25\ln[(\hat{u}_1+\sqrt{0.25+\hat{u}_1^2})/\sqrt{0.25}]$ where we made use of the fact that constraint (6.110) can be expressed explicitly in terms of u_1 as $u_2 = 0.5 - u_1^2$.

[‡] More in general, the columns of **V** may be any set of vectors that form a basis for the null space of **A**.

$$d_1 \equiv u_2 \le 0 \ ; \tag{6.139}$$

see Figure 6.17. The modified objective function becomes

$$\bar{\mathcal{J}}(\mathbf{u}) = u_1^2 - u_2^2 + \lambda u_2 \ , \tag{6.140}$$

from which we obtain

$$\frac{\partial \bar{\mathcal{J}}}{\partial u_1} \equiv 2u_1 = 0 \tag{6.141}$$

$$\frac{\partial \bar{\mathcal{J}}}{\partial u_2} \equiv 2u_2 + \lambda = 0 \tag{6.142}$$

$$\frac{\partial \bar{\mathcal{J}}}{\partial \lambda} \equiv u_2 = 0 \tag{6.143}$$

resulting in the stationary point

$$\begin{bmatrix} \mathbf{u}^0 \\ \lambda^0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \ . \tag{6.144}$$

The Hessian of the objective function follows as†

$$\mathbf{H} \equiv \frac{\partial^2 \mathcal{J}}{\partial u^2} = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}, \tag{6.145}$$

and the projected Hessian in the stationary point as

$$\left(\mathbf{v}^0\right)^T \mathbf{H}^0 \mathbf{v}^0 = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = 2 \, , \tag{6.146}$$

which is positive, suggesting a minimum. However, as can be seen in Figure 6.17, there are feasible perturbations that result in a decrease of the objective function, and the stationary point is therefore not a minimum. Sufficient conditions for a stationary point to be a minimum in case of linear or nonlinear inequality constraints are therefore that in the stationary point:

1) The KKT necessary conditions (6.95) to (6.98) are met.
2) The constraint qualifications are met, e.g. in the form of matrix $\hat{\mathbf{A}}$ as defined in equation (6.93) having independent rows.
3) The projected Hessian of the modified objective function is positive definite, i.e.

$$\hat{\bar{\mathbf{H}}}^0 \triangleq \left. \frac{\partial^2 \bar{\mathcal{J}}}{\partial \mathbf{u}^2} \right|_{\hat{\mathbf{V}}, \mathbf{u}=\mathbf{u}^0} \equiv \hat{\mathbf{V}}^T \left. \frac{\partial^2 \bar{\mathcal{J}}}{\partial \mathbf{u}^2} \hat{\mathbf{V}} \right|_{\mathbf{u}=\mathbf{u}^0} \ge 0 \ , \tag{6.147}$$

where the columns of $\hat{\mathbf{V}}$ are the unit vectors of the tangents to the feasible arcs for the active constraints†.

† Because we consider a linear inequality constraint, the Hessian of the objective function is identical to the Hessian of the modified objective function.

4) The active contraints are strongly active, i.e. they have non-zero Lagrange multipliers. The latter condition implies that the complementary condition (6.98) should hold strictly.

Alternatively, if weakly active constraints do occur in the stationary point, we may replace the second-order condition (6.147) by

$$\hat{\bar{\mathbf{H}}}^{0+} \triangleq \left.\frac{\partial^2 \bar{\mathcal{J}}}{\partial \mathbf{u}^2}\right|_{\hat{\mathbf{V}}^+,\mathbf{u}=\mathbf{u}^0} \equiv \left(\hat{\mathbf{V}}^+\right)^T \left.\frac{\partial^2 \bar{\mathcal{J}}}{\partial \mathbf{u}^2}\hat{\mathbf{V}}^+\right|_{\mathbf{u}=\mathbf{u}^0} \geq 0 \ , \tag{6.148}$$

where the columns of $\hat{\mathbf{V}}^+$ are now the unit vectors of the tangents to the feasible arcs for the *strongly* active constraints only. At the same time we can now relax the complimentarity condition (6.98) to hold non-strictly.

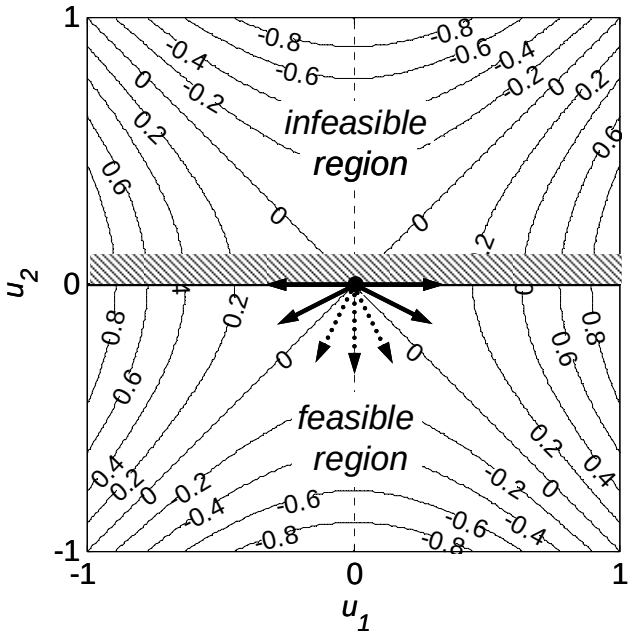


Figure 6.17: Contour lines of non-convex objective function (6.138) with linear inequality constraint (6.139). The dot indicates the stationary point, which is a saddle point, because the objective function has positive and negative curvatures along the x- and y-axes respectively. The arrows indicate examples of feasible perturbation directions. The solid arrows result in an increase of the objective function value, whereast the dotted arrows result in a decrease.

6.4.2 Saddle points for u and λ*

In Sections (6.3.2) and (6.4) we derived the first- and second-order sufficient optimality conditions for an equality-constrained minimum. We presented a two-dimensional numerical example to illustrate that the stationary points $(u_1^0, u_2^0)_{2,3} = (\pm 0.35, 0.25)$ correspond to a minimum of the modified objective function $\bar{\mathcal{J}}$. Indeed, if we consider admissible variations δu_1 and δu_2 i.e. variations that fulfill condition (6.35), the value of $\bar{\mathcal{J}}$ increases as can be verified in Figure 6.16 (top right). However $\bar{\mathcal{J}}$ is also a function of the Lagrange multiplier λ, and we can therefore also consider the effect of variations $\delta\lambda$. To do so we return to

† Because the admissible variations along the feasible arcs obey the relationship $\hat{\mathbf{A}}\delta\mathbf{u} = \mathbf{0}$, the columns of $\hat{\mathbf{V}}$ may be any set of vectors that form a basis for the null space of $\hat{\mathbf{A}}$.

equations (6.112) to (6.114), solve for u_1 and u_2 but leave λ undetermined, and substitute the results in the definition of $\bar{\mathcal{J}}$ given in equation (6.111) which leads to[†]

$$\bar{\bar{\mathcal{J}}}(\lambda) = \frac{1}{4}\left(1 - \lambda - \frac{1}{2}\lambda^2\right). \tag{6.149}$$

The first and second variations of equation (6.149) are given by

$$\delta\bar{\bar{\mathcal{J}}} = \frac{\partial\bar{\bar{\mathcal{J}}}}{\partial\lambda}\delta\lambda = -\frac{1}{4}(1+\lambda)\delta\lambda \ , \tag{6.150}$$

$$\delta^2\bar{\bar{\mathcal{J}}} = \frac{\partial^2\bar{\bar{\mathcal{J}}}}{\partial\lambda^2}\delta\lambda^2 = -\frac{1}{4}\delta\lambda^2 \ , \tag{6.151}$$

Requiring stationarity of $\delta\bar{\bar{\mathcal{J}}}$ for arbitrary variations $\delta\lambda$ leads to $\lambda^0 = -1$ in line with the earlier results for $\lambda^0_{2,3}$ from equation (6.116). Furthermore, equation (6.151) shows that always $\delta^2\bar{\bar{\mathcal{J}}} < 0$ which implies that the stationary points are *maxima* with respect to variations in λ. Apparently the stationary points $\mathbf{u}^0_{2,3}$ of the modified objective function $\bar{\mathcal{J}}$ are saddle points, i.e. minima with respect to admissible variations $\delta\mathbf{u}$ and maxima with respect to variations $\delta\lambda$. It can be shown that the occurrence of a saddle point is always the case for constrained minimization problems, and the unconstrained optimization problem

$$\max_{\lambda} \bar{\bar{\mathcal{J}}}(\lambda) \ , \tag{6.152}$$

is known as the *dual* to the constrained optimization problem

$$\min_{\mathbf{u}} \mathcal{J}(\mathbf{u}) \ \text{ subject to } \ c(\mathbf{u}) = 0 \ , \tag{6.153}$$

which is known as the *primal* problem. Visualizing the saddle point for our simple case cannot be done easily because that requires plotting the relationship between four variables: u_1, u_2, λ and J. We therefore consider another example with a single control variable u:

$$\min_{u} \mathcal{J}(u), \tag{6.154}$$

where

$$\mathcal{J}(u) = u^2 \ , \tag{6.155}$$

subject to

$$c(u) \equiv \exp\left(-\frac{1}{2} - u\right) - 1 = 0 \ . \tag{6.156}$$

Inspection of the constraint equation (6.156) reveals that the solution to this problem is simply $u^0 = -1/2$. The formal solution with the aid of a Lagrange multiplier follows as

$$\bar{\mathcal{J}}(u,\lambda) = u^2 + \lambda\left[\exp\left(-\frac{1}{2} - u\right) - 1\right] \ , \tag{6.157}$$

[†] Note that we chose the results corresponding to the minima $\mathbf{u}^0_{2,3}$ for which $u_1^0 = \pm 1/(2\sqrt{2})$ and $u_2^0 = -\lambda/4$.

$$\delta\bar{\mathcal{J}} = \left[2u - \lambda\exp\left(-\frac{1}{2}-u\right)\right]\delta u + \left[\exp\left(-\frac{1}{2}-u\right)-1\right]\delta\lambda \ , \tag{6.158}$$

$$\frac{\partial\bar{\mathcal{J}}}{\partial u} \equiv 2u - \lambda\exp\left(-\frac{1}{2}-u\right) = 0 \ , \tag{6.159}$$

$$\frac{\partial\bar{\mathcal{J}}}{\partial\lambda} \equiv \exp\left(-\frac{1}{2}-u\right) - 1 = 0 \ , \tag{6.160}$$

$$\begin{bmatrix} u^0 \\ \lambda^0 \end{bmatrix} = \begin{bmatrix} -\frac{1}{2} \\ -1 \end{bmatrix} . \tag{6.161}$$

The second variation leads to

$$\begin{aligned} \delta^2\bar{\mathcal{J}} &= \begin{bmatrix}\delta u & \delta\lambda\end{bmatrix}\begin{bmatrix} \frac{\partial^2\bar{\mathcal{J}}}{\partial u^2} & \frac{\partial^2\bar{\mathcal{J}}}{\partial u\partial\lambda} \\ \frac{\partial^2\bar{\mathcal{J}}}{\partial\lambda\partial u} & \frac{\partial^2\bar{\mathcal{J}}}{\partial\lambda^2} \end{bmatrix}\begin{bmatrix}\delta u \\ \delta\lambda\end{bmatrix} \\ &= \begin{bmatrix}\delta u & \delta\lambda\end{bmatrix}\begin{bmatrix} 2+\lambda\exp\left(-\frac{1}{2}-u\right) & -\exp\left(-\frac{1}{2}-u\right) \\ -\exp\left(-\frac{1}{2}-u\right) & 0 \end{bmatrix}\begin{bmatrix}\delta u \\ \delta\lambda\end{bmatrix}. \end{aligned} \tag{6.162}$$

The eigenvalues[†] of the Hessian for $[u\ \lambda]^T = [u^0\ \lambda^0]^T$ in equation (6.162) are $1/2 \pm \sqrt{5}/2 \approx 0.500 \pm 1.118$, i.e. one is positive and the other negative, which indeed corresponds to a saddle point. From the corresponding eigenvectors it follows that the principal directions are at angles of approximately 32 and -58 degrees with respect to the u coordinate; see Figure 6.18, bottom left. The bottom right picture in Figure 6.18 displays the objective function $\mathcal{J}$ and the modified objective function $\bar{\mathcal{J}}$, both as a function of u, and it can be seen that (local) minimum of $\bar{\mathcal{J}}(u)$ indeed occurs at $u^0 = -\frac{1}{2}$. In this particular case the maximum value of $\bar{\bar{\mathcal{J}}}(\lambda)$ does not occur for a single value of λ but is simply the entire line defined by $u^0 = 1$.

[†] See equation A.67 in Appendix A to compute the eigenvalues of a 2×2 matrix.

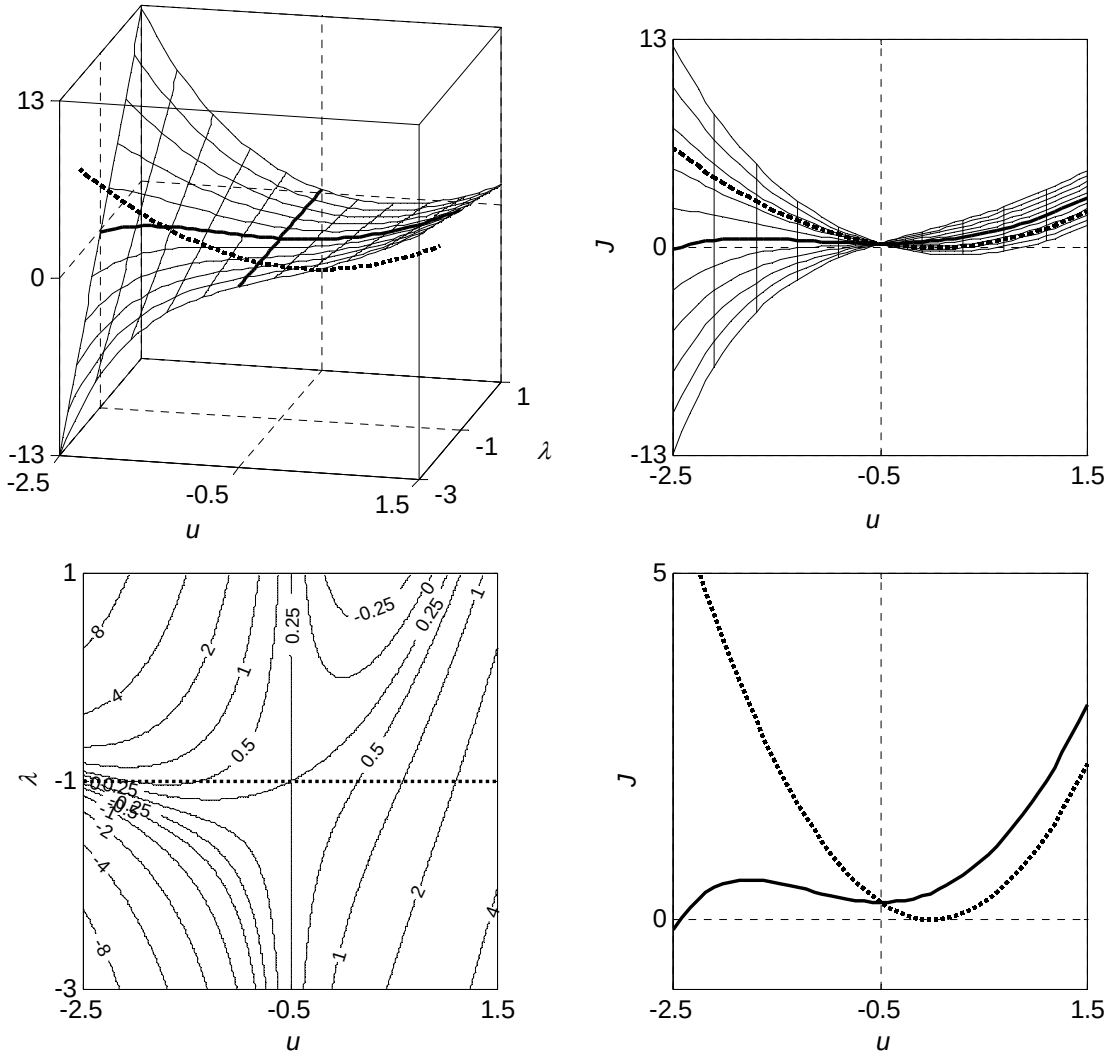


Figure 6.18: Saddle point. Top left: View in perspective. Top right: View in the λ direction. Bottom left: Contour lines. Bottom right: Cross-section at $\lambda = -1$ with expanded vertical scale; dotted line: objective function $\mathcal{J}$; solid line: modified objective function $\bar{\mathcal{J}}$.

6.4.3 Augmented Lagrangian*

Until now we considered simple examples where the sufficient and necessary conditions could be solved in closed form. In case of larger optimization problems, such as those encountered in reservoir engineering, closed-form answers will not be available and we will have to use an iterative approach to search for the minimum. As will be discussed in more detail in Section 6.5, we usually apply some form of gradient-based technique where a new estimate $\mathbf{u}^{i+1}$ can be obtained from the old estimate $\mathbf{u}^i$ by using the gradients $\nabla\bar{\mathcal{J}} \equiv [\partial\bar{\mathcal{J}}/\partial\mathbf{u}]^T$ evaluated at the old point. As can be inferred from Figure 6.18 such an iterative procedure may indeed reach a minimum, as long as $[\partial\bar{\mathcal{J}}/\partial\mathbf{u}]$ is evaluated for the correct value of λ^0. However, the value of λ^0 will usually also have to be determined iteratively and may therefore contain an error ε. In that case, minimizing $\bar{\mathcal{J}}(\mathbf{u}, \lambda^0 + \varepsilon)$ instead of $\bar{\mathcal{J}}(\mathbf{u}, \lambda^0)$ may lead to a step beyond the minimum, causing the iterative procedure to slide down the slope of the saddle rather than to converge. For example, it can be seen in Figure 6.18, bottom left, that for a value of λ slightly smaller than -1 it is no longer possible to find a minimum. In

realistic, large scale problems we may have many local minima and many saddle points (in many dimensions), and the sensitivity to wrong estimates of λ^0 will remain present. One way to reduce this effect is through the definition of an additional quadratic *penalty term* in the modified objective function to change the saddle problem to a (local) minimization problem. If we consider a situation with multiple constraints, this can be written as

$$\bar{\mathcal{J}}_a(\mathbf{u},\boldsymbol{\lambda}) \triangleq \mathcal{J}(\mathbf{u}) + \boldsymbol{\lambda}^T \mathbf{c}(\mathbf{u}) + \frac{\theta}{2}\mathbf{c}^T(\mathbf{u})\mathbf{c}(\mathbf{u}) \ , \tag{6.163}$$

where θ is a positive scalar, known as the penalty parameter, and where the quadratic term is formed by the squared constraint equation which will of course become zero in the constrained minimum. As mentioned in the footnote on page 127, the modified objective function is often called the Lagrangian, and therefore the *augmented modified objective function* (6.163) is often called the *augmented Lagrangian*. For the example with a single control variable *u*, examined in Section 6.4.2, the augmented Lagrangian becomes

$$\bar{\mathcal{J}}_a(u,\lambda) = u^2 + \lambda\left[\exp\left(-\frac{1}{2}-u\right)-1\right] + \frac{\theta}{2}\left[\exp\left(-\frac{1}{2}-u\right)-1\right]^2 \ . \tag{6.164}$$

Figure 6.19 shows various plots of this function for a penalty variable $\theta = 2$. In comparison with Figure 6.18 the 'robustness' against errors ε in estimates of λ^0 has improved considerably and for all values of λ considered in the figure, the stationary point of $\bar{\mathcal{J}}(u,\lambda^0+\varepsilon)$ is a true minimum rather than a saddle point. Moreover, for all values of *u* displayed in the figure, the minimum is now the only minimum. Apparently the use of the augmented Lagrangian has increased the *domain of attraction* of the stationary point $(-\frac{1}{2},-1)$, both in the λ and the *u* direction. A problem associated with the use of the augmented Lagrangian in realistic optimization problems is the difficulty to determine the optimal value of θ. A too small value will not increase the domain of attraction sufficiently, but a too large value may slow down the iterative optimization procedure or lead to numerical problems (ill-conditioning).

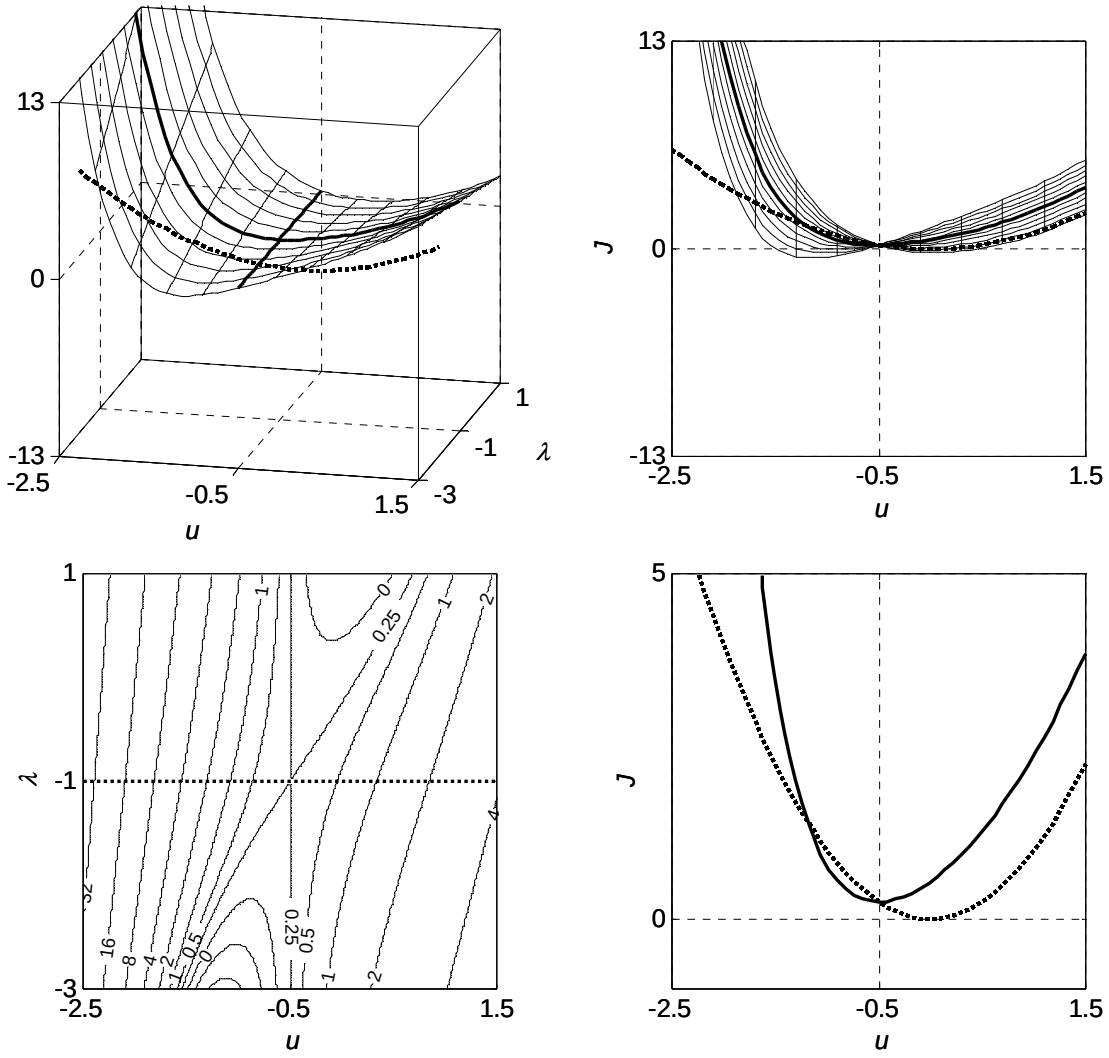


Figure 6.19: Augmented Lagrangian. Compare with Figure 6.18. Top left: View in perspective. Top right: View in the λ direction. Bottom left: Contour lines. Bottom right: Cross-section at $\lambda = -1$ with expanded vertical scale; dotted line: objective function $\mathcal{J}$; solid line: modified augmented objective function (augmented Lagrangian) $\bar{\mathcal{J}}_a$ for $\theta = 2$.

6.5 Numerical optimization

6.5.1 Gradient-based and gradient-free methods

The optimization problems in the previous section could mostly be solved directly. This involved computation of the first variation of the objective function in closed form, setting the result equal to zero, and then solving for the optimal input variables. For large scale optimization problem, like the ones we encounter in reservoir simulation, this is almost never possible, and we need a numerical, usually iterative, procedure to compute the optimum. Many numerical optimization methods exist and we refer again to the textbooks of Gill et al. (1986), Fletcher (1987), Rao (1996), Bonnans et al. (2003) or Nocedal and Wright (2006) for overviews. A major distinction between the different methods is obtained by deviding them in those that use gradient information, and those that do not. The classic metaphor for maximizing an objective function is climbing to the top of a hill while being surrounded by fog. The simplest conceivable algorithm to automate that activity is the *steepest ascent*

method, where we take steps towards the top such that every step points in steepest upward direction; see Figure 6.20. This illustrates the two important elements in gradient-based numerical optimization: determining the search direction and determining the step size at each step. A key feature of gradient-based methods is their tendency to find a local optimimum, rather than the global one, because once they have reached the top of a hill there is usually no mechanism to let them step towards another, higher top in the landscape. Searching for a global optimum with a gradient-based method therefore requires additional feaures such as e.g. starting from many initial guesses, and occasionally randomly perturbing the search direction. This is as opposed to most gradient-free methods which directly aim at finding the global optimum, usually also with some form of random sampling of the objective function value. The price to pay for searching for the global optimum, whichever method used, is the need to perform many *function evaluations*, i.e. computations of the objective function value. In most optimization problems in reservoir simulation a function evaluation is equivalent to performing a full simulation and the use of global search techniques is therefore computationally very expensive. Because the number of function evaluations used in gradient-free methods is typically an order of magnitude larger than the number used in gradient-based ones, the application of gradient-free methods in reservoir flooding optimization is restricted to cases with a small number of input parameters, say up to a few tens, unless massively parallel computing is used; see Echeverria et al. (2010). In this text we will therefore only address gradient-based methods.

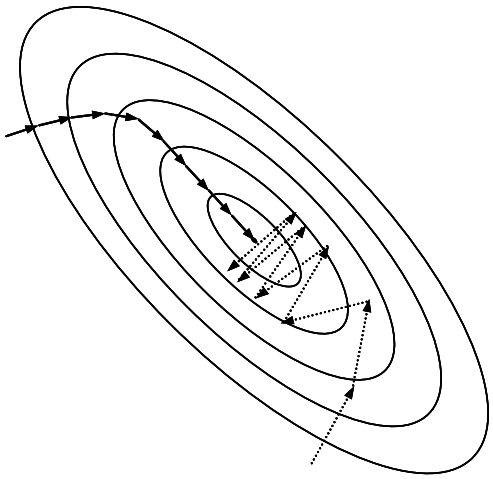


Figure 6.20: Contour lines of an objective function with two steepest ascent aproaches to the maximum using two different, fixed, step sizes. The solid line represents an effective ascent (the maximum is in this case, with some luck, reached exactly) which is also quite efficient (the path is going reasonably directly to the top); the dotted line represents a less effective ascent (the maximumis never reached exactly) which is also not very efficient (the trajectory 'zig zags').

6.5.2 Search direction

Newton-Raphson

Consider an unconstrained optimization problem with a convex objective function $\mathcal{J}(\mathbf{x})$. As discussed in Section 6.2.1, the necessary first-order condition for an optimum is

$$\left.\frac{\partial \mathcal{J}}{\partial \mathbf{x}}\right|_{\mathbf{x}=\mathbf{x}^0} = \mathbf{0}^T \ . \tag{6.165}$$

Suppose that we are in a point $\mathbf{x}^*$ away from the optimum $\mathbf{x}^0$. Equation (6.165) can then be approximated in $\mathbf{x}^*$ with the aid of a first-order Taylor expansion:

$$\mathbf{0}^T \approx \left.\frac{\partial \mathcal{J}}{\partial \mathbf{x}}\right|_{\mathbf{x}=\mathbf{x}^*} + \left(\mathbf{x}^0 - \mathbf{x}^*\right)^T \left.\frac{\partial^2 \mathcal{J}}{\partial \mathbf{x}^2}\right|_{\mathbf{x}=\mathbf{x}^*} . \tag{6.166}$$

From this expression we can derive that

$$\left(\mathbf{x}^0\right)^T = \left(\mathbf{x}^*\right)^T - \left.\frac{\partial \mathcal{J}}{\partial \mathbf{x}}\right|_{\mathbf{x}=\mathbf{x}^*} \left(\left.\frac{\partial^2 \mathcal{J}}{\partial \mathbf{x}^2}\right|_{\mathbf{x}=\mathbf{x}^*} \right)^{-1} , \tag{6.167}$$

which is a step of a Newton-Raphson procedure to iteratively approach the optimum. In a more general form this can be written as

$$\mathbf{x}^{i+1} = \mathbf{x}^i \underbrace{- \left(\mathbf{H}^i\right)^{-1} \nabla \mathbf{x}^i}_{\text{search direction}} , \tag{6.168}$$

where $\nabla \mathbf{x} \triangleq (\partial \mathcal{J} / \partial \mathbf{x})^T$ is the gradient, $\mathbf{H} \triangleq (\partial^2 \mathcal{J} / \partial \mathbf{x}^2)^T$ is the (transposed) Hessian and superscript i is the iteration counter. The product $-\mathbf{H}^{-1}\nabla \mathbf{x}$ is now the search direction, while the stepsize is equal to one. Note that in the special case that $\mathcal{J}(\mathbf{x})$ is a quadratic function of $\mathbf{x}$, equation (6.166) is exact and the single Newton-Raphson step (6.167) leads directly to the optimum. However, normally this is not the case and an iterative approach is required until the change in the objective function becomes smaller than a predefined convergence tolerance. As usual, in a numerical implementation the Hessian is not inverted but the Newton-Raphson step is computed through solving a linear system of equations according to

$$\mathbf{H}^i \mathbf{r}^i = -\nabla \mathbf{x}^i \ , \tag{6.169}$$

$$\mathbf{x}^{i+1} = \mathbf{x}^i + \gamma^i \mathbf{r}^i \ , \tag{6.170}$$

where the residual $\mathbf{r}^i \triangleq \mathbf{x}^{i+1} - \mathbf{x}^i$ can be seen to be identical to the search direction. The scalar γ represents the step size, which has been taken equal to one until now, but which may be choose differently as will be discussed below. It is well known that in the neigborhood of the optimum the Newton-Rapshon scheme converges quadratically, provided $\mathcal{J}$ is sufficiently smooth. Away from the optimum, convergence may be slower while in the presence of inflection points or discontinities the procedure may even diverge. Moreover, if $\mathcal{J}$ is nonconvex, the algorithm will converge to a local optimimum which may or may not be equal to the global one.

Steepest ascent

Numerical computation of the Hessian is usually not feasible, and therefore, instead of the true Hessian, an approximation is normally used. In the simplest case the Hessian is replaced by a negative unit matrix which therefore leads to

$$\mathbf{x}^{i+1} = \mathbf{x}^i + \gamma^i \nabla \mathbf{x}^i . \tag{6.171}$$

The search direction is now equal to the gradient $\nabla \mathbf{x}^i$ at the current iterate and the algorithm is therefore known as the steepest ascent method (for maximization problems) or steepest descent method (for minimization problems).

6.5.3 Step size

To be completed.

6.6 References for Chapter 6

Bonnans, J.F., Gilbert, J.C., Lamaréchal, C. and Sagastizábal, C.A., 2003: *Numerical optimization – Theoretical and practical aspects*, Springer, New York.

Echeverrıa Ciaurri, D., Isebor, O.J. and Durlofsky, L.J., 2010: Application of derivative-free methodologies to generally constrained oil production optimization problems. *Procedia Computer Science* **1** (1) 1295–1304. DOI: 10.1016/j.procs.2010.04.145.

Fletcher, R., 1987: *Practical methods of optimization, 2nd ed.*, Wiley, Chichester.

Gill, P.E., Murray, W. and Wright, M.H., 1986: *Practical optimization*, Elsevier Academic Press, London.

Rao, S.S., 1996: *Engineering optimization, 3rd ed.*, Wiley, New York.

Luenberger, D.G. and Ye, Y., 2010: *Linear and nonlinear programming, 3rd ed.* Springer, New York.

Nocedal, J. and Wright, S.J., 2006: *Numerical optimization, 2nd ed.*, Springer, New York.

7 Flooding optimization

7.1 Introduction

In this chapter we will consider methods to optimize the management of a reservoir, and in particular the optimization of injection and production rates of wells during water flooding. We restrict ourselves to open-loop optimization, i.e. to optimization during the design phase; see Figure 7.1. In Chapter 0 we will address closed-loop optimization, which involves a combination of model-based optimization and data assimilation. We will mainly consider optimization for a given configuration of wells, and we will only briefly discuss optimization in a free configuration, e.g. determining the optimal position of sidetracks or infill wells. A lot of production optimization efforts in the E&P industry are focused on short time scales. Instead we focus on life-cycle optimization, i.e. on optimization over the entire producing life of the reservoir with the aim to optimize ultimate recovery or present value. In particular we will consider gradient-based optimization where the derivative information is obtained through the use of an adjoint equation. Adjoint-based optimization techniques were introduced in reservoir engineering during the 1970s for computer-assisted history matching as will be discussed in detail in Chapter 8. About a decade later they also started to be used for the optimization of tertiary recovery processes such as surfactant, polymer, CO_2 or steam flooding; see e.g. Ramirez et al. (1984), Fathi and Ramirez (1984, 1986, 1987), Ramirez (1987), Mehos and Ramirez (1989), Liu et al. (1993), and Liu and Ramirez (1994). The first paper on gradient-based control of water flooding is the one by Asheim (1988), followed by, among others, Virnovsky (1991), Zakirov et al. (1996) and Sudaryanto and Yortsos (2000, 2001). However, industry uptake of these methods was almost absent until the advent of 'smart well' and 'smart fields' technology which caused a revival of interest; see Brouwer and Jansen (2004). Since that time a series of publications have appeared covering various aspects of adjoint-based optimization of reservoir flooding while several large reservoir simulation packages have been equipped with the adjoint functionality.

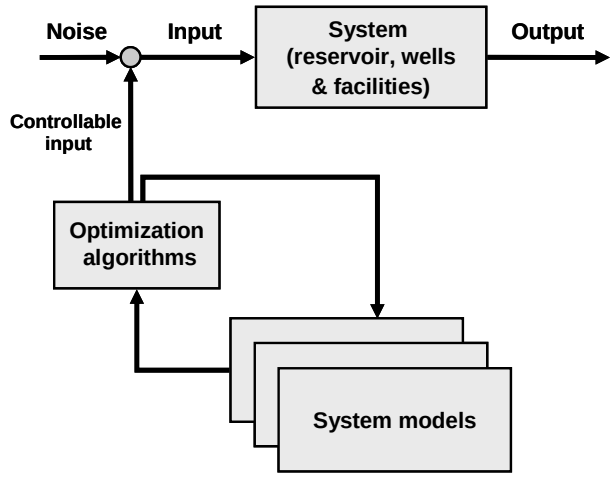


Figure 7.1: Open-loop optimization.

7.2 Problem statement

7.2.1 System model

As a start, we will consider a given configuration of wells, and we will not address the problem of optimizing the well location until later. As indicated in Figure 7.1, the

optimization will be based on one or more system models. We will further restrict our attention to numerical reservoir models, and not address wells and facilities. The optimization variables[§] $\mathbf{u}$, indicated as *controllable input* in Figure 7.1, could then be water injection rates $\mathbf{q}_{w,inj}$ or total production rates $\mathbf{q}_{t,prod}$ in the producers. Alternatively we could choose flowing tubing head pressures $\mathbf{p}_{tf}$, or flowing bottom hole pressures $\mathbf{p}_{wf}$ in the injectors and/or the producers, or valve settings $\boldsymbol{\alpha}$, with elements $0 \leq \alpha_i \leq 1$ representing the dimensionless opening of e.g. wellhead chokes or downhole inflow control valves (ICVs) in a smart well completion. We will start our analysis from an implicitly time-discretized version of the system equations, written in residual form (c.f. equation (4.84)):

$$\mathbf{g}_k\left(\mathbf{u}_k, \mathbf{x}_{k-1}, \mathbf{x}_k\right) = \mathbf{0} \ , \quad k = 1, 2, \ldots, K \ , \tag{7.1}$$

with the appropriate initial conditions

$$\mathbf{x}_0 = \breve{\mathbf{x}}_0 \, . \tag{7.2}$$

In addition we consider an output vector $\mathbf{y}_k$ that is assumed to be a nonlinear function of the input and the states[‡]:

$$\mathbf{j}_k\left(\mathbf{u}_k, \mathbf{x}_k, \mathbf{y}_k\right) = \mathbf{0} \ . \tag{7.3}$$

7.2.2 Objective function

As in any optimization problem, we aim at maximizing or minimizing an objective. For example, the objective could be to maximize a simple net present value (NPV) of the water flooding process. Generally, the objective function $\mathcal{J}$ can be expressed as:

$$\mathcal{J}\left(\mathbf{u}_{1:K}, \mathbf{y}_{1:K}\left(\mathbf{u}_{1:K}\right)\right) = \sum_{k=1}^{K} \mathcal{J}_k\left(\mathbf{u}_k, \mathbf{y}_k\right), \tag{7.4}$$

where $\mathcal{J}_k$ represents the contribution to $\mathcal{J}$ in each time step k (e.g. oil revenues and water injection and production costs during that time interval, where the costs have a negative value)[*]. Note that actually all inputs up to time k may play a role in $\mathcal{J}_k$ in equation (7.4) as can be seen through recursive application of equations (7.1) and (7.3). We should therefore formally write $\mathcal{J}_k(\mathbf{u}_k, \mathbf{y}_k(\mathbf{u}_k, \mathbf{x}_k(\mathbf{u}_{1:k})))$, but to keep the notation tractable we use $\mathcal{J}_k(\mathbf{u}_k, \mathbf{y}_k)$ instead. A typical objective function for flooding optimization is given by

---

[§] Also referred to as *manipulated variables* or *input variables*. The vector $\mathbf{u}$ is referred to as the *input vector*, *control vector* or *decision vector*.

[‡] In equation (4.71) we defined the explicit form (i.e. notation) of the implicitly discretized system equations as $\mathbf{x}_k = \mathbf{f}_k\left(\mathbf{u}_k, \mathbf{x}_{k-1}, \mathbf{x}_k\right)$ and in equation (4.77) we introduced the implicit form (also known as the residual form) as $\mathbf{g}_k\left(\mathbf{u}_k, \mathbf{x}_{k-1}, \mathbf{x}_k\right) \triangleq \mathbf{x}_k - \mathbf{f}_k\left(\mathbf{u}_k, \mathbf{x}_{k-1}, \mathbf{x}_k\right) = \mathbf{0}$. Similarly, in equation (4.59) we introduced the explicit form of the output equation $\mathbf{y}_k = \mathbf{h}_k\left(\mathbf{u}_k, \mathbf{x}_k\right)$, and, following the same notation convention, we therefore now introduce the implicit form as $\mathbf{j}_k\left(\mathbf{u}_k, \mathbf{x}_k, \mathbf{y}_k\right) \triangleq \mathbf{y}_k - \mathbf{h}_k\left(\mathbf{u}_k, \mathbf{x}_k\right) = \mathbf{0}$.

[*] Often the objective function $\mathcal{J}$ is split in two parts: one representing the contribution to $\mathcal{J}$ at terminal time K, and one representing the contribution before this time. Such a split is relevant for the continuous-time formulation, but for a discrete-time formulation, as used in our text, it offers no added value except to illustrate the analogy with the continuous case.

$$\mathcal{J}\left(\mathbf{u}_{1:K}, \mathbf{y}_{1:K}\left(\mathbf{u}_{1:K}\right)\right) = \sum_{k=1}^{K}\left[\frac{\sum_{i=1}^{N_{inj}} r_{wi}\cdot\left(u_{wi,i}\right)_k + \sum_{j=1}^{N_{prod}}\left(r_{wp}\cdot\left|y_{wp,j}\right|_k + r_o\cdot\left|y_{o,j}\right|_k\right)}{(1+b)^{\frac{t_k}{\tau_t}}}\times\Delta t_k\right], \qquad (7.5)$$

where $u_{wi,i}$ is an input variable representing the water injection rate of well i (positive in our sign convention[†]), $y_{wp,j}$ is an output representing the water production rate of well j (negative), $y_{o,j}$ is also an output, representing the oil production rate of well j (also negative), r_{wi} and r_{wp} are the (constant) water injection and production costs (negative, with units \$/m^3), r_o is the (constant) oil revenue (positive, \$/m^3), Δt_k is the time interval of time step k in days, b is the discount rate for a reference time interval τ_t (which is usually taken as a year), and N_{inj} and N_{prod} are the number of injection wells and production wells respectively. Equation (7.5) can be interpreted as a simplified NPV, i.e. the cumulative discounted cash flow over the producing life of a reservoir, disregarding the capital expenditure for wells, facilities etc. (which are assumed to be fixed). In this example the water injection rates are taken as inputs, and the oil and water production rates as outputs. In a more general case the injection and production rates could either be inputs or outputs, or even change role over time. Note that if we use an extended output vector as introduced in equation (3.154), all flow rates become outputs and the objective function is no longer a direct function of the inputs.

7.2.3 Constraints

In practice, the elements of the input vector $\mathbf{u}_k$ are often constrained to stay within certain limits. For example, bottom hole pressures in injectors are usually limited to a certain maximum because of the risk of fracturing the rock around the well. Similarly bottom hole pressures in the producers are usually limited to a certain minimum because otherwise it would not be possible to lift the produced fluids to surface. Another form of constraint is when we require the volume of injected water to be equal to the total volume of the fluids produced, a situation known as *voidage replacement*. Also well rates are usually constrained to maximum values. These limitations may all be expressed as *equality* or *inequality constraints*, which can be represented in a general form as

$$\mathbf{c}_k\left(\mathbf{u}_k, \mathbf{y}_k\right) = \mathbf{0} \ , \qquad (7.6)$$

and

$$\mathbf{d}_k\left(\mathbf{u}_k, \mathbf{y}_k\right) \le \mathbf{0} \ . \qquad (7.7)$$

The control problem can now be formulated as

$$\max_{\mathbf{u}_{1:K}} \mathcal{J}\left(\mathbf{u}_{1:K}, \mathbf{y}_{1:K}\left(\mathbf{u}_{1:K}\right)\right) \ , \qquad (7.8)$$

subject to

- system equations (7.1): $\mathbf{g}_k\left(\mathbf{u}_k, \mathbf{x}_{k-1}, \mathbf{x}_k\right) = \mathbf{0}$, $k = 1, 2, \ldots, K$,
- initial conditions (7.2): $\mathbf{x}_0 = \breve{\mathbf{x}}_0$,
- output equations (7.3): $\mathbf{j}_k\left(\mathbf{u}_k, \mathbf{x}_k, \mathbf{y}_k\right) = \mathbf{0}$,
- equality constraints (7.6): $\mathbf{c}_k\left(\mathbf{u}_k, \mathbf{y}_k\right) = \mathbf{0}$,

[†] Flow into the reservoir is taken as positive, while flow out of the reservoir is negative.

- inequality constraints (7.7): $\mathbf{d}_k(\mathbf{u}_k, \mathbf{y}_k) \le \mathbf{0}$.

In the following we will consider problems with

- n inputs, i.e. $\mathbf{u} \in U \subset \mathbb{R}^n$,
- m states, i.e. $\mathbf{x} \in X \subset \mathbb{R}^m$,
- p outputs, i.e. $\mathbf{y} \in Y \subset \mathbb{R}^p$,
- q equality constraints, i.e. $\mathbf{c} \in C \subset \mathbb{R}^q$,
- r inequality constraints, i.e. $\mathbf{d} \in D \subset \mathbb{R}^r$.

Note that the sets C, D, U, X and Y are subsets of the set of real numbers because their elements are constrained to stay within physical limits, e.g. pressures are always positive and saturations, by definition, have values between zero and one.

7.3 Optimal control theory

7.3.1 Adjoint equation - derivation

Derivatives

Starting from the optimization problem (7.8) we aim to compute the optimal control $\mathbf{u}_{1:K}$, with the aid of a gradient-based algorithm, which requires the derivatives of $\mathcal{J}(\mathbf{u}_{1:K}, \mathbf{y}_{1:K}(\mathbf{u}_{1:K}))$ with respect to $\mathbf{u}_{1:K}$. An efficient way to compute these derivatives is through the use of an optimization technique known as *optimal control*; see e.g. Bryson and Ho (1975) or Stengel (1986). The problem in determining the derivatives is the indirect dependence of the variation $\delta\mathcal{J}$ in the objective function on a variation δu_{ik} of the input. Here, δu_{ik} means the variation of element i of vector $\mathbf{u}_k$ at time k. A variation δu_{ik}, at an arbitrary time k, does not only directly influence $\mathcal{J}$ at time k, but also, as follows from recursive application of equation (7.1), the states $\mathbf{x}_{k:K}$, which in turn, through equation (7.3), influence the outputs $\mathbf{y}_{k:K}$ and thus $\mathcal{J}$ at later times. The effect of a single variation δu_{ik} should therefore be computed, using the chain rule for differentiation, as

$$\frac{d\mathcal{J}}{du_{ik}} = \left[\frac{\partial \mathcal{J}_k}{\partial \mathbf{u}_k} + \sum_{j=k}^{K} \frac{\partial \mathcal{J}_j}{\partial \mathbf{y}_j} \left(\frac{\partial \mathbf{y}_j}{\partial \mathbf{u}_k} + \frac{\partial \mathbf{y}_j}{\partial \mathbf{x}_j} \frac{\partial \mathbf{x}_j}{\partial \mathbf{u}_k} \right) \right] \frac{\partial \mathbf{u}_k}{\partial u_{ik}} , \tag{7.9}$$

where j is a dummy variable within the summation, and where we used the ordinary differential d instead of the variational symbol δ, to adhere to the usual notation in the literature. Output equation (7.3) is often an explicit linear algebraic equation such that the terms $\partial \mathbf{y}_j / \partial \mathbf{u}_k$ and $\partial \mathbf{y}_j / \partial \mathbf{x}_j$ in equation (7.9) can be computed directly[†]. Furthermore, the terms $\partial \mathcal{J}_k / \partial \mathbf{u}_k$ and $\partial \mathcal{J}_j / \partial \mathbf{y}_j$ can usually also be computed without problems. However, the term $\partial \mathbf{x}_j / \partial \mathbf{u}_k$ causes difficulties because we need to solve the recursive system of discrete-time differential equations (7.1) to connect the state vectors $\mathbf{x}_j, j = k, k+1, \ldots, K$ to the input $\mathbf{u}_k$ at time $j = k$. Often the need to determine the derivatives with respect to the individual elements of the input vector $\mathbf{u}_k$ is assumed to be understood tacitly in which case equation (7.9) could be written as

$$\frac{d\mathcal{J}}{d\mathbf{u}_k} = \frac{\partial \mathcal{J}_k}{\partial \mathbf{u}_k} + \sum_{j=k}^{K} \frac{\partial \mathcal{J}_j}{\partial \mathbf{y}_j} \left(\frac{\partial \mathbf{y}_j}{\partial \mathbf{u}_k} + \frac{\partial \mathbf{y}_j}{\partial \mathbf{x}_j} \frac{\partial \mathbf{x}_j}{\partial \mathbf{u}_k} \right). \tag{7.10}$$

[†] All entries of $\partial \mathbf{y}_j / \partial \mathbf{u}_k$ for $j \neq k$ will be equal to zero.

In the following sections we will describe an efficient numerical technique to compute the vector of total derivatives $d\mathcal{J}/d\mathbf{u}_k$ with the aid of a so-called *adjoint equation*.

Lagrange multipliers

The complex temporal dependence of the elements in $\partial\mathbf{x}_j/\partial\mathbf{u}_k$ can be taken into account by considering equation (7.1) as a set of additional constraints to the optimization problem, and applying the technique of Lagrange multipliers to solve the constrained optimization problem. Moreover, we may formally also consider the initial condition (7.2) and the output equation (7.3) as constraints, and, setting aside the 'ordinary constraints' **c** and **d**, we can therefore define a modified objective function

$$\bar{\mathcal{J}}\left(\mathbf{u}_{1:K},\mathbf{x}_{0:K},\mathbf{y}_{1:K},\boldsymbol{\lambda}_{0:K},\boldsymbol{\mu}_{1:K}\right)\triangleq\sum_{k=1}^{K}\begin{bmatrix}\mathcal{J}_k\left(\mathbf{u}_k,\mathbf{y}_k\right)\\+\boldsymbol{\lambda}_0^T\left(\mathbf{x}_0-\breve{\mathbf{x}}_0\right)\delta_{k-1}\\+\boldsymbol{\lambda}_k^T\mathbf{g}_k\left(\mathbf{u}_k,\mathbf{x}_{k-1},\mathbf{x}_k\right)\\+\boldsymbol{\mu}_k^T\mathbf{j}_k\left(\mathbf{u}_k,\mathbf{x}_k,\mathbf{y}_k\right)\end{bmatrix},\tag{7.11}$$

where the 'initial condition constraint' $\mathbf{x}_0-\breve{\mathbf{x}}_0=\mathbf{0}$, the 'system constraint' $\mathbf{g}_{1:K}=\mathbf{0}$, and the 'output constraint' $\mathbf{j}_{1:K}=\mathbf{0}$ have been 'adjoined' to $\mathcal{J}_{k+1}$ with the aid of vectors of Lagrange multipliers $\boldsymbol{\lambda}_0$, $\boldsymbol{\lambda}_{1:K}$ and $\boldsymbol{\mu}_{1:K}$ respectively. The Kronecker delta[‡] δ_{k-1} in equation (7.11) ensures that the initial condition constraint is included in the summation.

Euler-Lagrange equations

A necessary condition for a maximum of the modified objective function (7.11) is stationarity of the first variation of $\bar{\mathcal{J}}$ with respect to all dependent variables. In other words, all first-order derivatives should be equal to zero which, after some reorganization of terms, leads to the following set of equations:

$$\frac{\partial\bar{\mathcal{J}}}{\partial\mathbf{u}_k}\equiv\frac{\partial\mathcal{J}_k}{\partial\mathbf{u}_k}+\boldsymbol{\lambda}_k^T\frac{\partial\mathbf{g}_k}{\partial\mathbf{u}_k}+\boldsymbol{\mu}_k^T\frac{\partial\mathbf{j}_k}{\partial\mathbf{u}_k}=\mathbf{0}^T,\qquad k=1,2,\ldots,K,\tag{7.12}$$

$$\frac{\partial\bar{\mathcal{J}}}{\partial\mathbf{x}_0}\equiv\boldsymbol{\lambda}_1^T\frac{\partial\mathbf{g}_1}{\partial\mathbf{x}_0}+\boldsymbol{\lambda}_0^T=\mathbf{0}^T,\tag{7.13}$$

$$\frac{\partial\bar{\mathcal{J}}}{\partial\mathbf{x}_k}\equiv\boldsymbol{\lambda}_{k+1}^T\frac{\partial\mathbf{g}_{k+1}}{\partial\mathbf{x}_k}+\boldsymbol{\lambda}_k^T\frac{\partial\mathbf{g}_k}{\partial\mathbf{x}_k}+\boldsymbol{\mu}_k^T\frac{\partial\mathbf{j}_k}{\partial\mathbf{x}_k}=\mathbf{0}^T,\qquad k=1,2,\ldots,K-1,\tag{7.14}$$

$$\frac{\partial\bar{\mathcal{J}}}{\partial\mathbf{x}_K}\equiv\boldsymbol{\lambda}_K^T\frac{\partial\mathbf{g}_K}{\partial\mathbf{x}_K}+\boldsymbol{\mu}_K^T\frac{\partial\mathbf{j}_K}{\partial\mathbf{x}_K}=\mathbf{0}^T,\tag{7.15}$$

$$\frac{\partial\bar{\mathcal{J}}}{\partial\mathbf{y}_k}\equiv\frac{\partial\mathcal{J}_k}{\partial\mathbf{y}_k}+\boldsymbol{\mu}_k^T\frac{\partial\mathbf{j}_k}{\partial\mathbf{y}_k}=\mathbf{0}^T,\qquad k=1,2,\ldots,K,\tag{7.16}$$

$$\frac{\partial\bar{\mathcal{J}}}{\partial\boldsymbol{\lambda}_0}\equiv\left(\mathbf{x}_0-\breve{\mathbf{x}}_0\right)^T=\mathbf{0}^T,\tag{7.17}$$

[‡] The Kronecker delta is defined as $\delta_k=1$ if $k=0$, $\delta_k=0$ if $k\neq0$, and can be interpreted as the discrete version of the Dirac delta function introduced in equation (5.6). Note that it is indicated with the same symbol as a variation.

$$\frac{\partial \bar{\mathcal{J}}}{\partial \boldsymbol{\lambda}_k} \equiv \mathbf{g}_k^T\left(\mathbf{u}_k, \mathbf{x}_{k-1}, \mathbf{x}_k\right) = \mathbf{0}^T, \qquad k = 1, 2, \ldots, K, \tag{7.18}$$

$$\frac{\partial \bar{\mathcal{J}}}{\partial \boldsymbol{\mu}_k} \equiv \mathbf{j}_k^T\left(\mathbf{u}_k, \mathbf{x}_k, \mathbf{y}_k\right) = \mathbf{0}^T, \qquad k = 1, 2, \ldots, K. \tag{7.19}$$

Equations (7.12) to (7.19) are the first-order necessary conditions for an optimum, which are in optimal control theory often called the *Euler-Lagrange equations*[†]. We will discuss their meaning, going from the bottom to the top. The last three equations are identical to output equation (7.3), system equation (7.1) and initial condition (7.2), and are therefore automatically satisfied. Equation (7.16) allows us to compute the Lagrange multipliers $\boldsymbol{\mu}_{1:K}$. Next we can use equation (7.15) to compute multiplier $\boldsymbol{\lambda}_K$ for the final discrete time K, and thereafter equation (7.14) to recursively compute the multipliers $\boldsymbol{\lambda}_k$ for $k = K-1, K-2, \ldots, 1$, i.e. backward in time. This last step becomes more clear by rewriting equation (7.14) as

$$\left(\frac{\partial \mathbf{g}_k}{\partial \mathbf{x}_k}\right)^T \boldsymbol{\lambda}_k = -\left(\frac{\partial \mathbf{g}_{k+1}}{\partial \mathbf{x}_k}\right)^T \boldsymbol{\lambda}_{k+1} - \left(\frac{\partial \mathbf{j}_k}{\partial \mathbf{x}_k}\right)^T \boldsymbol{\mu}_k, \tag{7.20}$$

which is a discrete-time differential equation for $\boldsymbol{\lambda}_k$ that runs backward in time starting from 'final condition' $\boldsymbol{\lambda}_K$. Formally we can solve it as

$$\boldsymbol{\lambda}_k = -\left[\left(\frac{\partial \mathbf{g}_k}{\partial \mathbf{x}_k}\right)^T\right]^{-1}\left[\left(\frac{\partial \mathbf{g}_{k+1}}{\partial \mathbf{x}_k}\right)^T \boldsymbol{\lambda}_{k+1} + \left(\frac{\partial \mathbf{j}_k}{\partial \mathbf{x}_k}\right)^T \boldsymbol{\mu}_k\right], \tag{7.21}$$

although in practice we will, as usual, solve the system of equations (7.20) for the unknown $\boldsymbol{\lambda}_k$, rather than explicitly computing the inverse. Equation (7.13) then allows us to compute $\boldsymbol{\lambda}_0$[‡]. Finally, equation (7.12) represents the effect of changing the control $\mathbf{u}_k$ on the value of the objective function, while keeping all other variables fixed. Because $\partial \bar{\mathcal{J}} / \partial \mathbf{u}_k = d\mathcal{J} / d\mathbf{u}_k$, this is just the expression we were looking for, i.e. equation (7.10), but now with implicitly evaluated derivatives. For a non-optimal control this term is not equal to zero, but then its residual is just the modified gradient that we require to iteratively obtain the optimal control using a gradient-based algorithm.

Gradient computation

Computation of the gradient vectors as part of an iterative gradient-based optimization procedure can now be performed according to the following algorithm:

Algorithm 7.1

1. Choose an initial control vector $\mathbf{u}_{1:K}$.
2. Compute the states $\mathbf{x}_{1:K}$ and outputs $\mathbf{y}_{1:K}$ using equations (7.1) and (7.3), starting from initial conditions (7.2).

[†] Usually the term Euler-Lagrange equations is used for a subset of these equations, which implies that equations (7.12) to (7.19) could be referred to as *extended Euler-Lagrange equations*.

[‡] Formally, equation (7.13) represents the effect of changing the initial condition $\mathbf{x}_0$ on the value of the objective function, while keeping all other variables fixed. However, because we prescribed the initial condition through equation (7.2) this term is in our case only of theoretical relevance and we do not need to compute $\boldsymbol{\lambda}_0$. Alternatively, we could consider $\mathbf{x}_0$ as an additional control variable, but normally we will not be able to influence its value.

3. Compute the value of the objective function $\mathcal{J}$ using equation (7.4). If converged stop, else continue.
4. Compute the Lagrange multipliers $\boldsymbol{\mu}_{1:K}$ and $\boldsymbol{\lambda}_{1:K}$ using equations (7.16), (7.15) and (7.14).
5. Compute the total derivatives (transposed gradients) $d\mathcal{J}/d\mathbf{u}_{1:K}$ of the objective function to the controls from the residuals of equation (7.12) according to:

$$\frac{d\mathcal{J}}{d\mathbf{u}_k} \equiv \frac{\partial \bar{\mathcal{J}}}{\partial \mathbf{u}_k} = \frac{\partial \mathcal{J}_k}{\partial \mathbf{u}_k} + \boldsymbol{\lambda}_k^T \frac{\partial \mathbf{g}_k}{\partial \mathbf{u}_k} + \boldsymbol{\mu}_k^T \frac{\partial \mathbf{j}_k}{\partial \mathbf{u}_k}, \; k = 1, 2, \ldots, K \,. \tag{7.22}$$

   Compute an improved estimate of the control vector $\mathbf{u}_{1:K}$, using the derivatives as obtained from equation (7.22), and a gradient-based minimization routine of choice.
6. Return to 2.

Because of its computational efficiency in calculating the gradients of the objective function $\mathcal{J}$ with respect to the control variables $\mathbf{u}_{1:K}$ the use of optimal control theory is particularly useful in optimization problems with a large number of input variables. Implementation in a numerical reservoir simulator is conceptually relatively simple if the simulator is fully implicit, because in that case the partial derivatives $\partial \mathbf{g}_k / \partial \mathbf{x}_k$ and $\partial \mathbf{g}_{k+1} / \partial \mathbf{x}_k$, as required in equation (7.20), are already available; see Sarma et al. (2005a).

7.3.2 Lagrangian and Hamiltonian*

Sometimes, in the derivation of the adjoint equations use is made of an auxiliary variable, the the *Lagrangian*, defined as

$$\begin{aligned} &\mathcal{L}_k\left(\mathbf{u}_k, \mathbf{x}_{k-1}, \mathbf{x}_k, \mathbf{y}_k, \boldsymbol{\lambda}_0, \boldsymbol{\lambda}_k, \boldsymbol{\mu}_k\right) \\ &\triangleq \mathcal{J}_k\left(\mathbf{u}_k, \mathbf{y}_k\right) + \boldsymbol{\lambda}_0^T\left(\mathbf{x}_0 - \breve{\mathbf{x}}_0\right)\delta_{k-1} + \boldsymbol{\lambda}_k^T \mathbf{g}_k\left(\mathbf{u}_k, \mathbf{x}_{k-1}, \mathbf{x}_k\right) + \boldsymbol{\mu}_k^T \mathbf{j}_k\left(\mathbf{u}_k, \mathbf{x}_k, \mathbf{y}_k\right), \end{aligned} \tag{7.23}$$

with which we can rewrite equation (7.11) as

$$\bar{\mathcal{J}}\left(\mathbf{u}_{1:K}, \mathbf{x}_{0:K}, \mathbf{y}_{1:K}, \boldsymbol{\lambda}_{0:K}, \boldsymbol{\mu}_{1:K}\right) = \sum_{k=1}^{K} \mathcal{L}_k\left(\mathbf{u}_k, \mathbf{x}_{k-1}, \mathbf{x}_k, \mathbf{y}_k, \boldsymbol{\lambda}_0, \boldsymbol{\lambda}_k, \boldsymbol{\mu}_k\right) \,. \tag{7.24}$$

Taking the first variation of equation (7.24) we obtain:

$$\begin{aligned} \delta\bar{\mathcal{J}} = &\sum_{k=1}^{K} \frac{\partial \mathcal{L}_k}{\partial \mathbf{u}_k}\delta\mathbf{u}_k + \sum_{k=1}^{K} \frac{\partial \mathcal{L}_k}{\partial \mathbf{x}_{k-1}}\delta\mathbf{x}_{k-1} + \sum_{k=1}^{K} \frac{\partial \mathcal{L}_k}{\partial \mathbf{x}_k}\delta\mathbf{x}_k + \sum_{k=1}^{K} \frac{\partial \mathcal{L}_k}{\partial \mathbf{y}_k}\delta\mathbf{y}_k \\ &+ \frac{\partial \mathcal{L}_1}{\partial \boldsymbol{\lambda}_0}\delta\boldsymbol{\lambda}_0 + \sum_{k=1}^{K} \frac{\partial \mathcal{L}_k}{\partial \boldsymbol{\lambda}_k}\delta\boldsymbol{\lambda}_k + \sum_{k=1}^{K} \frac{\partial \mathcal{L}_k}{\partial \boldsymbol{\mu}_k}\delta\boldsymbol{\mu}_k \,. \end{aligned} \tag{7.25}$$

By splitting off the terms for $k = 0$ and $k = K$ in the second and third terms at the right-hand side of equation (7.25) respectively and reordering the results we obtain

$$\begin{aligned} \delta\bar{\mathcal{J}} = &\sum_{k=1}^{K} \frac{\partial \mathcal{L}_k}{\partial \mathbf{u}_k}\delta\mathbf{u}_k + \frac{\partial \mathcal{L}_1}{\partial \mathbf{x}_0}\delta\mathbf{x}_0 + \sum_{k=1}^{K-1}\left(\frac{\partial \mathcal{L}_{k+1}}{\partial \mathbf{x}_k} + \frac{\partial \mathcal{L}_k}{\partial \mathbf{x}_k}\right)\delta\mathbf{x}_k + \frac{\partial \mathcal{L}_K}{\partial \mathbf{x}_K}\delta\mathbf{x}_K \\ &+ \sum_{k=1}^{K} \frac{\partial \mathcal{L}_k}{\partial \mathbf{y}_k}\delta\mathbf{y}_{k+1} + \frac{\partial \mathcal{L}_1}{\partial \boldsymbol{\lambda}_0}\delta\boldsymbol{\lambda}_0 + \sum_{k=1}^{K} \frac{\partial \mathcal{L}_k}{\partial \boldsymbol{\lambda}_k}\delta\boldsymbol{\lambda}_k + \sum_{k=1}^{K} \frac{\partial \mathcal{L}_k}{\partial \boldsymbol{\mu}_k}\delta\boldsymbol{\mu}_k \,. \end{aligned} \tag{7.26}$$

Requiring stationarity of $\delta\bar{\mathcal{J}}$ for all variations, and using the definition of the Lagrangian (7.23) to work out the terms in equation (7.26), we recover the Euler-Lagrange equations (7.12) to (7.19). Sometimes another auxiliary function, the *Hamiltonian*, is used. Both the Lagrangian and the Hamiltonian have their origin in classical mechanics; see e.g. Landau and Lifshitz (1960). Use of the Hamiltonian is more appropriate when the system equations are expressed in explicit form $\mathbf{x}_k = \mathbf{f}_k(\mathbf{u}_k, \mathbf{x}_{k-1})$, in which case we have to adjoin the expression $\mathbf{x}_k - \mathbf{f}_k(\mathbf{u}_k, \mathbf{x}_{k-1})$ to the objective function. The Hamiltonian is then defined as

$$\mathcal{H}_k \triangleq \mathcal{J}_k\left(\mathbf{u}_k, \mathbf{y}_k\right) + \boldsymbol{\lambda}_0^T\left(\mathbf{x}_0 - \breve{\mathbf{x}}\right)\delta_{k-1} - \boldsymbol{\lambda}_k^T\mathbf{f}_k\left(\mathbf{u}_k, \mathbf{x}_{k-1}\right) + \boldsymbol{\mu}_k^T\mathbf{j}_k\left(\mathbf{u}_k, \mathbf{x}_k, \mathbf{y}_k\right), \tag{7.27}$$

such that we can rewrite equation (7.11) as

$$\bar{\mathcal{J}}\left(\mathbf{u}_{1:K}, \mathbf{x}_{0:K}, \mathbf{y}_{1:K}, \boldsymbol{\lambda}_{0:K}, \boldsymbol{\mu}_{1:K}\right) = \sum_{k=1}^{K}\left[\mathbf{x}_k + \mathcal{H}_k\left(\mathbf{u}_k, \mathbf{x}_{k-1}, \mathbf{y}_k, \boldsymbol{\lambda}_0, \boldsymbol{\lambda}_k, \boldsymbol{\mu}_k\right)\right]. \tag{7.28}$$

7.3.3 Adjoint equation – interpretations*

*Tangent linear model**

The adjoint formulation provides a computationally efficient means of computing derivatives $d\mathcal{J}/du_{ik}$ by considering the propagation of perturbations δu_{ik} (or $\delta\mathbf{u}_k$) through the system with the aid of a transposed tangent linear system model. This can be emphasized by rewriting the Jacobians used in the adjoint formulation as:

$$\bar{\hat{\mathbf{E}}}_k \triangleq \frac{\partial \mathbf{g}_k}{\partial \mathbf{x}_k},\ \bar{\hat{\mathbf{A}}}_k \triangleq \frac{\partial \mathbf{g}_k}{\partial \mathbf{x}_{k-1}},\ \bar{\hat{\mathbf{B}}}_k \triangleq \frac{\partial \mathbf{g}_k}{\partial \mathbf{u}_k}, \tag{7.29, 7.30, 7.31}$$

$$\bar{\hat{\mathbf{C}}}_k \triangleq \frac{\partial \mathbf{j}_k}{\partial \mathbf{x}_k},\ \bar{\hat{\mathbf{D}}}_k \triangleq \frac{\partial \mathbf{j}_k}{\partial \mathbf{u}_k},\ \bar{\hat{\mathbf{F}}}_k \triangleq \frac{\partial \mathbf{j}_k}{\partial \mathbf{y}_k}, \tag{7.32, 7.33, 7.34}$$

where the overbars indicate tangent matrices and the hats indicate that we use a generalized state space formulation, in line with the notation convention of Section 3.2[†]. To simplify the notation we will drop the overbars in the remainder of this chapter. With the aid of expressions (7.29) to (7.34) we can rewrite equations (7.12) to (7.16), and after reorganizing the terms we obtain the following computational sequence:

$$\hat{\mathbf{F}}_k^T\boldsymbol{\mu}_k = -\left(\frac{\partial \mathcal{J}_k}{\partial \mathbf{y}_k}\right)^T, \qquad k = 1, 2, \ldots, K, \tag{7.35}$$

$$\hat{\mathbf{E}}_K^T\boldsymbol{\lambda}_K = -\hat{\mathbf{C}}_K^T\boldsymbol{\mu}_K, \tag{7.36}$$

$$\hat{\mathbf{E}}_k^T\boldsymbol{\lambda}_k = -\hat{\mathbf{A}}_{k+1}^T\boldsymbol{\lambda}_{k+1} - \hat{\mathbf{C}}_k^T\boldsymbol{\mu}_k, \qquad k = 1, 2, \ldots, K-1, \tag{7.37}$$

$$\boldsymbol{\lambda}_0 = -\hat{\mathbf{A}}_1^T\boldsymbol{\lambda}_1, \tag{7.38}$$

$$\left(\frac{\partial \bar{\mathcal{J}}}{\partial \mathbf{u}_k}\right)^T = \left(\frac{\partial \mathcal{J}_k}{\partial \mathbf{u}_k}\right)^T + \hat{\mathbf{B}}_k^T\boldsymbol{\lambda}_k + \hat{\mathbf{D}}_k^T\boldsymbol{\mu}_k, \qquad k = 1, 2, \ldots, K. \tag{7.39}$$

[†] C.f. equations (3.30) to (3.35). Note that here we use derivatives with respect to the time-discretized variables $\mathbf{u}_k$, $\mathbf{x}_{k\text{-}1}$, $\mathbf{x}_k$ and $\mathbf{y}_k$, whereas in equations (3.30) to (3.35) the derivatives were taken with respect to the time-continuous variables $\mathbf{u}(t)$, $\mathbf{x}(t)$, $\dot{\mathbf{x}}(t)$ and $\mathbf{y}(t)$.

The linear discrete-time 'backward' differential equation (7.37), which describes the evolution of the Lagrange multipliers, can be interpreted as the dual of a tangent-linear 'forward' equation

$$\hat{\mathbf{E}}_k \mathbf{x}_k = \hat{\mathbf{A}}_{k-1} \mathbf{x}_{k-1} + \hat{\mathbf{B}}_k \mathbf{u}_k , \tag{7.40}$$

which can be obtained by linearization of the nonlinear discrete-time system equation $\mathbf{g}_k(\mathbf{u}_k, \mathbf{x}_{k-1}, \mathbf{x}_k) = \mathbf{0}$ and which describes the tangent evolution of the states. Note that in our derivation of the adjoint equations we first discretized and then linearized the system equation. It is also possible to use the reversed order and derive the adjoint equations from the continuous-time system equation and thereafter discretize them in time. The two different approaches may lead to slightly different formulations. It is important that the Jacobians used in the forward and backward equations are identical. This implies that they should both be obtained through differentiation with respect to either time-discretized or time-continuous variables, to ensure that the equations are truly each others adjoint.

*Super vectors**

An alternative way to interpret the adjoint equation has been presented in the reservoir engineering literature by Rodrigues (2006) and Kraaijevanger et al. (2007). Instead of considering the individual vectors $\mathbf{u}_k$, $\mathbf{x}_k$, and $\mathbf{y}_k$ for time steps $k = 1, 2, \ldots, K$ they combined them in *super vectors* defined as

$$\mathsf{u} \triangleq \begin{bmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \\ \vdots \\ \mathbf{u}_K \end{bmatrix}, \quad \mathsf{x} \triangleq \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \vdots \\ \mathbf{x}_K \end{bmatrix}, \quad \mathsf{y} \triangleq \begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \\ \vdots \\ \mathbf{y}_K \end{bmatrix}, \tag{7.41, 7.42, 7.43}$$

with which we can write the system equations (7.1) and ouput equations (7.3) in super vector form as

$$\mathsf{g}(\mathsf{u},\mathsf{x}) \triangleq \begin{bmatrix} \mathbf{g}_1(\mathbf{u}_1, \mathbf{x}_0, \mathbf{x}_1) \\ \mathbf{g}_2(\mathbf{u}_2, \mathbf{x}_1, \mathbf{x}_2) \\ \vdots \\ \mathbf{g}_K(\mathbf{u}_K, \mathbf{x}_{K-1}, \mathbf{x}_K) \end{bmatrix}, \quad \mathsf{j}(\mathsf{u},\mathsf{x},\mathsf{y}) \triangleq \begin{bmatrix} \mathbf{j}_1(\mathbf{u}_1, \mathbf{x}_1, \mathbf{y}_1) \\ \mathbf{j}_2(\mathbf{u}_2, \mathbf{x}_2, \mathbf{y}_2) \\ \vdots \\ \mathbf{j}_K(\mathbf{u}_K, \mathbf{x}_K, \mathbf{y}_K) \end{bmatrix}. \tag{7.44, 7.45}$$

Also equation (7.10) can now be written more simply as

$$\frac{d\mathcal{J}}{d\mathsf{u}} = \frac{\partial \mathcal{J}}{\partial \mathsf{u}} + \frac{\partial \mathcal{J}}{\partial \mathsf{y}} \left[\frac{\partial \mathsf{y}}{\partial \mathsf{u}} + \frac{\partial \mathsf{y}}{\partial \mathsf{x}} \frac{\partial \mathsf{x}}{\partial \mathsf{u}} \right]. \tag{7.46}$$

The term $\partial \mathsf{x} / \partial \mathsf{u}$ can be computed, in theory, through implicit differentiation as

$$\frac{\partial \mathsf{x}}{\partial \mathsf{u}} = -\left(\frac{\partial \mathsf{g}}{\partial \mathsf{x}} \right)^{-1} \frac{\partial \mathsf{g}}{\partial \mathsf{u}} , \tag{7.47}$$

and similar expressions can be obtained for the other terms[*]. With these, equation (7.46) can be rewritten as

[*] To compute $\partial \mathsf{y} / \partial \mathsf{u}$ for $\mathsf{j}(\mathsf{u}, \mathsf{x}, \mathsf{y}(\mathsf{u}, \mathsf{x})) = \mathsf{0}$, write $d\mathsf{j} \equiv (\partial \mathsf{j} / \partial \mathsf{u})\, d\mathsf{u} + (\partial \mathsf{j} / \partial \mathsf{x})\, d\mathsf{x} + (\partial \mathsf{j} / \partial \mathsf{y})\, d\mathsf{y} = \mathsf{0}$ from which follows $d\mathsf{y} / d\mathsf{u} = -(\partial \mathsf{j} / \partial \mathsf{y})^{-1} [(\partial \mathsf{j} / \partial \mathsf{u}) + (\partial \mathsf{j} / \partial \mathsf{x})(d\mathsf{x} / d\mathsf{u})]$. Because the function j does not specify an explicit

$$\underbrace{\frac{d\mathcal{J}}{d\mathbf{u}}}_{1\times M} = \underbrace{\frac{\partial\mathcal{J}}{\partial\mathbf{u}}}_{1\times M} - \underbrace{\frac{\partial\mathcal{J}}{\partial\mathbf{y}}}_{1\times P}\underbrace{\left(\frac{\partial\mathbf{j}}{\partial\mathbf{y}}\right)^{-1}}_{P\times P}\underbrace{\frac{\partial\mathbf{j}}{\partial\mathbf{u}}}_{P\times M} - \underbrace{\frac{\partial\mathcal{J}}{\partial\mathbf{y}}}_{1\times P}\underbrace{\left(\frac{\partial\mathbf{j}}{\partial\mathbf{y}}\right)^{-1}}_{P\times P}\underbrace{\frac{\partial\mathbf{j}}{\partial\mathbf{x}}}_{P\times N}\underbrace{\left(\frac{\partial\mathbf{g}}{\partial\mathbf{x}}\right)^{-1}}_{N\times N}\underbrace{\frac{\partial\mathbf{g}}{\partial\mathbf{u}}}_{N\times M}, \quad (7.48)$$

$$\underbrace{\phantom{\frac{\partial\mathcal{J}}{\partial\mathbf{u}}}}_{\text{I}} \quad \underbrace{\phantom{\frac{\partial\mathcal{J}}{\partial\mathbf{y}}\left(\frac{\partial\mathbf{j}}{\partial\mathbf{y}}\right)^{-1}\frac{\partial\mathbf{j}}{\partial\mathbf{u}}}}_{\text{II}} \quad \underbrace{\phantom{\frac{\partial\mathcal{J}}{\partial\mathbf{y}}\left(\frac{\partial\mathbf{j}}{\partial\mathbf{y}}\right)^{-1}\frac{\partial\mathbf{j}}{\partial\mathbf{x}}\left(\frac{\partial\mathbf{g}}{\partial\mathbf{x}}\right)^{-1}\frac{\partial\mathbf{g}}{\partial\mathbf{u}}}}_{\text{III}}$$

where $M = mK$, $N = nK$ and $P = pK$, with m, n, p and K representing the number of input variables, state variables, output variables and time steps respectively. We stress that super matrices are not used for actual computations because their storage requirements would be far too large for realistic applications, even although they have a sparse structure. However they do give us insight in the amount of operations involved in computing derivatives using different methods. In particular consider term III in equation (7.48). Evaluation of this term from right to left requires a series of matrix-matrix multiplications, whereas evaluation from left to right requires only matrix-vector computations which is computationally much more efficient. If we define the auxiliary super vectors

$$\boldsymbol{\mu}^T \triangleq -\frac{\partial\mathcal{J}}{\partial\mathbf{y}}\left(\frac{\partial\mathbf{j}}{\partial\mathbf{y}}\right)^{-1}, \quad \boldsymbol{\lambda}^T \triangleq -\frac{\partial\mathcal{J}}{\partial\mathbf{y}}\left(\frac{\partial\mathbf{j}}{\partial\mathbf{y}}\right)^{-1}\frac{\partial\mathbf{j}}{\partial\mathbf{x}}\left(\frac{\partial\mathbf{g}}{\partial\mathbf{x}}\right)^{-1}, \quad (7.49)$$

we can rewrite equation (7.48) as

$$\frac{d\mathcal{J}}{d\mathbf{u}} = \frac{\partial\mathcal{J}}{\partial\mathbf{u}} + \boldsymbol{\mu}^T\frac{\partial\mathbf{j}}{\partial\mathbf{u}} + \boldsymbol{\lambda}^T\frac{\partial\mathbf{g}}{\partial\mathbf{u}}. \quad (7.50)$$

The auxiliary super vectors $\boldsymbol{\lambda}$ and $\boldsymbol{\mu}$ can be interpreted as Lagrange multipliers and equation (7.50) is therefore the super-vector form of equation (7.12) and represents the effect of changing the control $\mathbf{u}$ on the value of the objective function $\mathcal{J}$, while keeping all other variables fixed. From detailed inspection of the structure of the super matrices in equation (7.48) it follows that reverse evaluation of term III also implies reverse operation in time; see Kraaijevanger et al. (2007). This is in line with the 'backward' computation of the Lagrange multipliers in equation (7.20). Finally, comparison of equations (7.46) and (7.12) to equations (6.33) and (6.39) respectively illustrates the parallel between the simple constrained optimization example treated in Section 6.1 and the large-scale discrete-time optimal control problem treated here.

*Repeated implicit differentiation**

Yet another interpretation† of the adjoint equations can be obtained by starting from equation (7.10) again:

$$\frac{d\mathcal{J}}{d\mathbf{u}_k} = \sum_{j=k}^{K}\frac{\partial\mathcal{J}_j}{\partial\mathbf{y}_j}\left(\frac{\partial\mathbf{y}_j}{\partial\mathbf{u}_k} + \frac{\partial\mathbf{y}_j}{\partial\mathbf{x}_j}\frac{\partial\mathbf{x}_j}{\partial\mathbf{u}_k}\right), \quad (7.51)$$

where we have left out the term $\partial\mathcal{J}_k/\partial\mathbf{u}_k$ to simplify the exposition. As discussed in Section 7.3.1 the term $\partial\mathbf{x}_j/\partial\mathbf{u}_k$ is problematic because we need to connect the state vectors $\mathbf{x}_\kappa$ at times $j = k, k+1, \ldots, K$ to the input $\mathbf{u}_k$ at time $j = k$. This can be expressed as

---

relationship between $\mathbf{x}$ an $\mathbf{u}$ we have $d\mathbf{x}/d\mathbf{u} = \mathbf{0}$ and therefore $\partial\mathbf{y}/\partial\mathbf{u} = d\mathbf{y}/d\mathbf{u} = -(\partial\mathbf{j}/\partial\mathbf{y})^{-1}(\partial\mathbf{j}/\partial\mathbf{u})$. In a similar way we find that $\partial\mathbf{y}/\partial\mathbf{x} = d\mathbf{y}/d\mathbf{x} = -(\partial\mathbf{j}/\partial\mathbf{y})^{-1}(\partial\mathbf{j}/\partial\mathbf{x})$.

† Personal communication A.W. Heemink, Delft Institute of Applied Mathematics, TU Delft.

$$\frac{\partial \mathbf{x}_j}{\partial \mathbf{u}_k} = \frac{\partial \mathbf{x}_j}{\partial \mathbf{x}_{j-1}} \frac{\partial \mathbf{x}_{j-1}}{\partial \mathbf{x}_{j-2}} \cdots \frac{\partial \mathbf{x}_{k+2}}{\partial \mathbf{x}_{k+1}} \frac{\partial \mathbf{x}_{k+1}}{\partial \mathbf{x}_k} \frac{\partial \mathbf{x}_k}{\partial \mathbf{u}_k}, \tag{7.52}$$

where the terms of the form $\partial \mathbf{x}_{k+1} / \partial \mathbf{x}_k$ can, in theory, be computed through implicit differentiation according to[‡]

$$\frac{\partial \mathbf{x}_k}{\partial \mathbf{x}_{k-1}} = -\left(\frac{\partial \mathbf{g}_k}{\partial \mathbf{x}_k} \right)^{-1} \frac{\partial \mathbf{g}_k}{\partial \mathbf{x}_{k-1}} \equiv -\left(\hat{\mathbf{E}}_k \right)^{-1} \hat{\mathbf{A}}_k, \tag{7.53}$$

where the last term is expressed in terms of the matrices introduced in equations (7.29) and (7.30). Similarly the terms $\partial \mathbf{y}_j / \partial \mathbf{u}_k$ and $\partial \mathbf{x}_k / \partial \mathbf{u}_k$ in equations (7.51) and (7.52) can be computed through implicit differentiation using equations (7.1) and (7.3). If we furthermore define the matrices

$$\mathbf{A}_k \triangleq -\left(\hat{\mathbf{E}}_k \right)^{-1} \hat{\mathbf{A}}_k \;, \quad \mathbf{B}_k \triangleq -\left(\hat{\mathbf{E}}_k \right)^{-1} \hat{\mathbf{B}}_k, \tag{7.54, 7.55}$$

$$\mathbf{C}_k \triangleq -\left(\hat{\mathbf{F}}_k \right)^{-1} \hat{\mathbf{C}}_k \;, \quad \mathbf{D}_k \triangleq -\left(\hat{\mathbf{F}}_k \right)^{-1} \hat{\mathbf{D}}_k, \tag{7.56, 7.57}$$

equation (7.51) can be rewritten as

$$\underbrace{\frac{d\mathcal{J}}{d\mathbf{u}_k}}_{1\times m} = \sum_{j=k}^{K} \underbrace{\frac{\partial \mathcal{J}_j}{\partial \mathbf{y}_j}}_{1\times q} \underbrace{\left(\mathbf{D}_k + \mathbf{C}_j \mathbf{A}_j \mathbf{A}_{j-1} \cdots \mathbf{A}_{k+2} \mathbf{A}_{k+1} \mathbf{B}_k \right)}_{q\times m}, \tag{7.58}$$

which illustrates that the total derivative $d\mathcal{J}/d\mathbf{u}_k$ can be obtained formally through application of the chain rule and repeated implicit differentiation. Equation (7.58) can also be written, in transposed form, as

$$\underbrace{\left[\frac{d\mathcal{J}}{d\mathbf{u}_k} \right]^T}_{m\times 1} = \sum_{j=k}^{K} \underbrace{\left(\mathbf{B}_k^T \mathbf{A}_{k+1}^T \mathbf{A}_{k+2}^T \cdots \mathbf{A}_{j-1}^T \mathbf{A}_j^T \mathbf{C}_j^T + \mathbf{D}_k^T \right)}_{m\times q} \underbrace{\left[\frac{\partial \mathcal{J}_j}{\partial \mathbf{y}_j} \right]^T}_{q\times 1}. \tag{7.59}$$

Evaluation of the terms in the $m \times q$ matrix in this equation from left to right, i.e. forward in time, requires a series of matrix-matrix multiplications, whereas evaluation from right to left i.e. backward in time, requires only matrix-vector computations which is computationally much more efficient. Even although the matrices are typically very sparse, it is computationally much more efficient to perform a large number of matrix-vector computations than the same number of matrix-matrix computations.

Remarks

- The computation of the the total derivative $d\mathcal{J}/d\mathbf{u}_k$ using a forward-in-time computation of the $m \times q$ matrix in equation (7.58) or (7.59) is known as the *forward sensitivity method*. Similarly, computation of $d\mathcal{J}/d\mathbf{u}_k$ using a backward-in-time computation is known as the *backward sensitivity method* (or the adjoint method). Because we consider only one objective function $\mathcal{J}$, and a multivariate input vector $\mathbf{u}$, the backward method is computationally more efficient. If we would consider a vector of multiple objective

[‡] Here we make use of the fact that the functions $\mathbf{g}_k$ and $\mathbf{j}_k$ can be expressed as $\mathbf{g}_k(\mathbf{u}_k, \mathbf{x}_{k-1}, \mathbf{x}_k(\mathbf{u}_k, \mathbf{x}_{k-1})) = \mathbf{0}$ and $\mathbf{j}_k(\mathbf{u}_k, \mathbf{x}_k, \mathbf{y}_k(\mathbf{u}_k, \mathbf{x}_k)) = \mathbf{0}$ respectively. See also the footnote on page 166.

functions, the forward method would become the computationally preferred choice once the number of objective functions exceeded the number of input variables[†].

- Just as the matrix sequence $\mathbf{A}_j\mathbf{A}_{j-1}\cdots\mathbf{A}_{k+2}\mathbf{A}_{k+1}$ in equation (7.58) represents the partial derivatives $(\partial\mathbf{x}_j/\partial\mathbf{x}_{j-1})(\partial\mathbf{x}_{j-1}/\partial\mathbf{x}_{j-2})\cdots(\partial\mathbf{x}_{k+2}/\partial\mathbf{x}_{k+1})(\partial\mathbf{x}_{k+1}/\partial\mathbf{x}_k)$, see equations (7.51) to (7.58), the matrix sequence $\mathbf{A}_{k+1}^T\mathbf{A}_{k+2}^T\cdots\mathbf{A}_{j-1}^T\mathbf{A}_j^T$ in equation (7.59) can be interpreted to also represent partial derivatives, but now expressed in terms of Lagrange multipliers, and reversed in time: $(\partial\boldsymbol{\lambda}_k/\partial\boldsymbol{\lambda}_{k+1})(\partial\boldsymbol{\lambda}_{k+1}/\partial\boldsymbol{\lambda}_{k+2})\cdots(\partial\boldsymbol{\lambda}_{j-2}/\partial\boldsymbol{\lambda}_{j-1})(\partial\boldsymbol{\lambda}_{j-1}/\partial\boldsymbol{\lambda}_j)$.

7.3.4 Optimality conditions

To be continued.

7.4 Constrained optimization

7.4.1 Input constraints and output constraints

Several authors discussed the incorporation of constraints in the optimal control problem for flooding optimization in reservoir engineering: Virnovski (1991), Brouwer and Jansen (2004), Wang et al. (2009), and van Essen et al. (2009) described partial and sometimes heuristic solutions, only valid for particular types of constraints, while Sarma et al. (2005a, 2008b), De Montleau et al. (2006), Kraaijevanger et al. (2007), Chen et al. (2010), Suwartadi et al. (2009, 2010) and Chen et al. (2012) made more systematic studies, valid for a much broader range of constraint equations. A relatively simple situation occurs when the constraints are specified only in terms of the input variables:

$$\mathbf{c}_k\left(\mathbf{u}_k\right)=\mathbf{0}\ , \tag{7.60}$$

$$\mathbf{d}_k\left(\mathbf{u}_k\right)\le\mathbf{0}\ , \tag{7.61}$$

where **c** and **d** are vector-valued equality and inequality constraint functions respectively. An example of inequality constraints (7.61) occurs in a situation where we use water injection rates as control variables and specify maximum values for the individual rates. An example of equality constraints (7.60) is a similar situation with water injection rates as controls, however under the constraint that the sum of the rates remains constant. A more complicated situation occurs when the constraints also depend on the state variables or the outputs. In particular we will consider *output constraints*[†] of the form

$$\mathbf{c}_k\left(\mathbf{u}_k,\mathbf{y}_k\right)\equiv\mathbf{c}_k\left(\mathbf{u}_k,\mathbf{y}_k\left(\mathbf{u}_k,\mathbf{x}_k\left(\mathbf{u}_{1:k}\right)\right)\right)=\mathbf{0}\ , \tag{7.62}$$

$$\mathbf{d}_k\left(\mathbf{u}_k,\mathbf{y}_k\right)\equiv\mathbf{d}_k\left(\mathbf{u}_k,\mathbf{y}_k\left(\mathbf{u}_k,\mathbf{x}_k\left(\mathbf{u}_{1:k}\right)\right)\right)\le\mathbf{0}\ , \tag{7.63}$$

where $\mathbf{y}_k$ is a function of the input vector $\mathbf{u}_k$ and the state vector $\mathbf{x}_k$, which, in turn, is a function of the input vectors $\mathbf{u}_{1:k}$ as follows from recursive application of equations (7.1) and (7.3). An example of equality constraints (7.62) occurs when we specify a *voidage replacement* condition by requiring that the total amount of produced fluids equals the total

[†] There is no need for a large number of objective functions in case of flooding optimization as discussed in this chapter. However, there may be other applications, e.g. history matching as will be discussed in Chapter 8, where many objective functions may be required.

[†] Because the outputs depend directly on the states, output constraints are special case of *state constraints*. Sometimes these are called *state-path constraints* to emphasize that they contain dependencies along the trajectory in state-time space. In particular, the state values $\mathbf{x}_k$ at time k depend on all inputs $\mathbf{u}_{1:k}$.

amount of injected fluids at each moment in time. Inequality constraints (7.63) occur e.g. when we specify well constraints in terms of bottom hole pressures, or in terms of total rates, phase rates or phase fractions in production wells. In general, such output constraints are much more difficult to handle than the *input constraints* (7.60) and (7.61).

7.4.2 Bound constraints on the input

A special case of inequality constraints, known as bound constraints, are those that limit the input variables to stay within upper and lower bounds:

$$\mathbf{u}_k^- \le \mathbf{u}_k \le \mathbf{u}_k^+ \ , \tag{7.64}$$

where $\mathbf{u}^-$ and $\mathbf{u}^+$ are vectors of lower and upper bounds respectively. Wang et al. (2009) addressed this problem with a nonlinear transformation

$$\tilde{\mathbf{u}}_k = \ln\left(\frac{\mathbf{u}_k - \mathbf{u}_k^-}{\mathbf{u}_k^+ - \mathbf{u}_k}\right). \tag{7.65}$$

The elements of the transformed vector $\tilde{\mathbf{u}}_k$ have lower and upper bounds of $-\infty$ and ∞ respectively such that they can be can be optimized over the entire real axis. Van Essen et al. (2009) applied a different method to cope with bound constraints on the inputs which makes use of a projected gradient

$$\left(\frac{\partial \overline{J}}{\partial \mathbf{u}_k}\right)_{proj} = \mathbf{P}^{\perp}\frac{\partial \overline{J}}{\partial \mathbf{u}_k}. \tag{7.66}$$

Here $\mathbf{P}^{\perp}$ is an orthogonal-projection matrix defined as

$$\mathbf{P}^{\perp} \triangleq \mathbf{I} - \left(\frac{\partial \hat{\mathbf{d}}_k}{\partial \mathbf{u}_k}\right)^T \left[\frac{\partial \hat{\mathbf{d}}_k}{\partial \mathbf{u}_k}\left(\frac{\partial \hat{\mathbf{d}}_k}{\partial \mathbf{u}_k}\right)^T \frac{\partial \hat{\mathbf{d}}_k}{\partial \mathbf{u}_k}\right]^{-1}, \tag{7.67}$$

where $\partial \hat{\mathbf{d}}_k / \partial \mathbf{u}_k$ is the active-constraint matrix, i.e. a matrix of vectors tangent to the active constraints $\hat{\mathbf{d}}$; c.f. equation (6.93) in Section 6.3.5. Matrix $\mathbf{P}^{\perp}$ projects the vector $\partial \overline{J} / \partial \mathbf{u}_k$ on the null space of $\partial \hat{\mathbf{d}}_k / \partial \mathbf{u}_k$; for a derivation see Section A.3.3 in Appendix A. This 'gradient projection method' is most effective for linear input constraints which could be equality or inequality group constraints or bound constraints on individual inputs. However, the situation becomes more complicated for nonlinear input constraints, and in particular for output constraints, whether linear or nonlinear. We refer to Luenberger and He (2010) for an extensive analysis of the method for time-invariant applications. For time-dependent ('optimal control') problems with output constraints, like most reservoir flooding problems, the method is less applicable.

7.4.3 External constraint handling

The simplest way to cope with other equality and inequality constraints is to use standard gradient-based optimization methods for constrained optimization which typically require the following information:

1. Values of the constraint functions $\mathbf{c}_k$ and $\mathbf{d}_k$, and their derivative matrices $[\partial \mathbf{c}/\partial \mathbf{u}]_k$ and $[\partial \mathbf{d}/\partial \mathbf{u}]_k$.
2. Values of the objective function $\mathcal{J}$, and the derivative vector $d\mathcal{J}/d\mathbf{u}_k$ as obtained from equation (7.22).

In case of input constraints we can obtain the constraint derivative matrices $[\partial \mathbf{c}/\partial \mathbf{u}]_k$ and $[\partial \mathbf{d}/\partial \mathbf{u}]_k$ without the need to perform reservoir simulations. However, in case of output constraints the situation is more complicated and determination of the constraint derivative matrices requires the solution of an adjoint equation for each element of $\mathbf{c}_k$ and $\mathbf{d}_k$. In practice this is nearly always a much too large number of adjoint simulations to be computationally feasible, and therefore some form of approximate treatment is required.

7.4.4 Equality constraints

Internal constraint handling

As an alternative to handling the constraints externally, we can incorporate them in the definition of the modified objective function. Restricting the analysis to equality constraints for the moment this results in (Jansen, 2011):

$$\bar{\mathcal{J}}\left(\mathbf{u}_{1:K},\mathbf{x}_{0:K},\mathbf{y}_{1:K},\boldsymbol{\lambda}_{0:K},\boldsymbol{\mu}_{1:K},\mathbf{v}_{1:K}\right)\triangleq\sum_{k=1}^{K}\begin{bmatrix}\mathcal{J}_k\left(\mathbf{u}_k,\mathbf{y}_k\right)\\+\boldsymbol{\lambda}_0^T\left(\mathbf{x}_0-\breve{\mathbf{x}}_0\right)\delta_{k-1}\\+\boldsymbol{\lambda}_k^T\mathbf{g}_k\left(\mathbf{u}_k,\mathbf{x}_{k-1},\mathbf{x}_k\right)\\+\boldsymbol{\mu}_k^T\mathbf{j}_k\left(\mathbf{u}_k,\mathbf{x}_k,\mathbf{y}_k\right)\\+\mathbf{v}_k^T\mathbf{c}_k\left(\mathbf{u}_k,\mathbf{y}_k\right)\end{bmatrix}, \tag{7.68}$$

which is identical to eqution (7.11) except for the addition of the q equality constraints $\mathbf{c}_{1:K}=\mathbf{0}$ which have been adjoined to $\mathcal{J}$ with the aid of q Lagrange multipliers $\mathbf{v}_{1:K}$. Following the same procedure as in Section 7.3.1, we obtain the extended Euler-Lagrange equations

$$\frac{\partial\bar{\mathcal{J}}}{\partial\mathbf{u}_k}\equiv\frac{\partial\mathcal{J}_k}{\partial\mathbf{u}_k}+\boldsymbol{\lambda}_k^T\frac{\partial\mathbf{g}_k}{\partial\mathbf{u}_k}+\boldsymbol{\mu}_k^T\frac{\partial\mathbf{j}_k}{\partial\mathbf{u}_k}+\mathbf{v}_k^T\frac{\partial\mathbf{c}_k}{\partial\mathbf{u}_k}=\mathbf{0}^T,\qquad k=1,2,\ldots,K, \tag{7.69}$$

$$\frac{\partial\bar{\mathcal{J}}}{\partial\mathbf{x}_0}\equiv\boldsymbol{\lambda}_1^T\frac{\partial\mathbf{g}_1}{\partial\mathbf{x}_0}+\boldsymbol{\lambda}_0^T=\mathbf{0}^T, \tag{7.70}$$

$$\frac{\partial\bar{\mathcal{J}}}{\partial\mathbf{x}_k}\equiv\boldsymbol{\lambda}_{k+1}^T\frac{\partial\mathbf{g}_{k+1}}{\partial\mathbf{x}_k}+\boldsymbol{\lambda}_k^T\frac{\partial\mathbf{g}_k}{\partial\mathbf{x}_k}+\boldsymbol{\mu}_k^T\frac{\partial\mathbf{j}_k}{\partial\mathbf{x}_k}=\mathbf{0}^T,\qquad k=1,2,\ldots,K-1, \tag{7.71}$$

$$\frac{\partial\bar{\mathcal{J}}}{\partial\mathbf{x}_K}\equiv\boldsymbol{\lambda}_K^T\frac{\partial\mathbf{g}_K}{\partial\mathbf{x}_K}+\boldsymbol{\mu}_K^T\frac{\partial\mathbf{j}_K}{\partial\mathbf{x}_K}=\mathbf{0}^T, \tag{7.72}$$

$$\frac{\partial\bar{\mathcal{J}}}{\partial\mathbf{y}_k}\equiv\frac{\partial\mathcal{J}_k}{\partial\mathbf{y}_k}+\boldsymbol{\mu}_k^T\frac{\partial\mathbf{j}_k}{\partial\mathbf{y}_k}+\mathbf{v}_k^T\frac{\partial\mathbf{c}_k}{\partial\mathbf{y}_k}=\mathbf{0}^T,\qquad k=1,2,\ldots,K, \tag{7.73}$$

$$\frac{\partial\bar{\mathcal{J}}}{\partial\boldsymbol{\lambda}_0}\equiv\left(\mathbf{x}_0-\breve{\mathbf{x}}_0\right)^T=\mathbf{0}^T, \tag{7.74}$$

$$\frac{\partial\bar{\mathcal{J}}}{\partial\boldsymbol{\lambda}_k}\equiv\mathbf{g}_k^T\left(\mathbf{u}_k,\mathbf{x}_{k-1},\mathbf{x}_k\right)=\mathbf{0}^T,\qquad k=1,2,\ldots,K, \tag{7.75}$$

$$\frac{\partial\bar{\mathcal{J}}}{\partial\boldsymbol{\mu}_k}\equiv\mathbf{j}_k^T\left(\mathbf{u}_k,\mathbf{x}_k,\mathbf{y}_k\right)=\mathbf{0}^T,\qquad k=1,2,\ldots,K. \tag{7.76}$$

$$\frac{\partial \bar{\mathcal{J}}}{\partial \mathbf{v}_k} \equiv \mathbf{c}_k^T\left(\mathbf{u}_k, \mathbf{y}_k\right) = \mathbf{0}^T, \qquad k = 1, 2, \ldots, K. \tag{7.77}$$

Reduced gradient

In comparison with equations (7.12) and (7.16), equations (7.69) and (7.73) each have an additional term that contains the q unknown multipliers $\mathbf{v}_k$. At first sight we might expect that the additional set of q equations (7.77) should enable us to solve for the additional multipliers, but because they do not explicitly contain the multipliers we cannot use them directly and need a somewhat more elaborate procedure. As before we can choose an initial control strategy $\mathbf{u}_{1:K}$ to compute numerical values for the corresponding state and output variables $\mathbf{x}_{1:K}$ and $\mathbf{y}_{1:K}$ and for the Jacobians $\partial \mathbf{g}_k / \partial \mathbf{u}_k$, $\partial \mathbf{g}_k / \partial \mathbf{x}_{k-1}$, etc. During this forward integration of the system equations we may, or may not, choose to obey some or all of the constraints $\mathbf{c}_{1:K} = \mathbf{0}$ using some heuristic, i.e. generally non-optimal, strategy. With the aid of equations (7.69), (7.72) and (7.73) we can set up a system of equations for the multipliers $\boldsymbol{\lambda}_K$, $\boldsymbol{\mu}_K$ and $\mathbf{v}_K$ at the final discrete time K:

$$\underbrace{\begin{bmatrix} \boldsymbol{\lambda}_K^T & \boldsymbol{\mu}_K^T & \mathbf{v}_K^T \end{bmatrix}}_{1\times(n+p+q)} \underbrace{\begin{bmatrix} \dfrac{\partial \mathbf{g}_K}{\partial \mathbf{u}_K} & \dfrac{\partial \mathbf{g}_K}{\partial \mathbf{x}_K} & \mathbf{0} \\ \dfrac{\partial \mathbf{j}_K}{\partial \mathbf{u}_K} & \dfrac{\partial \mathbf{j}_K}{\partial \mathbf{x}_K} & \dfrac{\partial \mathbf{j}_K}{\partial \mathbf{y}_K} \\ \dfrac{\partial \mathbf{c}_K}{\partial \mathbf{u}_K} & \mathbf{0} & \dfrac{\partial \mathbf{c}_K}{\partial \mathbf{y}_K} \end{bmatrix}}_{(n+p+q)\times(m+n+p)} = -\underbrace{\begin{bmatrix} \dfrac{\partial \mathcal{J}_K}{\partial \mathbf{u}_K} & \mathbf{0} & \dfrac{\partial \mathcal{J}_K}{\partial \mathbf{y}_K} \end{bmatrix}}_{1\times(m+n+p)}. \tag{7.78}$$

In the special case that there are $q = m$ linearly independent constraints[†], i.e. just as many constraints as there are control variables, the system matrix in equation (7.78) is square and invertible and allows us to solve for the vector of Lagrange multipliers $[\boldsymbol{\lambda}_K^T \ \boldsymbol{\mu}_K^T \ \mathbf{v}_K^T]^T$. Note that in this special case all elements of $\partial \bar{\mathcal{J}}_K / \partial \mathbf{u}_K$ are identical to zero, i.e. all elements of $\mathbf{u}_K$ are fixed and, at least for this time step, there is nothing left to optimize. Normally, however, there will be fewer constraints than control variables in which case the system matrix in equation (7.78) becomes rectangular. In that case we may decide beforehand which q elements of $\mathbf{u}_K$ are fixed and which $m - q$ elements are left free. Indicating the $q \times 1$ vector of fixed input variables with $\hat{\mathbf{u}}$ we can then replace the derivative matrices $\partial \bullet / \partial \mathbf{u}_K$ in equation (7.78) by reduced-size matrices $\partial \bullet / \partial \hat{\mathbf{u}}_K$ such that the system can be solved again. Therafter we can then substitute the multipliers in equation (7.69) and compute the $m - q$ non-zero elements of $\partial \bar{\mathcal{J}}_K / \partial \mathbf{u}_K$. Going backwards in time we can compute the non-zero elements of $\partial \bar{\mathcal{J}}_k / \partial \mathbf{u}_k$, $k = K-1, \ldots, 1$ in a similar fashion, where the system of equations for the multiplier vector $[\boldsymbol{\lambda}_k^T \ \boldsymbol{\mu}_k^T \ \mathbf{v}_k^T]^T$ is formed with the aid of equations (7.69), (7.71) and (7.73), i.e. for the case of reduced-size matrices $\partial \bullet / \partial \hat{\mathbf{u}}_k$:

[†] With 'linearly independent constraints' we mean that the Jacobian $\partial \mathbf{c}_K / \partial \mathbf{u}_K$ is regular, i.e. that the linearized constraints are linearly independent.

$$\underbrace{\begin{bmatrix}\boldsymbol{\lambda}_k^T & \boldsymbol{\mu}_k^T & \mathbf{v}_k^T\end{bmatrix}}_{1\times(n+p+q)}\underbrace{\begin{bmatrix}\dfrac{\partial \mathbf{g}_k}{\partial \hat{\mathbf{u}}_k} & \dfrac{\partial \mathbf{g}_k}{\partial \mathbf{x}_k} & \mathbf{0} \\ \dfrac{\partial \mathbf{j}_k}{\partial \hat{\mathbf{u}}_k} & \dfrac{\partial \mathbf{j}_k}{\partial \mathbf{x}_k} & \dfrac{\partial \mathbf{j}_k}{\partial \mathbf{y}_k} \\ \dfrac{\partial \mathbf{c}_k}{\partial \hat{\mathbf{u}}_k} & \mathbf{0} & \dfrac{\partial \mathbf{c}_k}{\partial \mathbf{y}_k}\end{bmatrix}}_{(n+p+q)\times(q+n+p)} = -\underbrace{\begin{bmatrix}\dfrac{\partial \mathcal{J}_k}{\partial \hat{\mathbf{u}}_k} & \boldsymbol{\lambda}_{k+1}^T\dfrac{\partial \mathbf{g}_{k+1}}{\partial \mathbf{x}_k} & \dfrac{\partial \mathcal{J}_k}{\partial \mathbf{y}_k}\end{bmatrix}}_{1\times(q+n+p)}, \quad k = K-1, K-2, \ldots, 1. \tag{7.79}$$

Thereafter equation (7.69) gives the *reduced gradient* (c.f. equation (7.22))

$$\frac{d\mathcal{J}}{d\bar{\mathbf{u}}_k} \equiv \frac{\partial \bar{\mathcal{J}}}{\partial \bar{\mathbf{u}}_k} = \frac{\partial \mathcal{J}_k}{\partial \bar{\mathbf{u}}_k} + \boldsymbol{\lambda}_k^T \frac{\partial \mathbf{g}_k}{\partial \bar{\mathbf{u}}_k} + \boldsymbol{\mu}_k^T \frac{\partial \mathbf{j}_k}{\partial \bar{\mathbf{u}}_k} + \mathbf{v}_k^T \frac{\partial \mathbf{c}_k}{\partial \bar{\mathbf{u}}_k}, \qquad k = 1, 2, \ldots, K, \tag{7.80}$$

where $\bar{\mathbf{u}}$ indicates the the $(m-q)\times 1$ vector of free input variables.

To be continued.

7.4.5 Inequality constraints

To incorporate state or output inequality constraints in the optimization, various methods originating from *nonlinear programming* are available, see e.g. Rao (1996) or Luenberger and Ye (2010), and several of them have been applied in water flooding optimization. Virnovski (1991), De Montleau et al. (2006) and Kraaijevanger et al. (2007) have implemented variations of the *generalized reduced gradient* (GRG) method. The latter two introduced so-called *slack variables* to transform inequality constraints to equality constraints according to

$$\mathbf{d}_k\left(\mathbf{u}_k, \mathbf{y}_k\right) - \mathbf{s}_k = \mathbf{0}, \tag{7.81}$$

$$\mathbf{s}_k \geq \mathbf{0}. \tag{7.82}$$

The slack variables $\mathbf{s}_k$ can now be treated as bounded input variables, while the equality constraints (7.81) can be treated using Lagrange multipliers as described in Section 7.4.4 above. Note that although the use of slack variables at first sight appears to remove the difficulty of dealing with state and output constraints, this is not really the case because a non-trivial strategy is required to select the 'fixed' input variables $\hat{\mathbf{u}}_k$ and $\hat{\mathbf{s}}_k$ and to switch between active and inactive inequality constraints. This may require multiple forward and backward (adjoint) simulations before an acceptable search direction has been found. An alternative method to incorporate constraints was proposed by Sarma et al. (2008b) which makes use of a *constraint-lumping* technique in combination with a search strategy known as *Zoutendijk's method* to obtain an approximate constrained gradient. Yet another method was proposed by Chen et al. (2010) who applied an *augmented Lagrangian* method which uses a penalty function that allows for temporary violation of the constraints during the iterative optimization procedure. Another penalty-function approach, but restricted to operate within the feasible region, known as a *barrier method*, was proposed by Suwartadi et al. (2009, 2010). Further possibilities for constrained optimization have been discussed by Sarma et al. (2005a, 2008b) but a systematic comparison of the various methods, in particular for realistically sized reservoir models is lacking at this moment. Two implementations in professional reservoir simulators have been described; see De Montleau et al. (2006) and

Kraaijevanger et al. (2007). Although it is clear that they both use a form of the GRG method, the details of the strategy to search for active constraints cannot be inferred from these publications.

To be continued

7.5 Auxiliary topics

7.5.1 Bang-bang control

Sudaryanto and Yortsos (2000, 2001) observed that sometimes the iteration process during flooding optimization leads to controls that have reached either their maximum or their minimum allowed values. Zandvliet et al. (2007) analyzed under which conditions such 'bang-bang' controls can occur, and showed that this is the case if the problem is linear in the controls, i.e. if the influence of $\mathbf{u}$ in equations (7.1) and (7.3) can be expressed as $\mathbf{x}_k(\mathbf{u}_k) = \mathbf{B}\mathbf{u}_k$ and $\mathbf{y}_k(\mathbf{u}_k) = \mathbf{D}\mathbf{u}_k$ with $\mathbf{B} \in \mathbb{R}^{n\times m}$ and $\mathbf{D} \in \mathbb{R}^{p\times m}$ time-invariant matrices, while at the same time the objective function is linear in the controls, and the constraints are limited to input bounds as expressed in equation (7.64). If it is known in advance that the optimal solution is of a bang-bang nature, the iterative optimization procedure can be simplified by searching for the switching moments for the inputs only, instead of for the entire input trajectory. A practical advantage of bang-bang control is the possibility to implement the inputs with on-off control valves which are considerably cheaper than continuously-variable valves.

7.5.2 Augmented Lagrangian

Doublet et al. (2009) described the use of an alternative definition of the modified objective function, known as an augmented Lagrangian formulation, according to

$$\begin{aligned}\bar{\bar{J}}\left(\mathbf{u}_{1:K},\mathbf{x}_{0:K},\mathbf{y}_{1:K},\boldsymbol{\lambda}_{0:K},\boldsymbol{\mu}_{1:K},\mathbf{v}_{1:K}\right) \triangleq\ & \bar{J}\left(\mathbf{u}_{1:K},\mathbf{x}_{0:K},\mathbf{y}_{1:K},\boldsymbol{\lambda}_{0:K},\boldsymbol{\mu}_{1:K},\mathbf{v}_{1:K}\right) \\ & +\frac{c}{2}\sum_{k=1}^{K}\mathbf{g}_k^T\left(\mathbf{u}_k,\mathbf{x}_{k-1},\mathbf{x}_k\right)\mathbf{g}_k\left(\mathbf{u}_k,\mathbf{x}_{k-1},\mathbf{x}_k\right),\end{aligned} \tag{7.83}$$

where c is a positive scalar, referred to as the penalty parameter, and where $\mathbf{g}_k$ is defined by system equation (7.1). The effect of the additional quadratic terms $\mathbf{g}_k^T\mathbf{g}_k$ is meant to make the optimization procedure less sensitive to errors (residuals) in the 'forward' solution of equation (7.1). Such errors sometimes lead to computational problems during the 'backward' solution of the Lagrange multiplier equation (7.20); see Vakili et al. (2005). Moreover, reducing the sensitivity to errors allows for a reduced tolerance on the Newton-Raphson iterations during the forward solution which may lead to computational gains. As noted in Section 7.4.5, the use of an augmented Lagrangian formulation can also be applied to implement state or input constraints; see Chen et al. (2010).

7.5.3 Continuous versus discrete adjoint

In various areas of engineering discussions have been held about the relative merits of first discretizing the forward flow equations in time and thereafter deriving the discrete-time adjoint equations (the 'first discretize-then-differentiate' approach), versus first deriving the continuous-time adjoint equations and then discretizing the forward and adjoint equations in time (the 'first differentiate-then-discretize' approach). Most authors seem to agree that both methods can be applied as long as the forward and backward equations are truly each others adjoint, which implies discretization at identical moments in time of the forward and backward equations using identical discretization schemes. Brouwer (2004) discussed these

aspects, and recently Kourounis et al. (2010) re-evaluated the matter in relation to adjoints for multi-component ('compositional') simulation. Currently available large-scale reservoir simulation packages all follow the 'first-discretize-then-differentiate' approach.

7.5.4 Multi-level optimization

The total number of control variables in optimization problem (7.8) is equal to $n \times K$ where n is the number of elements in the input vector **u** and K is the number of simulation time steps. With up to hundreds of wells and hundreds of time steps for realistically-sized problems the total number of control variables may therefore be in the order of 10^3 to 10^4. Experience shows that often the optimal trajectories, as found in an iterative procedure, display a 'nervous' character with rapid fluctuations around a slowly changing average. Moreover, it is found that these fine-scale time fluctuations of the optimal trajectories have little or no influence on the objective function value, which implies that there is scope to simplify or 'regularize' the input trajectories. A straightforward way to do so is to use control intervals of predefined length which are spanning many time steps, see e.g. Sarma et al. (2005a) or Kraaijevanger et al. (2007). An alternative method was proposed by Lien et al. (2008) in which the problem is initially solved for just two control intervals in time and two groups of controls in space, i.e. for a total number of four generalized input variables corresponding to a control vector **u** with four elements only. Using a sensitivity measure of the objective function to changes in each of the four elements, a gradual refinement of the control vector, both in time and space is obtained, eventually leading to smooth optimal input trajectories and sometimes also a faster convergence of the optimization algorithm.

7.5.5 Reduced-order modeling

Although typical reservoir models contain 10^4 to 10^6 state variables (grid block pressures and saturations) the extent to which these individual states can be influenced through changing the controls **u** is very limited. It can be shown that only a very small number of spatial patterns, in terms of pressures or saturations, are controllable, and consequently, the amount of control that can be exerted on the oil-water front is also very limited; see e.g. Fyrozjaee and Yortsos (2006), Ramakrishnan (2007), Zandvliet et al. (2008b) and Jansen et al. (2009). Therefore, the dynamics of a 'high-order' reservoir flow model, if controlled by a only limited number of inputs, can be captured in a 'low-order' model in terms of a small number of 'generalized' state variables **z.** An efficient way to empirically derive such a low-order model is with the aid of a technique called 'proper orthogonal decomposition' (POD), also known as 'principal component analysis', 'Karhuenen-Loève decomposition' or the 'method of empirical eigen functions'. For applications of POD to flow through porous media see e.g. Vermeulen et al. (2004), Hein et al. (2004), Van Doren et al. (2006) and Cardoso (2009). In the POD method the generalized state variables **z** are related to the original state variables **x** according to

$$\mathbf{x} = \mathbf{\Phi z}\,, \tag{7.84}$$

where $\mathbf{\Phi} = [\boldsymbol{\varphi}_1 \quad \boldsymbol{\varphi}_2 \quad \cdots \quad \boldsymbol{\varphi}_\ell] \in \mathbb{R}^{n \times \ell}$ is a transformation matrix with $\ell << n$. The basis functions $\boldsymbol{\varphi}_i$ are the eigenvectors of a matrix $\mathbf{X}\mathbf{X}^T$ which is a low-rank approximation of the spatial covariance between the state variables over time:

$$\mathrm{cov}\left(\mathbf{x}_{1:\ell}\right) \approx \mathbf{X}\mathbf{X}^T = \left[\mathbf{x}_1 - \overline{\mathbf{x}} \quad \mathbf{x}_2 - \overline{\mathbf{x}} \quad \ldots \quad \mathbf{x}_\ell - \overline{\mathbf{x}}\right] \begin{bmatrix} \left(\mathbf{x}_1 - \overline{\mathbf{x}}\right)^T \\ \left(\mathbf{x}_2 - \overline{\mathbf{x}}\right)^T \\ \vdots \\ \left(\mathbf{x}_\ell - \overline{\mathbf{x}}\right)^T \end{bmatrix}, \tag{7.85}$$

where $\overline{\mathbf{x}} = \Sigma_{i=1}^{\ell} \mathbf{x}_i / \ell$ is the time-averaged state. Note that because of the rank deficiency of $\mathbf{X}\mathbf{X}^T$ it is sufficient to solve the eigen value problem for the much smaller matrix $\mathbf{X}^T\mathbf{X}$. The theoretical significance of the POD reduction method is that the matrix of basis functions $\mathbf{\Phi}$ can be interpreted as a low-rank approximation to the controllability Gramian of the (linearized) system equations, see e.g. Antoulas (2005). Controllability Gramians can be used to quantify the (limited) extent to which it is possible to influence the states by changing the controls (Zandvliet et al., 2008b). An attempt to benefit from the limited controllability, and from the associated low-order dynamics of the controlled system, to speed-up the flooding optimization procedure was presented by van Doren et al. (2006). They used a nested approach with an inner loop to perform the optimization in reduced-order space and outer loop to correct the approximate results from the inner loop using the high-order model. The speed-up in the inner loop results from replacing the high-order system of equations (7.1) by a low-order equivalent,

$$\mathbf{g}_k\left(\mathbf{u}_k, \mathbf{\Phi}\mathbf{z}_{k-1}, \mathbf{\Phi}\mathbf{z}_k\right) = \mathbf{0}, \quad k = 1, 2, \ldots, K, \tag{7.86}$$

which strongly reduces the size of the underlying linear systems of equations that need to be solved during the Newton-Raphson iterations in the forward simulation. Although the number of state variables was reduced drastically (from 4050 to between 20 and 100), the computational gain was limited (about 35%) because of the nonlinear nature of the problem. Nevertheless the results demonstrated that the dynamics of flooding optimization with a fixed well configuration is indeed governed by a low-order set of equations. If and how this theoretical result can be used to achieve practically relevant computational speed-ups remains a matter of further research. Promising results were recently obtained in Krogstad et al. (2009) and Cardoso et al. (2010).

7.6 Towards operational use

7.6.1 Reservoir surveillance and history matching

Within the exploration and production life cycle of an oil field various phases can be distinguished. The use of adjoint-based flooding optimization is particularly relevant to the field development phase, during which decisions are taken about the position and the number of wells and about the capacities of the surface facilities for processing of the produced and injected fluids (Sarma and Chen, 2008a; Van Essen et al., 2010). During the operational phase, surveillance of reservoir performance is usually performed by measuring pressures at the well head daily, and production rates of oil, gas and water in the individual wells more infrequently, say monthly. All these measurements are typically used in tabular form or displayed graphically to get an impression of the state of the reservoir. Moreover, they form the input for specialized semi-analytical techniques such as ‘material balance analysis’ or ‘decline curve analysis’ to extrapolate past well performance with the aim to predict future performance on a time horizon of weeks to months. However, large–scale numerical reservoir simulation traditionally plays no role during surveillance, and only appears on the stage again

after many years, say five to ten, when it is necessary to perform a 'field redevelopment' study, which often involves building a set of entirely new geological and reservoir simulation models. Calibration of these new models is usually performed using historic production data. This 'history matching' process involves adapting uncertain model parameters, such as permeabilities or porosities, until the simulated well rates and pressures match their measured counterparts in some averaged sense. Traditionally performed manually, history matching is becoming more and more dependent numerical procedures, known as 'computer-assisted history matching', and a vast number of methods have been proposed over the past decades; see e.g. Oliver et al. (2008) or Evensen (2009) for recent overviews. Parameter estimation through minimizing the mismatch between measured and simulated production data was in fact the first area of application for adjoint-based methods in the petroleum industry; see e.g. Chen et al. (1974), Chavent et al. (1975), Li et al. (2003) and Oliver et al. (2008). More recently, ensemble Kalman filtering, streamline-based methods, genetic algorithms and other techniques have become serious competitors.

7.6.2 Closed-loop reservoir management

Over the past decade a gradually increasing number of studies has proposed to combine computer-assisted history matching and flooding optimization to keep reservoir simulation models 'evergreen' such that they can also be used during the operational phase of oil field development, and not just during field development planning. This idea for 'closed-loop reservoir management' has been around for many years in different forms, often centered around attempts to improve reservoir characterization from a geosciences perspective. Moreover, recently 'closed-loop' or 'real-time' approaches to hydrocarbon production have received growing attention as part of various industry initiatives with names as 'smart fields', 'i-fields', 'e-fields', 'self-learning reservoir management' or 'integrated operations'; see Jansen et al. (2005, 2008, 2009) for references. However, whereas the focus of most of these initiatives is primarily on optimization of short-term production, 'closed-loop reservoir management' is more focused on life-cycle optimization, i.e. on processes at a timescale from years to tens of years. Moreover, in contrast to the geosciences-focused approach, we emphasize the need to focus on those elements of the modeling process that can both be verified from measurements and bear relevance to controllable parameters such as well locations or, in particular, production parameter settings. The underlying hypothesis is that

"It will be possible to significantly increase life-cycle value by changing reservoir management from a periodic to a near-continuous model-based controlled activity."

We stress that, in our view, "closed-loop" does not imply removal of human judgment from the loop. The use of model-based optimization and data assimilation techniques should result in a reduction of time spent on repetitive and tedious human activities and thus in more time that may be spent on judging results and taking decisions. We will not address closed-loop reservoir management in any further detail in these notes, and refer to Jansen et al. (2009), and Peters et al. (2010) for recent detailed information. Although the particular history matching and flooding optimization techniques used in closed-loop reservoir management are not of prime importance, we have, in line with the topic of these notes, listed several closed-loop studies that have used adjoint-based flooding optimization; see Brouwer et al. (2004), Overbeek et al. (2004), Jansen et al. (2005, 2008, 2009), Sarma et al. (2005b, 2006, 2008a), Naevdal et al. (2006), Wang et al. (2009), Chen et al. (2010), and Peters et al. (2010).

7.6.3 Robust control

One of the major obstacles on the road to industry uptake of model-based flooding optimization, whether just for field development planning or for more operational use, is the very large uncertainty of the model parameters. One of the ways to cope with this uncertainty during the field development phase of a reservoir is to use multiple subsurface models, also known as geological realizations. Van Essen et al. (2009) implemented such a robust ensemble-based optimization strategy to maximize the expected value E of the objective function J according to

$$\max_{\mathbf{u}_{1:K}} E_{\boldsymbol{\theta}}\left[J\left(\mathbf{u}_{1:K},\mathbf{y}_{1:K}^{1:N_R}\left(\mathbf{u}_{1:K}\right),\boldsymbol{\theta}^{1:N_R}\right)\right] \approx \max_{\mathbf{u}_{1:K}} \frac{1}{N_R}\sum_{i=1}^{N_R} J\left(\mathbf{u}_{1:K},\mathbf{y}_{1:K}^{i}\left(\mathbf{u}_{1:K}\right),\boldsymbol{\theta}^{i}\right), \qquad (7.87)$$

where $\boldsymbol{\theta}^i$ and $\mathbf{y}^i$ are the model parameter and output vectors of realizations $i = 1, \ldots, N_R$. The authors compared three flooding methods as applied to the hundred realizations: 1) an often used reactive water flooding strategy, where the production wells are shut-in once the water/oil ratio exceeds a preset maximum, 2) 'nominal' optimization strategies (100 in total) based on the individual realizations, and 3) the robust strategy based on maximizing the expectation as per equation (7.87). In addition the authors applied the same robust strategy to a different set of 100 realizations drawn from the same population of reservoir models, to confirm the robustness of the strategy. The results of the comparison clearly show the value of optimization compared to reactive control, and the additional benefit of a robust optimization strategy: not only is the mean recovery of the 100 simulation highest for the robust strategy, also the standard deviation is lowest. The price to pay is the need to perform forward and adjoint simulations for each realization during every iteration step in the optimization procedure.

7.6.4 Hierarchical optimization

Section 7.5.4 addressed some problems that may occur in realistically-sized flooding optimization studies because of the very large number of control variables: 1) rapidly fluctuating controls, and 2) different combinations of controls that result in nearly identical objective function values. The latter implies that there is most likely redundancy in the controls. A related issue was highlighted by Van Essen et al. (2011). They showed that an optimal trajectory may result in increased recovery or financial benefits at the end of the optimization period, but sometimes at the cost of reduced short-term benefits, at least seemingly so. Such reduced short-term gains will be a major obstacle to implementation of the optimal trajectory in practice because it will be very difficult to convince the production-oriented part of an organization to reduce a certain short-term income in favor of a much more uncertain long-term benefit. Fortunately, it turns out that the drop in short-term gains is often not an essential element of the optimal strategy but rather a result of the redundancy in the controls in combination with a long-term objective. Van Essen et al. (2011) demonstrated that it is possible to first optimize the long-term objective and thereafter, using the redundancy in the controls, a second, short-term objective. Formally this can be done by determining the Hessian of the objective function along the optimal trajectory for the primary (i.e. long-term) objective. This gives the direction in which it is possible to maximize a secondary (short-term) objective without changing the primary objective's value. A somewhat more ad-hoc, but computationally much more efficient method to achieve the same

result is to first optimize the primary objective, and thereafter alternatingly the secondary and the primary objective until convergence.

7.6.5 Multi-level optimization

Other reasons why closed-loop reservoir management, i.e. combined model-based flooding optimization and computer-assisted history matching may be difficult to implement in practice are 1) the time-consuming nature of the history matching process, 2) the poor representation of near-well bore reservoir dynamics by reservoir simulation models (because of discretization errors), the very large uncertainty in reservoir parameters (even when using frequent model updating, and even when somewhat mitigating the uncertainty with the aid of robust optimization), and 4) the cultural differences between the reservoir engineering and production engineering disciplines. Similar problems occur in the process industry and the standard solution is to use a multi-level control structure in which the results of a higher layer serve as optimal reference for the next lower layer. A similar structure has been proposed by Saputelli et al. (2006) for use in flooding optimization, but a systematic implementation of multi-level flooding optimization has not yet been reported. Van Essen et al. (2012) proposed a two-level control structure where the upper level is formed by an adjoint-based life-cycle optimization algorithm. The lower level is formed by a data-driven (black-box) response model which has a limited prediction horizon but runs very fast and can be used in an operational environment. The lower-level algorithm attempts to track the optimal output of the upper-level algorithm by manipulation of the inputs, and thus acts as a 'disturbance rejection' technique that compensates for the errors in the upper-level model. Outstanding questions include the best choice for the lower-level algorithm, and workflow aspects, and we expect that the development of practically applicable multi-level optimization techniques will be an essential prerequisite to extend the use of flooding optimization from field development planning into the operational domain.

7.6.6 Other applications

For completeness sake, we will list some other applications of adjoint-based flooding optimization. This concerns primarily the optimization of well locations and well trajectories, which is an important aspect of field development planning. Many algorithms have been proposed to solve the well location optimization problem of which the adjoint-based ones form only a small subset; see Wang and Reynolds (2007), Zandvliet et al (2008a), Sarma and Chen (2008a) and Vlemmix et al. (2009). Moreover, similar flooding optimization problems as described in these notes for oil reservoir flow occur in groundwater flow, where the objective is to clean up pollution through prolonged flushing with water, possibly in combination with surfactants or other cleaning agents. However, nearly all of the groundwater optimization references describe adjoint-free optimization methods with the exception of two papers by Merckx (1991a, 1991b). We conclude by noting that adjoint-based techniques are certainly not the only option for flooding optimization problems. Although they are extremely efficient, their implementation has two major drawbacks: 1) it requires access to the simulator code, and 2) it takes a vast programming effort, even when the Jacobians of the system equations with respect to the states are already available. Therefore various alternative flooding optimization methods have been proposed, of which we mention, without being complete, genetic algorithms and simulated annealing (Yang et al., 2003), streamline-based methods (Thiele and Batycky, 2006; Alhuthali et al. 2007, 2008, 2009), and ensemble-based methods (Lorentzen et al., 2006; Chen et al., 2009).

7.7 Ensemble optimization

7.8 References for Chapter 7

Alhuthali, A.H., Oyerinde, D. and Datta-Gupta, A., 2007: Optimal waterflood management using rate control. *SPE Reservoir Evaluation and Engineering* **10** (5) 539-551. DOI: 10.2118/102478-PA.

Alhuthali, A.H., Datta-Gupta, A., Yuen, B. and Fontanilla, J.P., 2008: Optimal rate control under geologic uncertainty. Paper SPE 113628 presented at the *SPE/DOE Symposium on Improved Oil Recovery*, Tulsa, USA, 19-23 April. DOI: 10.2118/113628-MS.

Alhuthali, A.H., Datta-Gupta, A., Yuen, B. and Fontanilla, J.P., 2009: Field applications of waterflood optimization via optimal rate control with smart wells. Paper SPE 118948 presented at the *SPE Reservoir Simulation Symposium*, The Woodlands, USA, 2-4 February. DOI: 10.2118/118948-MS.

Antoulas, A.C., 2005: *Approximation of large-scale dynamical systems*, SIAM, Philadelphia.

Asheim, H., 1988: Maximization of water sweep efficiency by controlling production and injection rates. Paper SPE 18365 presented at the *SPE European Petroleum Conference*, London, UK, October 16-18. DOI: 10.2118/18365-MS.

Brouwer, D.R., 2004: *Dynamic water flood optimization with smart wells using optimal control theory*, PhD thesis, Delft University of Technology, Delft, The Netherlands.

Brouwer, D.R. and Jansen, J.D., 2004: Dynamic optimization of water flooding with smart wells using optimal control theory. *SPE Journal* **9** (4) 391-402. DOI: 10.2118/78278-PA.

Brouwer, D.R., Naevdal, G., Jansen, J.D., Vefring, E. and van Kruijsdijk, C.P.J.W., 2004: Improved reservoir management through optimal control and continuous model updating. Paper SPE 90149 presented at the *SPE Annual Technical Conference and Exhibition*, Houston, Texas, USA, 26-29 September. DOI: 10.2118/90149-MS.

Bryson, A.E. and Ho, Y-C., 1975: *Applied optimal control*, Taylor and Francis (Hemisphere), Levittown.

Cardoso, M.A., Durlofsky, L.J. and Sarma, P., 2009: Development and application of reduced-order modeling procedures for subsurface flow simulation. *International Journal for Numerical Methods in Engineering* **77** (9) 1322-1350. DOI: 10.1002/nme.2453.

Cardoso, M.A., and Durlofsky, L.J., 2010: Use of reduced-order modeling procedures for production optimization. *SPE Journal* **15** (2) 426-435. DOI: 10.2118/119057-PA.

Chavent, G., Dupuy, M. and Lemonnier, P., 1975: History matching by use of optimal theory. *SPE Journal* **15** (1) 74-86. DOI: 10.2118/4627-PA.

Chen, W.H., Gavalas, G.R. and Wasserman, M.L., 1974: A new algorithm for automatic history matching. *SPE Journal* **14** (6) 593-608. DOI: 10.2118/4545-PA.

Chen, Y., Oliver, D.S. and Zhang, D., 2009: Efficient ensemble-based closed-loop production optimization. *SPE Journal* **14** (4) 634-645. DOI: 10.2118/112873-PA.

Chen, C., Wang, Y., Li, G. and Reynolds, A.C., 2010: Closed-loop reservoir management on the Brugge test case. *Computational Geosciences* **14** (4) 691-703. DOI: 10.1007/s10596-010-9181-7.

Chen, C., Li, G. and Reynolds, A.C., 2012: Robust constrained optimization of short- and long-term net present value for closed-loop reservoir management. *SPE Journal* **17** (3) 849-864. DOI: 10.2118/141314-PA.

De Montleau, P., Cominelli, A., Neylon, K. and Rowan, D., Pallister, I., Tesaker, O. and Nygard, I., 2006: Production optimization under constraints using adjoint gradients. *Proc. 10th European Conference on the Mathematics of Oil Recovery (ECMOR X)*, Paper A041, Amsterdam, The Netherlands, September 4-7.

Doublet, D.C., Aanonsen, S.I. and Tai, X-C, 2009: An efficient method for smart well production optimisation. *Journal of Petroleum Science and Engineering*, **69** (1-2) 25-39. DOI: 10.1016/j.petrol.2009.06.008.

Evensen, G., 2009: *Data assimilation – The ensemble Kalman filter*, 2nd ed., Springer, Berlin.

Fathi, Z. and Ramirez, W.F., 1984: Optimal injection policies for enhanced oil recovery: Part 2 – Surfactant flooding. *SPE Journal* **24** (3) 333-341. DOI: 10.2118/12814-PA.

Fathi, Z. and Ramirez, W.F., 1986: Use of optimal control theory for computing optimal injection policies for enhanced oil recovery. *Automatica* **22** (1) 33-42. DOI: 10.1016/0005-1098(86)90103-2.

Fathi, Z. and Ramirez, W.F., 1987: Optimization of an enhanced oil recovery process with boundary controls - A large-scale non-linear maximization. *Automatica* **23** (3) 301-310. DOI: 10.1016/0005-1098(87)90004-5.

Fyrozjaee, M. H. and Yortsos, Y., 2006: Control of a displacement front in potential flow using flow-rate partition. Paper SPE 99524, presented at the SPE Intelligent Energy Conference, Amsterdam, The Netherlands, 11-13 April. DOI: 10.2118/99524-MS.

Heijn, T., Markovinović, R. and Jansen, J.D., 2004: Generation of low-order reservoir models using system-theoretical concepts. *SPE Journal* **9** (2) 202-218. DOI: 10.2118/88361-PA.

Jansen, J.D., 2011: Adjoint-based optimization of multiphase flow through porous media – a review. *Computers and Fluids* **46** (1) 40-51. DOI: 10.1016/j.compfluid.2010.09.039.

Jansen, J.D., Brouwer, D.R., Nævdal, G. and van Kruijsdijk, C.P.J.W., 2005: Closed-loop reservoir management. *First Break*, January, **23**, 43-48.

Jansen, J.D., Bosgra, O.H. and van den Hof, P.M.J., 2008: Model-based control of multiphase flow in subsurface oil reservoirs. *Journal of Process Control* **18**, 846-855. DOI: 10.1016/j.jprocont.2008.06.011.

Jansen, J.D., Douma, S.G., Brouwer, D.R., Van den Hof, P.M.J., Bosgra, O.H. and Heemink, A.W., 2009: Closed-loop reservoir management. Paper SPE 119098 presented at the *SPE Reservoir Simulation Symposium*, The Woodlands, USA, 2-4 February. DOI: 10.2118/119098-MS.

Jansen, J.D., van Doren, J.F.M., Heidary-Fyrozjaee, M. and Yortsos, Y.C., 2009: Front controllability in two-phase porous media flow. In: Van den Hof, P.M.J., Scherer, C. and Heuberger, P.S.C., Eds.: *Model-based control – Bridging rigorous theory and advanced control.* Springer, 203-219.

Kourounis, D., Voskov, D. and Aziz, K., 2010: Adjoint methods for multicomponent flow simulations. *Proc. 12th European Conference on the Mathematics of Oil Recovery (ECMOR XII)*, Oxford, UK, September 6-9.

Kraaijevanger, J.F.B.M., Egberts, P.J.P., Valstar, J.R. and Buurman, H.W., 2007: Optimal waterflood design using the adjoint method. Paper SPE 105764 presented at the *SPE Reservoir Simulation Symposium*, Houston, USA, 26-28 February. DOI: 10.2118/105764-MS.

Krogstad, S., Hauge, V.L. and Gulbransen, A.F., 2009: Adjoint multiscale mixed finite elements. *SPE Journal* **16** (1) 162-171. DOI: 10.2118/119112-PA.

Landau, L.D. and Lifshitz, E.M., 1960: *Course of theoretical physics, Vol.1 (Mechanics)*, Pergamon, Oxford.

Li, R., Reynolds, A.C., and Oliver, D.S., 2003: History matching of three-phase flow production data. *SPE Journal* **8** (4): 328-340. DOI: 10.2118/87336-PA.

Lien, M., Brouwer, D.R., Manseth, T. and Jansen, J.D., 2008: Multiscale regularization of flooding optimization for smart field management. *SPE Journal* **13** (2) 195-204. DOI: 10.2118/99728-PA.

Liu, W., Ramirez, W.F. and Qi, Y.F., 1993: Optimal control of steam flooding. *SPE Advanced Technology Series* **1** (2) 73-82. DOI: 10.2118/21619-PA.

Liu, W. and Ramirez, W.F., 1994: Optimal control of three-dimensional steamflooding processes. *Journal of Petroleum Science and Engineering* **11** (2) 137-154. DOI: 10.1016/0920-4105(94)90035-3.

Lorentzen, R.J., Berg, A.M., Naevdal, G. and Vefring, E.H., 2006: A new approach for dynamic optimization of waterflooding problems. Paper SPE 99690 presented at the *SPE Intelligent Energy Conference and Exhibition*, Amsterdam, The Netherlands, 25-27 February. DOI: 10.2118/99690-MS.

Luenberger, D.G. and Ye, Y., 2010: *Linear and nonlinear programming, 3rd ed.* Springer, New York.

Mehos, G.J. and Ramirez, W.F., 1989: Use of optimal control theory to optimize carbon dioxide miscible-flooding enhanced oil recovery. *Journal of Petroleum Science and Engineering* **2** (4) 247-260. DOI: 10.1016/0920-4105(89)90002-8.

Merckx, C., 1991a: Non-linear programming in groundwater decontamination problems. *Engineering Optimization* **18** (1) 121-136. DOI: 10.1080/03052159108941016.

Merckx, C., 1991b: Optimal pumping strategy for groundwater decontamination. *International Journal of Control* **53** (4) 889-905. DOI: 10.1080/00207179108953655.

Naevdal, G., Brouwer, D.R. and Jansen, J.D., 2006: Waterflooding using closed-loop control. *Computational Geosciences* **10** (1) 37-60. DOI: 10.1007/s10596-005-9010-6.

Oliver, D.S., Reynolds, A.C. and Liu, N., 2008: *Inverse theory for petroleum reservoir characterization and history matching*, Cambridge University Press, Cambridge.

Overbeek, K.M., Brouwer, D.R., Naevdal, G., van Kruijsdijk, C.P.J.W. and Jansen, J.D., 2004: Closed-loop waterflooding. *Proc. 9th European Conference on Mathematics in Oil Recovery (ECMOR IX)*, Cannes, France, 28 August - 2 September.

Peters, L., Arts, R.J., Brouwer, G.K., Geel, C.R., Cullick, S., Lorentzen, R.J., Chen, Y., Dunlop, K.N.B., Vossepoel, F.C., Xu, R., Sarma, P., Alhuthali, A.H. and Reynolds, A.C., 2010: Results of the Brugge benchmark study for flooding optimization and history matching. *SPE Reservoir Evaluation and Engineering* **13** (3) 391-405. DOI: 10.2118/119094-PA.

Ramakrishnan, T.S., 2007: On reservoir fluid-flow control with smart completions. *SPE Production and Operations* **22** (1) 4-12. DOI: 10.2118/84219-PA.

Ramirez, W.F., 1987: *Application of optimal control theory to enhanced oil recovery,* Elsevier, Amsterdam.

Ramirez, W.F., Fathi, Z. and Cagnol, J.L., 1984: Optimal injection policies for enhanced oil recovery: Part 1 – Theory and computational strategies. *SPE Journal* **24** (3) 328-332. DOI: 10.2118/11285-PA.

Rao, S.S., 1996: *Engineering optimization, 3rd ed.*, Wiley, New York.

Rodrigues, J.R.P., 2006: Calculating derivatives for automatic history matching. *Computational Geosciences* **10** (1) 119-136. DOI: 10.1007/s10596-005-9013-3.

Saputelli, L., Nikolaou, M. and Economides, M.J., 2006: Real-time reservoir management: a multi-scale adaptive optimization and control approach. *Computational Geosciences* **10** (1) 61-96. DOI: 10.1007/s10596-005-9011-5.

Sarma, P. and Chen, W.H., 2008a: Efficient well placement optimization with gradient-based algorithms and adjoint models. Paper SPE 112257 presented at the *SPE Intelligent Energy Conference and Exhibition*, Amsterdam, The Netherlands, 25-27 February. DOI: 10.2118/112257-MS.

Sarma, P. and Chen, W.H., 2008b: Applications of optimal control theory for efficient production optimization of realistic reservoirs. Paper IPTC 12480 presented at the *International Petroleum Technology Conference*, Kuala Lumpur, Malaysia, 3-5 December.

Sarma, P., Aziz, K. and Durlofsky, L.J., 2005a: Implementation of adjoint solution for optimal control of smart wells. Paper SPE 92864 presented at the *SPE Reservoir Simulation Symposium*, Houston, USA, 31 January – 2 February. DOI: 10.2118/92864-MS.

Sarma, P., Durlofsky, L.J. and Aziz, K., 2005b: Efficient closed-loop production optimization under uncertainty. Paper SPE 94241 presented at the *SPE Europec/EAGE Annual Conference*, Madrid, Spain, 13-16 June. DOI: 10.2118/94241-MS.

Sarma, P., Durlofsky, L.J., Aziz, K., Chen, W.H., 2006: Efficient real-time reservoir management using adjoint-based optimal control and model updating. *Computational Geosciences* **10** (1) 3-36. DOI: 10.1007/s10596-005-9009-z.

Sarma, P., Durlofsky, L.J. and Aziz, K., 2008a: Computational techniques for closed-loop reservoir modeling with application to a realistic reservoir. *Petroleum Science and Technology* **26** (10 & 11) 1120-1140. DOI: 10.1080/10916460701829580.

Sarma, P., Chen, W.H. Durlofsky, L.J. and Aziz, K., 2008b: Production optimization with adjoint models under nonlinear control-state path inequality constraints. *SPE Reservoir Evaluation and Engineering* **11** (2) 326-339. DOI: 10.2118/99959-PA.

Stengel, R.F., 1986: *Stochastic optimal control: theory and application*, Wiley, New York. Reprinted as *Optimal control and estimation* in 1994 by Dover, New York.

Sudaryanto, B. and Yortsos, Y.C., 2000: Optimization of fluid front dynamics in porous media using rate control. *Physics of Fluids* **12** (7) 1656-1670. DOI: 10.1063/1.870417.

Sudaryanto, B. and Yortsos, Y.C., 2001: Optimization of displacements in porous media using rate control. Paper SPE 71509 presented at the *SPE Annual Technical Conference and Exhibition*, New Orleans, Louisiana, USA, 30 September - 3 October. DOI: 10.2118/71509-MS.

Suwartadi, E., Krogstad, S. and Foss, B., 2009: On state constraints of adjoint optimization in oil reservoir water-flooding. Paper SPE 125557 presented at the *SPE Reservoir Characterization and Simulation Conference*, Abu Dhabi, UAE, 19-21 October. DOI: 10.2118/125557-MS.

Suwartadi, E., Krogstad, S. and Foss, B., 2010: Nonlinear output constraints handling for production optimization. *Proc. 12th European Conference on the Mathematics of Oil Recovery (ECMOR XII)*, Oxford, UK, September 6-9.

Thiele, M.R. and Batycky, R.P., 2006: Using streamline-derived injection efficiencies for improved waterflood management. *SPE Reservoir Evaluation and Engineering* **9** (2) 187-196. DOI: 10.2118/84080-PA.

Vakili, A., Jansen, J.D., Esmaiel, T. and van Kruijsdijk, C.P.J.W., 2005: On the adjoint of a nonlinear diffusion-convection equation to describe flow in porous media. Paper SPE 93566 presented at the *14th SPE Middle East Oil & Gas Show and Conference*, Bahrain, 12-15 March. DOI: 10.2118/93566-MS.

Van Doren, J.F.M., Markovinović, R. and Jansen, J.D., 2006: Reduced-order optimal control of water flooding using POD. *Computational Geosciences* **10** (1) 137-158. DOI: 10.1007/s10596-005-9014-2.

Van Essen, G.M., Zandvliet, M.J., Van den Hof, P.M.J., Bosgra, O.H. and Jansen, J.D., 2009: Robust water flooding optimization of multiple geological scenarios. *SPE Journal* **14** (1) 202-210. DOI: 10.2118/102913-PA.

Van Essen, G.M., Jansen, J.D., Brouwer, D.R. Douma, S.G., Zandvliet, M.J., Rollett, K.I. and Harris, D.P., 2010: Optimization of smart wells in the St. Joseph field. *SPE Reservoir Evaluation and Engineering* **13** (4) 588-595. DOI: 10.2118/123563-PA.

Van Essen, G.M., Van den Hof, P.M.J. and Jansen, J.D., 2011: Hierarchical long-term and short-term production optimization. *SPE Journal* **16** (1) 191-199. DOI: 10.2118/124332-PA.

Van Essen G.M., Van den Hof, P.M.J. and Jansen, J.D., 2012: A two-level strategy to realize life-cycle production optimization in an operational setting. Paper SPE149736 presented at the *SPE Intelligent Energy Conference*, Utrecht, The Netherlands, 27–29 March.

Vermeulen, P.T.M., Heemink, A.W. and Te Stroet, C.B.M., 2004: Reduced models for linear groundwater flow models using empirical orthogonal functions. *Advances in water resources* **27** 57-69. DOI: 10.1016/j.advwatres.2003.09.008.

Virnovski, G.A., 1991: Water flooding strategy design using optimal control theory, *Proc. 6th European Symposium on IOR*, Stavanger, Norway, 437-446.

Vlemmix, S., Joosten, G.J.P., Brouwer, D.R. and Jansen, J.D., 2009: Adjoint-based well trajectory optimization in a thin oil rim. Paper SPE 121891 presented at the *SPE European Petroleum Conference / EAGE Annual Conference and Exhibition* held in Amsterdam, The Netherlands, 8–11 June.

Wang, C., Li, G. and Reynolds, A.C., 2007: Optimal well placement for production optimization. Paper SPE 111154 presented at the *SPE Eastern Regional Meeting*, Lexington, USA, 11-14 October. DOI: 10.2118/111154-MS.

Wang, C., Li, G. and Reynolds, A.C., 2009: Production optimization in closed-loop reservoir management. *SPE Journal* **14** (3) 506-523. DOI: 10.2118/109805-PA.

Yang, D., Zhang, Q., Gua, Y., 2003: Integrated optimization and control of the production-injection operation systems for hydrocarbon reservoirs. *Journal of Petroleum Science and Engineering* **37** (1 & 2) 69-81. DOI: 10.1016/S0920-4105(02)00311-X.

Zakirov, I.S., Aanonsen, S.I., Zakirov, E.S., and Palatnik, B.M., 1996: Optimization of reservoir performance by automatic allocation of well rates. *Proc. 5th European Conference on the Mathematics of Oil Recovery (ECMOR V)*, Leoben, Austria.

Zandvliet, M.J., Bosgra, O.H., Jansen, J.D., Van den Hof, P.M.J. and Kraaijevanger, J.F.B.M., 2007: Bang-bang control and singular arcs in reservoir flooding. *Journal of Petroleum Science and Engineering* **58** (1 & 2) 186-200. DOI: 10.1016/j.petrol.2006.12.008.

Zandvliet, M.J., Handels, M., Van Essen, G.M., Brouwer, D.R. and Jansen, J.D., 2008a: Adjoint-based well placement optimization under production constraints. *SPE Journal* **13** (4) 392-399. DOI: 10.2118/105797-PA.

Zandvliet, M.J., van Doren, J.F.M., Bosgra, O.H., Jansen, J.D. and van den Hof, P.M.J., 2008b: Controllability, observability and identifiability in single-phase porous media flow. *Computational Geosciences* **12** (4) 605-622. DOI: 10.1007/s10596-008-9100-3.

8 Data assimilation

8.1 State and parameter estimation

In this chapter we will focus on the use of measured data to reduce the uncertainty in subsurface models; see Figure 8.1. In particular we will consider *data assimilation* techniques to combine information from (noisy) measurements with information from (uncertain) prior knowledge to obtain state and parameter estimates that are optimal in some pre-defined sense. In the petroleum industry data assimilation is often referred to as *computer-assisted history matching* or even *automatic history matching*. The latter term, however, incorrectly suggest the absense of human involvement in the data assimilation process. We already addressed aspects of *state estimation* in Chapters 4 and 5 where we treated the concepts of observability and controllability of state variables in some detail. The concept of *parameter identifiability* appears at first sight to be more closely related to observability than to controllability. However, as will be shown below, in order to identify a parameter from an output, we often need to actively disturb the input, which implies that controllability has to play a role as well. This becomes even more the case for the broader concept of *system identifiability*, also referred to as *structural identifiability*, which involves inferring information about the *model structure* from the input-output behavior of the system. Here, the term model structure refers to set of equations that govern the behavior of the system, including e.g. the order of the differential equations. *System identification*, also referred to as *realization theory* in mathematical systems theory, is a well developed topic in the measurement and control community with a wide range of applications, see e.g. Ljung (1999). The more narrow identification problem, in which it is assumed that the model structure is largely known but that there is uncertainty about the value of the parameters, is also well developed and is often referred to as *parameter estimation* or *inverse modeling*; see e.g. Tarantola (2005)[†]. In line with the developments in the previous chapters, we will assume the model structure to be known and therefore concentrate on state and parameter estimation.

[†] This terminology is not unique, and some authors refer to the parameters as the model and to the governing equations as the mathematical model.

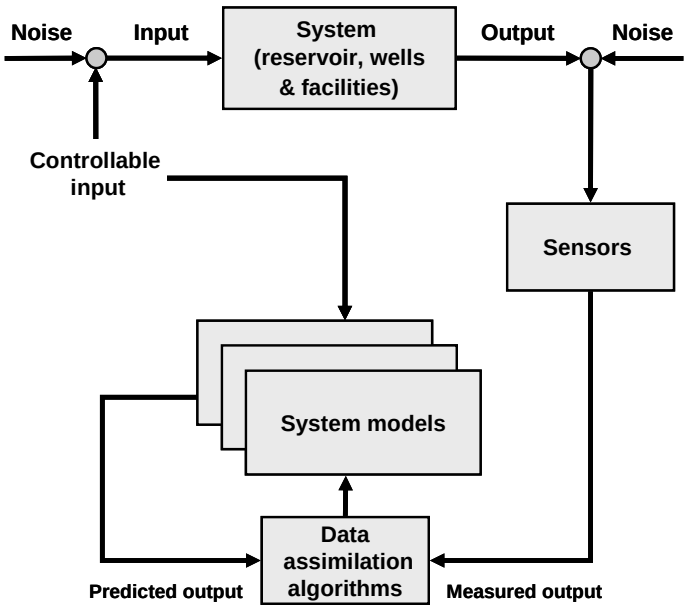


Figure 8.1: Data assimilation.

8.2 Problem statement

8.2.1 Governing equations

Consider an implicitly time-discretized nonlinear system equation:

$$\mathbf{g}_k\left(\mathbf{u}_k, \mathbf{x}_{k-1}, \mathbf{x}_k, \boldsymbol{\theta}, \boldsymbol{\varepsilon}_k\right) = \mathbf{0} \ , \quad k = 1, 2, \ldots, K \ , \tag{8.1}$$

with uncertain initial conditions

$$\mathbf{x}_0 = \breve{\mathbf{x}}_0 \, , \tag{8.2}$$

where k is discrete time, $\mathbf{u}_k$ is the input vector, $\mathbf{x}_k$ is the state vector, $\boldsymbol{\theta}$ is a vector of time-independent parameters, and $\boldsymbol{\varepsilon}_k$ a vector of time-dependent *model errors*. In addition to the assumptions made for equation (7.1), which were used to state the flooding optimization problem, we assume that the initial conditions and possibly also some of the inputs are poorly known. Moreover, we have introduced two new vectors of uncertain parameters: $\boldsymbol{\theta}$ and $\boldsymbol{\varepsilon}_k$. Of these, the parameter vector $\boldsymbol{\theta} \in \mathbb{R}^q$ appears naturally because we are considering a parameter estimation problem. We assume that the parameters have been scaled such that their domain is the set of real numbers, as will be discussed in more detail below. The vector of model errors $\boldsymbol{\varepsilon}_k$, assumed to be additive to $\mathbf{x}_k$, reflects two sources of uncertainty. In the first place it allows for deviations between the model equations $\mathbf{g}_k$ and the real system, resulting from ignorance, assumptions and simplifications during the modeling process. Secondly, it allows for the effects of unknown inputs, indicated as 'noise' in the top left corner of Figure 8.1. For the case of nonlinear system equations that can be expressed in matrix form, such as e.g. in equation (4.81), equation (8.1) can be rewritten as

$$\mathbf{x}_k = \mathbf{A}_d\left(\mathbf{x}_k, \boldsymbol{\theta}\right)\mathbf{x}_{k-1} + \mathbf{B}_d\left(\mathbf{x}_k, \boldsymbol{\theta}\right)\mathbf{u}_k + \boldsymbol{\varepsilon}_k \ , \quad k = 1, 2, \ldots, K \ . \tag{8.3}$$

We could have introduced the model errors in Chapter 6 also, but, instead, we treated uncertainty through the use of ensembles of models rather than through the addition of an error vector. In the present chapter we will continue the use of ensembles, but, in addition, use the concept of model errors as a necessary step to explain some of the data assimilation

techniques. We assume that the output vector $\mathbf{y}_k$ (i.e. the vector of predicted measurements) is a nonlinear function of the inputs and the states:

$$\mathbf{j}_k\left(\mathbf{u}_{k,}\mathbf{x}_k,\mathbf{y}_k\right)=\mathbf{0}\ . \tag{8.4}$$

For the case of an LTI measurement operator, as often encountered in reservoir simulation, equation (8.4) can be rewritten as

$$\mathbf{y}_k=\mathbf{C}\mathbf{x}_k+\mathbf{D}\mathbf{u}_k\ , \tag{8.5}$$

The actual 'measured measurements' will be indicated with $\mathbf{d}_k$. Generally the predicted measurements will not be equal to the actual measurements such that we can write

$$\mathbf{d}_k-\mathbf{y}_k=\boldsymbol{\eta}_k\ , \tag{8.6}$$

where $\boldsymbol{\eta}_k$ is a vector of *measurement errors* which, like $\boldsymbol{\varepsilon}_k$, reflects two sources of uncertainty. It allows for deviations between the predicted and the actual measurements resulting from imperfections in the modeling process, and for the effects of unknown inputs, indicated as 'noise' in the top right corner of Figure 8.1. The uncertainty in the initial conditions, the inputs, the parameters, the model errors and the measurement errors is represented by covariance matrices $\mathbf{P}_{x_0}$, $\mathbf{P}_u$, $\mathbf{P}_\theta$, $\mathbf{P}_{x_k}$ and $\mathbf{P}_y$ respectively. We assume that the inputs $\mathbf{u}_k$, the initial condition $\mathbf{x}_0$, and the parameters $\boldsymbol{\theta}$ are only known approximately and have to be estimated, but that their second-order error statistics, i.e. their various covariance matrices are known. The model errors $\boldsymbol{\varepsilon}_{1:K}$ are initially taken as $\mathbf{0}^\dagger$. The corresponding covariance matrix etc. etc. 4D-VAR.

8.2.2 Minimization problem

As a first step we can now reformulate our data assimilation problem as a minimization problem; see e.g. Bennett (1992, 2002), Tarantola (2005), Aster et al. (2005), Lewis et al. (2006), Oliver et al. (2008), or Evensen (2009). In that case we have to minimize the value of an objective function $\mathcal{J}$ defined as:

$$\mathcal{J}\left(\mathbf{u}_{1:K},\mathbf{x}_0,\mathbf{y}_{1:K}\left(\mathbf{u}_{1:K},\mathbf{x}_0,\boldsymbol{\theta},\boldsymbol{\varepsilon}_{1:K}\right),\boldsymbol{\theta},\boldsymbol{\varepsilon}_{1:K}\right)\triangleq\sum_{k=1}^{K}\mathcal{J}_k\left(\mathbf{u}_k,\mathbf{x}_0,\mathbf{y}_k,\boldsymbol{\theta},\boldsymbol{\varepsilon}_k\right), \tag{8.7}$$

where

$$\begin{aligned}\mathcal{J}_k\left(\mathbf{u}_k,\mathbf{x}_0,\mathbf{y}_k,\boldsymbol{\theta},\boldsymbol{\varepsilon}_k\right)\triangleq&\left(\mathbf{d}_k-\mathbf{y}_k\right)^T\mathbf{P}_y^{-1}\left(\mathbf{d}_k-\mathbf{y}_k\right)\\&+\left(\mathbf{u}_k-\breve{\mathbf{u}}_k\right)^T\mathbf{P}_u^{-1}\left(\mathbf{u}_k-\breve{\mathbf{u}}_k\right)\\&+\left(\mathbf{x}_0-\breve{\mathbf{x}}_0\right)^T\mathbf{P}_{x_0}^{-1}\left(\mathbf{x}_0-\breve{\mathbf{x}}_0\right)\delta_{k-1}\\&+\left(\boldsymbol{\theta}-\breve{\boldsymbol{\theta}}\right)^T\mathbf{P}_\theta^{-1}\left(\boldsymbol{\theta}-\breve{\boldsymbol{\theta}}\right)\delta_{k-1}\\&+\boldsymbol{\varepsilon}_k^T\mathbf{P}_{x_k}^{-1}\boldsymbol{\varepsilon}_k\end{aligned} \tag{8.8}$$

and where K is the total number of time steps, $\mathcal{J}_k$ represents the contribution to $\mathcal{J}$ in each time step k, while $\breve{\mathbf{u}}_{1:K}$, $\breve{\mathbf{x}}_0$ and $\breve{\boldsymbol{\theta}}$ are vectors of *prior* values, i.e. initial estimates, of the

† If we would have an idea of the magnitude of the errors $\boldsymbol{\varepsilon}_{1:K}$ and $\boldsymbol{\eta}_{1:K}$, we could adapt the model and measurement equations to make the errors $\mathbf{0}$ again.

initial state, inputs and parameters[†]. The Kronecker delta δ_k ensures that the time-independent parameters $\mathbf{x}_0$ and $\boldsymbol{\theta}$ are included in the summation. Note that actually the initial condition and all inputs and model errors up to time k may play a role in $\mathcal{J}_k$ in equation (8.7) as can be seen through recursive application of equations (8.1) and (8.4). We should therefore formally write $\mathcal{J}_k(\mathbf{u}_k, \mathbf{x}_0, \mathbf{y}_k(\mathbf{u}_k, \mathbf{x}_{1:k}(\mathbf{u}_{1:k}, \mathbf{x}_0, \boldsymbol{\theta}, \boldsymbol{\varepsilon}_{1:k})), \boldsymbol{\theta}, \boldsymbol{\varepsilon}_k)$, but to keep the notation tractable we use $\mathcal{J}_k(\mathbf{u}_k, \mathbf{x}_0, \mathbf{y}_k, \boldsymbol{\theta}, \boldsymbol{\varepsilon}_k)$ instead. The first term at the right-hand side of equation (8.8) may be rewritten in terms of the measurement error as

$$\left(\mathbf{d}_k - \mathbf{y}_k\right)^T \mathbf{P}_y^{-1} \left(\mathbf{d}_k - \mathbf{y}_k\right) \equiv \boldsymbol{\eta}_k^T \mathbf{P}_{y_k}^{-1} \boldsymbol{\eta}_k \,. \tag{8.9}$$

The next three terms can be rewritten as

$$\left(\mathbf{u}_k - \breve{\mathbf{u}}_k\right)^T \mathbf{P}_u^{-1} \left(\mathbf{u}_k - \breve{\mathbf{u}}_k\right) \equiv \tilde{\mathbf{u}}_k^T \mathbf{P}_{u_k}^{-1} \tilde{\mathbf{u}}_k \,, \tag{8.10}$$

$$\left(\mathbf{x}_0 - \breve{\mathbf{x}}_0\right)^T \mathbf{P}_{x_0}^{-1} \left(\mathbf{x}_0 - \breve{\mathbf{x}}_0\right) \equiv \tilde{\mathbf{x}}_0^T \mathbf{P}_{x_0}^{-1} \tilde{\mathbf{x}}_0 \,, \tag{8.11}$$

$$\left(\boldsymbol{\theta} - \breve{\boldsymbol{\theta}}\right)^T \mathbf{P}_\theta^{-1} \left(\boldsymbol{\theta} - \breve{\boldsymbol{\theta}}\right) \equiv \tilde{\boldsymbol{\theta}}^T \mathbf{P}_\theta^{-1} \tilde{\boldsymbol{\theta}} \,, \tag{8.12}$$

where

$$\tilde{\mathbf{u}}_k \triangleq \mathbf{u}_k - \breve{\mathbf{u}}_k \,, \;\; \tilde{\mathbf{x}}_0 \triangleq \mathbf{x}_0 - \breve{\mathbf{x}}_0 \text{ and } \tilde{\boldsymbol{\theta}} \triangleq \boldsymbol{\theta} - \breve{\boldsymbol{\theta}} \tag{8.13, 8.14, 8.15}$$

are the deviations from the prior values of the respective parameters. We can obtain a more compact notation by defining the generalized parameter vector

$$\mathbf{m}_k \triangleq \begin{bmatrix} \mathbf{u}_k \\ \mathbf{x}_0 \delta_{k-1} \\ \boldsymbol{\theta} \delta_{k-1} \\ \boldsymbol{\varepsilon}_k \end{bmatrix}, \tag{8.16}$$

the corresponding prior vector

$$\breve{\mathbf{m}}_k \triangleq \begin{bmatrix} \breve{\mathbf{u}}_k \\ \breve{\mathbf{x}}_0 \delta_{k-1} \\ \breve{\boldsymbol{\theta}} \delta_{k-1} \\ \mathbf{0} \end{bmatrix}, \tag{8.17}$$

and the generalized parameter covariance matrix

$$\mathbf{P}_{m_k} \triangleq \begin{bmatrix} \mathbf{P}_u & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{P}_{x_0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{P}_\theta & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{P}_{x_k} \end{bmatrix}, \tag{8.18}$$

such that equations (8.7) and (8.8) reduce to

$$\mathcal{J}\left(\mathbf{m}_{1:K}, \mathbf{y}_{1:K}\left(\mathbf{m}_{1:K}\right)\right) \triangleq \sum_{k=1}^{K} \mathcal{J}_k\left(\mathbf{m}_k, \mathbf{y}_k\right), \tag{8.19}$$

[†] In meteorology, the prior initial condition $\mathbf{x}_0$ is often referred to as the *background state*.

where

$$\begin{aligned}\mathcal{J}_k\left(\mathbf{m}_k,\mathbf{y}_k\right) \triangleq & \left(\mathbf{d}_k-\mathbf{y}_k\right)^T \mathbf{P}_y^{-1}\left(\mathbf{d}_k-\mathbf{y}_k\right) \\ & +\left(\mathbf{m}_k-\breve{\mathbf{m}}_k\right)^T \mathbf{P}_{m_k}^{-1}\left(\mathbf{m}_k-\breve{\mathbf{m}}_k\right),\end{aligned} \tag{8.20}$$

for $k=1,\ldots,K$. The data assimilation problem can now be formulated as

$$\min_{\mathbf{m}_{1:K}} \mathcal{J}\left(\mathbf{m}_{1:K},\mathbf{y}_{1:K}\left(\mathbf{m}_{1:K}\right)\right) , \tag{8.21}$$

subject to

- system equation (8.1), and
- output equation (8.4),

which can now be expressed as

$$\mathbf{g}_k\left(\mathbf{m}_k,\mathbf{x}_{k-1},\mathbf{x}_k\right)=\mathbf{0} , \tag{8.22}$$

$$\mathbf{j}_k\left(\mathbf{m}_k,\mathbf{x}_k,\mathbf{y}_k\right)=\mathbf{0} . \tag{8.23}$$

Unlike in the flooding optimization problem, defined in equation (7.8), we do not aim at finding the optimum subject to a known initial condition. In the present case the initial condition is uncertain and therefore not taken into account as a *strong constraint* but, instead, as a *weak constraint* through its presence as a term in the objective function (8.7), just like the uncertain inputs, parameters and model errors. Although the minimization problem as defined in expression (8.21) contains both strong and weak constraints, it is, somewhat confusingly, known as a *weak constraint problem* because of the presence of the error terms $\boldsymbol{\varepsilon}_{1:K}$ in the objective function (8.7) which allow for an approximate solution of the state equations $\mathbf{g}_{1:K}$. The case without model error, i.e. when $\boldsymbol{\varepsilon}_{1:K}=\mathbf{0}$, is then known as a *strong constraint problem*.

8.2.3 Parameter scaling

As mentioned above, we assume that the values of the uncertain parameters have been scaled such that their values are in the set of real numbers $\mathbb{R}$. E.g. permeability values k, which have a physical dimension of m^2 and can only have positive values, may be rescaled as

$$\tilde{k}=\ln\left(\frac{k}{k_{ref}}\right) , \tag{8.24}$$

where k_{ref} is an appropriate reference value. Porosity values ϕ, which are dimensionless but are restricted to values between zero and one, may be rescaled as

$$\tilde{\phi}=\ln\left[-\ln\left(\phi\right)\right] , \tag{8.25}$$

or as[‡]

$$\tilde{\phi}=\operatorname{atan}\left[\pi\left(\phi-\frac{1}{2}\right)\right] . \tag{8.26}$$

[‡] The choice of the transformation becomes particularly important if the uncertainty in the parameters has been quantified statistically, i.e. with a probability density function (pdf), because the scaling will also influence the shape of the pdf.

8.3 Variational data assimilation

8.3.1 Optimal control theory

Following the same approach as used to solve the flooding optimization problem in Section 6.4.3, we define a modified objective function

$$\bar{\mathcal{J}}\left(\mathbf{m}_{1:K},\mathbf{x}_{0:K},\mathbf{y}_{1:K},\boldsymbol{\lambda}_{1:K},\boldsymbol{\mu}_{1:K}\right)\triangleq\sum_{k=1}^{K}\begin{bmatrix}\frac{1}{2}\mathcal{J}_k\left(\mathbf{m}_k,\mathbf{y}_k\right)\\+\boldsymbol{\lambda}_k^T\mathbf{g}_k\left(\mathbf{m}_k,\mathbf{x}_{k-1},\mathbf{x}_k\right)\\+\boldsymbol{\mu}_k^T\mathbf{j}_k\left(\mathbf{m}_k,\mathbf{x}_k,\mathbf{y}_k\right)\end{bmatrix} \tag{8.27}$$

where the 'system constraint' $\mathbf{g}_{1:K}=\mathbf{0}$, and the 'output constraint' $\mathbf{j}_{1:K}=\mathbf{0}$ have been 'adjoined' to $\mathcal{J}_{1:K}$ with the aid of vectors of Lagrange multipliers $\boldsymbol{\lambda}_{1:K}$ and $\boldsymbol{\mu}_{1:K}$ respectively. The factor ½ in front of the term $\mathcal{J}_k(\mathbf{m}_k,\mathbf{y}_k)$ has been introduced to cancel out a factor 2 that will appear later when we will differentiate this term. Just as in Section 6.4.3 we may find the necessary conditions for a minimum through taking the first variation, and then use the resulting equations to develop a scheme to iteratively compute the minimum; which explains the term *variational data assimilation* in the heading of this section. Thus, requiring stationarity of $\delta\bar{\mathcal{J}}$ with respect to variations in $\mathbf{m}_{1:K},\mathbf{x}_{0:K},\mathbf{y}_{1:K},\boldsymbol{\lambda}_{1:K}$ and $\boldsymbol{\mu}_{1:K}$ respectively, and splitting off the derivatives for $k = 0$ and $k = K$, we obtain the following set of equations:

$$\frac{\partial\bar{\mathcal{J}}}{\partial\mathbf{m}_k}\equiv\left(\mathbf{m}_k-\breve{\mathbf{m}}_k\right)^T\mathbf{P}_{m_k}^{-1}+\boldsymbol{\lambda}_k^T\frac{\partial\mathbf{g}_k}{\partial\mathbf{m}_k}+\boldsymbol{\mu}_k^T\frac{\partial\mathbf{j}_k}{\partial\mathbf{m}_k}=\mathbf{0}^T,\quad k=1,2,\ldots,K, \tag{8.28}$$

$$\frac{\partial\bar{\mathcal{J}}}{\partial\mathbf{x}_k}\equiv\boldsymbol{\lambda}_{k+1}^T\frac{\partial\mathbf{g}_{k+1}}{\partial\mathbf{x}_k}+\boldsymbol{\lambda}_k^T\frac{\partial\mathbf{g}_k}{\partial\mathbf{x}_k}+\boldsymbol{\mu}_k^T\frac{\partial\mathbf{j}_k}{\partial\mathbf{x}_k}=\mathbf{0}^T, \tag{8.29}$$

$$\frac{\partial\bar{\mathcal{J}}}{\partial\mathbf{x}_K}\equiv\boldsymbol{\lambda}_K^T\frac{\partial\mathbf{g}_K}{\partial\mathbf{x}_K}+\boldsymbol{\mu}_K^T\frac{\partial\mathbf{j}_K}{\partial\mathbf{x}_K}=\mathbf{0}^T, \tag{8.30}$$

$$\frac{\partial\bar{\mathcal{J}}}{\partial\mathbf{y}_k}\equiv\left(\mathbf{d}_k-\mathbf{y}_k\right)^T\mathbf{P}_{\eta_k}^{-1}+\boldsymbol{\mu}_k^T\frac{\partial\mathbf{j}_k}{\partial\mathbf{y}_k}=\mathbf{0}^T, \tag{8.31}$$

$$\frac{\partial\bar{\mathcal{J}}}{\partial\boldsymbol{\lambda}_k}\equiv\mathbf{g}^T\left(\mathbf{m}_k,\mathbf{x}_{k-1},\mathbf{x}_k\right)=\mathbf{0}^T, \tag{8.32}$$

$$\frac{\partial\bar{\mathcal{J}}}{\partial\boldsymbol{\mu}_k}\equiv\mathbf{j}^T\left(\mathbf{m}_k,\mathbf{x}_k,\mathbf{y}_k\right)=\mathbf{0}^T. \tag{8.33}$$

Just as in Section 6.4.3, the Euler-Lagrange equations (8.29) to (8.33) are the first-order necessary conditions for a minimum. Also their interpretations are similar to those given in Section 6.4.3. In particular, equations (8.33) and (8.32) are identical to output equation (8.23) and system equation (8.22). Equations (8.31), (8.30) and (8.29) allow us to compute first the Lagrange multipliers $\boldsymbol{\mu}_{1:K}$, then $\boldsymbol{\lambda}_K$ for the final discrete time K, and thereafter $\boldsymbol{\lambda}_k$ for times $k=K-1,K-2,\ldots,1$. Finally, equation (8.28) represents the effect of changing the values of the uncertain generalized model parameters $\mathbf{m}_{1:K}$ on the value of the objective function while keeping all other variables fixed. Outside a minimum these terms are not equal to zero, but then they give us information on how to adapt the values of the uncertain variables to make the terms zero. Alternatively, the terms may be used to compute gradients required to obtain

the minimum by adapting the values of the uncertain variables with the aid of a gradient-based minimization algorithm, as will be discussed in more detail below.

8.3.2 Computation of the uncertain parameters

Iterative solution

We can take the following steps to iteratively adapt the values of $\mathbf{m}_{1:K}$ until $\bar{\mathcal{J}}$ has been decreased to below a preset limit:

Algorithm8.1

1) Choose prior values $\breve{\mathbf{m}}_{1:K}$ for the uncertain generalized parameters.
2) Compute the states $\mathbf{x}_{1:K}$ and outputs $\mathbf{y}_{1:K}$ from equations (8.32) and (8.33).
3) Compute the Lagrange multipliers $\boldsymbol{\mu}_{1:K}$ and $\boldsymbol{\lambda}_{1:K}$ from equations (8.31), (8.30) and (8.29).
4) Compute new values for the uncertain generalized parameters $\mathbf{m}_{1:K}$ from equation (8.28) according to

$$\mathbf{m}_k^T = \breve{\mathbf{m}}_k^T - \boldsymbol{\lambda}_k^T \frac{\partial \mathbf{g}_k}{\partial \mathbf{m}_k} \mathbf{P}_{m_k} - \boldsymbol{\mu}_k^T \frac{\partial \mathbf{j}_k}{\partial \mathbf{m}_k} \mathbf{P}_{m_k}, \qquad k = 1, 2, \ldots, K, \tag{8.34}$$

5) Compute $\bar{\mathcal{J}}$ and check for convergence. If converged stop, else return to step 2.

Computational issues

- During the iteration process, the Jacobians $\partial \mathbf{g}/\partial \mathbf{x}_{0:K}$, $\partial \mathbf{j}/\partial \mathbf{x}_{1:K}$ and $\partial \mathbf{j}/\partial \mathbf{y}_{1:K}$ need to be recomputed using the most recent iterates of $\mathbf{x}_{0:K}$ and $\mathbf{y}_{1:K}$, such that in the limit of convergence the derivatives are computed exactly in the stationary point.
- Within the iterative process another iterative process takes place: step 2, i.e. the 'forward' integration, is performed with the aid of an iterative time stepping procedure because the system equations $\mathbf{g}_k$, and occasionally also the output equations $\mathbf{j}_k$, are nonlinear. Step 3, the 'backward' integration, can be performed with a non-iterative method because equations (8.32) and (8.33) are linear.
- Step 4 can be replaced by an approach to compute the derivatives $d\mathcal{J}/d\mathbf{m}_k \equiv \partial \bar{\mathcal{J}}/\partial \mathbf{m}_k$ as follows. If, instead of computing new values for the uncertain variables $\mathbf{m}_k$ from equation (8.28) (i.e. as in equation (8.34)), we insert the current values, we can use the residual of equation (8.28) to compute derivatives according to

$$\frac{d\mathcal{J}}{d\mathbf{m}_k} \equiv \frac{\partial \bar{\mathcal{J}}}{\partial \mathbf{m}_k} = \left(\mathbf{m}_k - \breve{\mathbf{m}}_k\right)^T \mathbf{P}_{m_k}^{-1} + \boldsymbol{\lambda}_k^T \frac{\partial \mathbf{g}_k}{\partial \mathbf{m}_k} + \boldsymbol{\mu}_k^T \frac{\partial \mathbf{j}_k}{\partial \mathbf{m}_k}. \tag{8.35}$$

  Once the derivatives have been computed, new estimates of $\mathbf{m}_k$ can be determined using a gradient-based minimization routine and a convergence criterion of choice. E.g., using a simple steepest-descent algorithm, a new estimate for $\mathbf{m}_k$ could be obtained as

$$\mathbf{m}_k^i = \mathbf{m}_{k-1}^{i-1} + \gamma \left(\frac{\partial \bar{\mathcal{J}}}{\partial \mathbf{m}_{k-1}^{i-1}} \right)^T, \tag{8.36}$$

  where the superscript i is the iteration counter and where γ is a small number; see e.g. Gill, Murray and Wright (1986).

Regularization

Often the number of parameters in realistic problems is very large, and the minimization problem may have a large number of local minima. Regularization may be required to reduce the dimension of the parameter space, i.e. to obtain a smoother objective function. Following Rommelse *et al*. (2010), consider a parameter vector $\mathbf{m} \in \mathbb{R}^p$, and define a transformation

$$\mathbf{m} = \mathbf{\Phi}\boldsymbol{\beta} \quad , \tag{8.37}$$

where $\boldsymbol{\beta} \in \mathbb{R}^q$ is a reduced-order parameter vector with $q \ll p$, and $\mathbf{\Phi} \in \mathbb{R}^{p \times q}$ is an orthonormal matrix, $\mathbf{\Phi}^T\mathbf{\Phi} = \mathbf{I} \in \mathbb{R}^{q \times q}$, such that

$$\boldsymbol{\beta} = \mathbf{\Phi}^T\mathbf{m} \quad , \tag{8.38}$$

which allows us to rewrite equation (8.36) as

$$\mathbf{\Phi}\boldsymbol{\beta}^i = \mathbf{\Phi}\boldsymbol{\beta}^{i-1} + \gamma\left(\frac{\partial\bar{\mathcal{J}}}{\partial\mathbf{m}^{i-1}}\right)^T \quad , \tag{8.39}$$

or

$$\boldsymbol{\beta}^i = \boldsymbol{\beta}^{i-1} + \gamma\mathbf{\Phi}^T\left(\frac{\partial\bar{\mathcal{J}}}{\partial\mathbf{m}^{i-1}}\right)^T \quad . \tag{8.40}$$

The iterative procedure now involves changing the values of the reduced-order parameter vector $\boldsymbol{\beta}$. Alternatively, we may back-substitute equation (8.38) in equation (8.40), resulting in

$$\mathbf{\Phi}^T\mathbf{m}^i = \mathbf{\Phi}^T\mathbf{m}^{i-1} + \gamma\mathbf{\Phi}^T\left(\frac{\partial\bar{\mathcal{J}}}{\partial\mathbf{m}^{i-1}}\right)^T \quad , \tag{8.41}$$

or

$$\mathbf{m}^i = \mathbf{m}^{i-1} + \gamma\,\mathbf{\Phi}\mathbf{\Phi}^T\left(\frac{\partial\bar{\mathcal{J}}}{\partial\mathbf{m}^{i-1}}\right)^T \quad , \tag{8.42}$$

in which case the iterations are still performed in terms of the high-order parameter **m**, but with the aid of a regularized gradient. More in general, the regularized gradient $\mathbf{\Phi}\mathbf{\Phi}^T(\partial\bar{\mathcal{J}}/\partial\mathbf{m})^T$ can be used in any gradient-based minimization algorithm. The matrix $\mathbf{\Phi}\mathbf{\Phi}^T \in \mathbb{R}^{p \times p}$ projects the *p*-dimensional gradient on a lower-order, *q*-dimensional subspace. The exact form of the projection will be have to be determined for the individual elements of the generalized parameter vector **m** as listed in definition (8.16). E.g. consider the case where the parameter vector $\boldsymbol{\theta}$ is equal to the vector of grid block permeabilities **k**. If we then apply equation (8.37) to just $\boldsymbol{\theta}$, instead of to the entire vector **m**, the columns of $\mathbf{\Phi}$ can be interpreted as spatial patterns of permeability values, i.e. as basis functions in permeability space. The matrix product $\mathbf{\Phi}\mathbf{\Phi}^T$ acts as a 'low-pass filter' that only transmits the spatial fluctuations with low spatial frequencies and blocks those that display rapid fluctuations. A convenient choice for the columns of $\mathbf{\Phi}$ would then e.g. be the eigenvectors corresponding to the *q* lowest eigenvalues of the covariance matrix $\mathbf{P}_\theta$, or, equivalently, the first *q* left-singular vectors of the square root $\mathbf{L}_\theta$ of $\mathbf{P}_\theta = \mathbf{L}_\theta\mathbf{L}_\theta^T$.

8.3.3 The representer method

The representer method

If we consider the special case of linear system and output equations, and a generalized parameter vector $\mathbf{m}$ that only contains $\mathbf{x}_0$ and $\mathbf{u}$, the estimation problem becomes a linear strong constraint one, for which we can obtain the solution through a single iteration of one forward and one backward integration. If $\mathbf{m}$ also contains model errors $\boldsymbol{\varepsilon}$, the problem becomes a linear weak constraint one, which has to be solved iteratively because the errors appear in the 'forward' equation (8.32), whereas we can only compute their values from equation (8.34) once we have obtained the values for the Lagrange multipliers $\boldsymbol{\lambda}$ from the 'backward' equation (8.29). An alternative approach was proposed by Bennett (1992), who developed a way to decouple and at the same time regularize the linear weak constraint equations, which became known as the *representer method*. We stress that although the representer method is usually introduced as an efficient method to solve the weak constraint problem, it may equally well be applied to the strong constraint problem. Subsequent work by Bennett (2002) also addressed nonlinear applications for state estimation. Baird and Dawson (2005) applied the method to linear state estimation in single-phase reservoir flow, Valstar *et al*. (2004) extended it to nonlinear parameter estimation in ground water flow, and Przybysz-Jarnut *et al*. (2007) and Baird and Dawson (2007) extended the method to two-phase reservoir flow. Here we follow an approach inspired by the implementation of Rommelse *et al*. (2010) with some further modifications. The crux of the representer method is to choose a regularization in which the deviations of all variables and parameters in equations (8.28) to (8.33) from their prior values are expressed as the product of matrices $\mathbf{R}$, with columns $\mathbf{r}$ known as representers, and coefficients $\mathbf{b} \in \mathbb{R}^r$ defined as

$$\mathbf{b}^T \triangleq \begin{bmatrix} \mathbf{b}_1^T & \mathbf{b}_2^T & \dots & \mathbf{b}_K^T \end{bmatrix}, \tag{8.43}$$

where

$$\mathbf{b}_k^T \triangleq \left(\mathbf{d}_k - \mathbf{y}_k\right)^T \mathbf{P}_{\eta_k}^{-1} \equiv \boldsymbol{\eta}_k^T \mathbf{P}_{\eta_k}^{-1}, \quad k = 1, 2, \dots, K . \tag{8.44}$$

In words, $\mathbf{b}$ is defined as a super vector of misfits between *all* predicted measurements $\mathbf{y}_{1:K}$ and *all* actual measurements $\mathbf{d}_{1:K}$, weighted by uncertainty in the measurements as expressed by the inverse of the measurement error covariance matrices $\mathbf{P}_{\eta_{1:K}}$. The r columns $\mathbf{r}$ of the representer matrices may be interpreted as basis functions, where r is equal to all measurements, i.e. to the number of elements m of the output vector times the total number of time steps K. This total number of measurements $r = mK$ is typically much smaller than the number of elements n of the state vector. Alternatively, we may express equations (8.43) and (8.44) more compactly as

$$\mathbf{b}^T \triangleq \left(\mathbf{d} - \mathbf{y}\right)^T \mathbf{P}_{\eta}^{-1} \tag{8.45}$$

where

$$\mathbf{d}^T \triangleq \begin{bmatrix} \mathbf{d}_1^T & \mathbf{d}_2^T & \dots & \mathbf{d}_K^T \end{bmatrix}, \tag{8.46}$$

$$\mathbf{y}^T \triangleq \begin{bmatrix} \mathbf{y}_1^T & \mathbf{y}_2^T & \dots & \mathbf{y}_K^T \end{bmatrix}, \tag{8.47}$$

and

$$\mathbf{P}_\eta \triangleq \begin{bmatrix} \mathbf{P}_{\eta_1} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{P}_{\eta_2} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{P}_{\eta_K} \end{bmatrix}. \tag{8.48}$$

In practice we may have output vectors $\mathbf{y}_k$ that have different numbers of elements at each time step, including vectors with zero elements. We can then reduce the size of $\mathbf{d}$ and $\mathbf{y}$ by maintaining only the non-zero elements, and reduce the size of $\mathbf{P}_\eta$ accordingly.

Representers

A special feature of the representer method is that the numerical values of neither $\mathbf{R}$ nor $\mathbf{b}$ are fixed a priori and that their computation forms part of the minimization process. We can now define the various representer matrices according to:

$$\mathbf{R}_k^m \mathbf{b} \triangleq \mathbf{m}_k - \breve{\mathbf{m}}_k \quad \text{(the parameter representers)}, \tag{8.49}$$

$$\mathbf{R}_k^x \mathbf{b} \triangleq \mathbf{x}_k - \breve{\mathbf{x}}_k \quad \text{(the state representers)}, \tag{8.50}$$

$$\mathbf{R}_k^y \mathbf{b} \triangleq \mathbf{y}_k - \breve{\mathbf{y}}_k \quad \text{(the output representers)}, \tag{8.51}$$

$$\mathbf{R}_k^\lambda \mathbf{b} \triangleq \boldsymbol{\lambda}_k \quad \text{(the adjoint state representers)}, \tag{8.52}$$

$$\mathbf{R}_k^\mu \mathbf{b} \triangleq \boldsymbol{\mu}_k \quad \text{(the adjoint output representers)}, \tag{8.53}$$

where we used superscripts to indicate the different representer matrices. Note that the parameter, state and output representers have been defined in terms of the differences between the respective variables and their prior values, whereas the adjoint representers have been defined directly in terms of the adjoint variables. The reason is that the former appear in the system and output equations, which are nonlinear, while the latter appear in the adjoint equations, which are linear.

Representer equations

Substitution of equations (8.49) to (8.53) in equations (8.28) to (8.33) and dividing out the terms $\mathbf{b}^T$ results in

$$\left(\mathbf{R}_k^m\right)^T \mathbf{P}_{m_k}^{-1} + \left(\mathbf{R}_k^\lambda\right)^T \frac{\partial \mathbf{g}_k}{\partial \mathbf{m}_k} + \left(\mathbf{R}_k^\mu\right)^T \frac{\partial \mathbf{j}_k}{\partial \mathbf{m}_k} = \mathbf{0}^T, \quad k = 1, 2, \ldots, K \tag{8.54}$$

$$\left(\mathbf{R}_{k+1}^\lambda\right)^T \frac{\partial \mathbf{g}_{k+1}}{\partial \mathbf{x}_k} + \left(\mathbf{R}_k^\lambda\right)^T \frac{\partial \mathbf{g}_k}{\partial \mathbf{x}_k} + \left(\mathbf{R}_k^\mu\right)^T \frac{\partial \mathbf{j}_k}{\partial \mathbf{x}_k} = \mathbf{0}^T, \tag{8.55}$$

$$\left(\mathbf{R}_K^\lambda\right)^T \frac{\partial \mathbf{g}_K}{\partial \mathbf{x}_K} + \left(\mathbf{R}_K^\mu\right)^T \frac{\partial \mathbf{j}_K}{\partial \mathbf{x}_K} = \mathbf{0}^T, \tag{8.56}$$

$$\left(\mathbf{R}_k^y\right)^T \mathbf{P}_{\eta_k}^{-1} + \left(\mathbf{R}_k^\mu\right)^T \frac{\partial \mathbf{j}_k}{\partial \mathbf{y}_k} = \mathbf{0}^T, \tag{8.57}$$

$$\mathbf{g}_k^T \left(\breve{\mathbf{m}}_k + \mathbf{R}_k^m \mathbf{b}, \breve{\mathbf{x}}_{k-1} + \mathbf{R}_{k-1}^x \mathbf{b} \breve{\mathbf{x}}_k + \mathbf{R}_k^x \mathbf{b}\right) = \mathbf{0}^T, \tag{8.58}$$

$$\mathbf{j}_k^T \left(\breve{\mathbf{m}}_k + \mathbf{R}_k^m \mathbf{b}, \breve{\mathbf{x}}_k + \mathbf{R}_k^x \mathbf{b}, \breve{\mathbf{y}}_k + \mathbf{R}_k^y \mathbf{b}\right) = \mathbf{0}^T, \tag{8.59}$$

Linearized system and output equations

Approximating expressions (8.58) and (8.59) by using a *backward* first-order Taylor expansion with respect to the small terms $\mathbf{R}_{k-1}^{x}\mathbf{b}$, $\mathbf{R}_{k}^{x}\mathbf{b}$, etc. around the (unknown) stationary point results in

$$\begin{aligned}
&\mathbf{g}_k^T\left(\breve{\mathbf{m}}_k, \breve{\mathbf{x}}_{k-1}, \breve{\mathbf{x}}_k\right) \\
&= \mathbf{g}_k^T\left(\mathbf{m}_k - \mathbf{R}_k^m\mathbf{b}, \mathbf{x}_{k-1} - \mathbf{R}_{k-1}^x\mathbf{b}, \mathbf{x}_k - \mathbf{R}_k^x\mathbf{b}\right) \\
&\approx \mathbf{g}_k^T\left(\mathbf{m}_k, \mathbf{x}_{k-1}, \mathbf{x}_k\right) - \frac{\partial \mathbf{g}_k^T}{\partial \mathbf{m}_k}\mathbf{R}^m\mathbf{b} - \frac{\partial \mathbf{g}_k^T}{\partial \mathbf{x}_{k-1}}\mathbf{R}_{k-1}^x\mathbf{b} - \frac{\partial \mathbf{g}_k^T}{\partial \mathbf{x}_k}\mathbf{R}_k^x\mathbf{b},
\end{aligned} \tag{8.60}$$

$$\begin{aligned}
&\mathbf{j}_k^T\left(\breve{\mathbf{m}}_k, \breve{\mathbf{x}}_k, \breve{\mathbf{y}}_k\right) \\
&= \mathbf{j}_k^T\left(\mathbf{m}_k - \mathbf{R}_k^m\mathbf{b}, \mathbf{x}_k - \mathbf{R}_k^x\mathbf{b}, \mathbf{y}_k - \mathbf{R}_k^y\mathbf{b}\right) \\
&\approx \mathbf{j}_k^T\left(\mathbf{m}_k, \mathbf{x}_k, \mathbf{y}_k\right) - \frac{\partial \mathbf{j}_k^T}{\partial \mathbf{m}_k}\mathbf{R}^m\mathbf{b} - \frac{\mathbf{j}_k^T}{\partial \mathbf{x}_k}\mathbf{R}_k^x\mathbf{b} - \frac{\mathbf{j}_k^T}{\partial \mathbf{y}_k}\mathbf{R}_k^y\mathbf{b},
\end{aligned} \tag{8.61}$$

from which we obtain‡

$$\frac{\partial \mathbf{g}_k}{\partial \mathbf{m}_k}\mathbf{R}_k^m + \frac{\partial \mathbf{g}_k}{\partial \mathbf{x}_{k-1}}\mathbf{R}_{k-1}^x + \frac{\partial \mathbf{g}_k}{\partial \mathbf{x}_k}\mathbf{R}_k^x = \mathbf{0}, \tag{8.62}$$

$$\frac{\partial \mathbf{j}_k}{\partial \mathbf{m}_k}\mathbf{R}_k^m + \frac{\partial \mathbf{j}_k}{\partial \mathbf{x}_k}\mathbf{R}_k^x + \frac{\partial \mathbf{j}_k}{\partial \mathbf{y}_k}\mathbf{R}_k^y = \mathbf{0}, \tag{8.63}$$

where $k = 1, 2, \ldots, K$, and where the derivatives should be evaluated at the stationary point.‡

Coefficient equation

Finally, we rewrite equation (8.45) in terms of the output representers:

$$\mathbf{b}^T = \left(\mathbf{d} - \breve{\mathbf{y}} - \mathbf{R}^y\mathbf{b}\right)^T \mathbf{P}_\eta^{-1}, \tag{8.64}$$

where

$$\breve{\mathbf{y}}^T \triangleq \begin{bmatrix} \breve{\mathbf{y}}_1^T & \breve{\mathbf{y}}_2^T & \cdots & \breve{\mathbf{y}}_K^T \end{bmatrix} \tag{8.65}$$

is the combined prior output, and

$$\mathbf{R}^y \triangleq \begin{bmatrix} \mathbf{R}_1^y & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{R}_2^y & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{R}_K^y \end{bmatrix} \tag{8.66}$$

the combined output representer for all times steps. We can now obtain **b** from equation (8.64) by solving the system of *r* linear equations

‡ The terms $\mathbf{g}_k^T\left(\mathbf{m}_k, \mathbf{x}_{k-1}, \mathbf{x}_k\right)$ and $\mathbf{j}_k^T\left(\mathbf{m}_k, \mathbf{x}_k, \mathbf{y}_k\right)$ are equal to **0** because they exactly obey the system and output equations.

‡ If we would have used a *forward* Taylor expansion around the priors we would have obtained exactly the same expressions, but with derivatives evaluated at the priors rather than at the stationary point.

$$\left(\mathbf{P}_\eta + \mathbf{R}^y\right)\mathbf{b} = \mathbf{d} - \breve{\mathbf{y}}\,, \tag{8.67}$$

where, as per design of the representer method, $r = mK$ is the total amount of measurements.

Iterative procedure

To start the representer method we first perform an ordinary forward simulation to obtain the prior values of all variables and parameters. Thereafter, we iteratively compute the representers, using 'forward' and 'backward' simulations, until $\bar{\mathcal{J}}$ has converged to below a preset limit, which leads to the following scheme:

Algorithm 8.2

1) Choose prior values $\breve{\mathbf{m}}_{1:K}$ for the uncertain parameters.
2) Compute prior states $\breve{\mathbf{x}}_{1:K}$ and outputs $\breve{\mathbf{y}}_{1:K}$ from equations (8.32) and (8.33).
3) Compute the adjoint representers $\mathbf{R}^{\mu}_{1:K}$ and $\mathbf{R}^{\lambda}_{1:K}$ from equations (8.57), (8.56) and (8.55).
4) Compute the representers $\mathbf{R}^{m}_{1:K}$ for the uncertain parameters from equation (8.54).
5) Compute the state representers $\mathbf{R}^{x}_{1:K}$ and the output representers $\mathbf{R}^{y}_{1:K}$ from equations (8.62) and (8.63).
6) Compute the representer coefficients **b** from equation (8.67).
7) Compute new values for $\mathbf{m}_{1:K}$, $\mathbf{x}_{1:K}$ and $\mathbf{y}_{1:K}$ from equations (8.49) to (8.51).
8) Compute $\bar{\mathcal{J}}$ and check for convergence. If converged stop, else return to step 4.

Computational issues

- The representer matrices $\mathbf{R}_k$ consist of r columns $\mathbf{r}_{ik}$, $i = 1,\ldots,r$, and solving a system of equations in terms of $\mathbf{R}_k$ is therefore equivalent to solving the system r times with r different right-hand sides. For each iteration, we therefore have to perform r forward computations of the state representers and r backward computations of the adjoint representers[†].
- During the iteration process, the Jacobians $\partial\mathbf{g}_k/\partial\mathbf{x}_{k-1}$, $\partial\mathbf{g}_k/\partial\mathbf{x}_k$, etc. need to be recomputed using the most recent iterates of $\mathbf{m}_k$, $\mathbf{x}_{k\text{-}1}$, $\mathbf{x}_k$ and $\mathbf{y}_k$ such that in the limit of convergence the derivatives are computed exactly in the stationary point.
- Just as in the strong-constraint case we can replace the direct computation of new values for the uncertain variables $\mathbf{m}_{1:K}$ by an approach to compute the derivatives $\partial\bar{\mathcal{J}}/\partial\mathbf{m}_k$, by inserting the present values of the parameters in equations (8.54) leading to

$$\frac{\partial\bar{\mathcal{J}}}{\partial\mathbf{m}_k} = \mathbf{b}^T\left[\left(\mathbf{R}^m_k\right)^T\mathbf{P}^{-1}_{m_k} + \left(\mathbf{R}^{\lambda}_k\right)^T\frac{\partial\mathbf{g}_k}{\partial\mathbf{m}_k} + \left(\mathbf{R}^{\mu}_k\right)^T\frac{\partial\mathbf{j}_k}{\partial\mathbf{m}_k}\right] \tag{8.68}$$

  Once the derivatives have been computed, new estimates of the uncertain variables may be determined using a gradient-based minimization routine.

[†] For small systems that can be solved using direct linear solvers we benefit from the fact that solving the same system with r different right-hand sides is considerably faster than solving r different systems. However, the large number of state variables in realistic reservoir models implies that we have to use iterative linear solvers, in which case the benefits are less, although still significant because we only need to compute the preconditioner once.

Reduced-order representers

Usually, measurements will not be available at all times steps $k = 1, 2, \ldots, K$, and the representers **b** as defined in equation (8.45) may therefore contain many zeros. Even when **b** is not sparse, some of the measurements may be linearly dependent or the total number of elements may be too large to be computationally efficient. Moreover, it may be required to further regularize the parameter space to reduce the chance of ending up in a local minimum. Such a further regularization can be obtained by a more general definition of the representer coefficients as proposed by Rommelse et al. (2010):

$$\mathbf{b}^T \triangleq \left(\mathbf{d} - \breve{\mathbf{y}}\right)^T \mathbf{P}_\eta^{-1} \boldsymbol{\Phi}, \tag{8.69}$$

where $\boldsymbol{\Phi} \in \mathbb{R}^{mK \times r}$ is an orthonormal reduction matrix with $r < mK$. We can now rewrite equation (8.67) as:

$$\underbrace{\left(\mathbf{P}_\eta \boldsymbol{\Phi} + \mathbf{R}^y\right)}_{mK \times r} \mathbf{b} = \mathbf{d} - \breve{\mathbf{y}}. \tag{8.70}$$

Because the matrix at the left hand side of equation (8.70) is rectangular, which implies that we have more equations than unknowns, we can solve it formally in a least square sense according to

$$\mathbf{b} = \left(\mathbf{Q}^T \mathbf{Q}\right)^{-1} \mathbf{Q}^T \left(\mathbf{d} - \breve{\mathbf{y}}\right), \tag{8.71}$$

where

$$\mathbf{Q} \triangleq \mathbf{P}_\eta \boldsymbol{\Phi} + \mathbf{R}^y, \tag{8.72}$$

or, computationally more efficient by solving the system of equations

$$\left(\mathbf{Q}^T \mathbf{Q}\right) \mathbf{b} = \mathbf{Q}^T \left(\mathbf{d} - \breve{\mathbf{y}}\right). \tag{8.73}$$

A convenient choice for the columns of $\boldsymbol{\Phi}$ are the eigenvectors corresponding to the *r* lowest eigenvalues of the $mK \times mK$ measurement sensitivity matrix $\partial \mathbf{j} / \partial \mathbf{x}$, where

$$\mathbf{j}^T \triangleq \begin{bmatrix} \mathbf{j}_1^T & \mathbf{j}_2^T & \cdots & \mathbf{j}_K^T \end{bmatrix}, \tag{8.74}$$

or, equivalently, the first *r* left-singular vectors of the square root $\mathbf{L}$ of $\partial \mathbf{j} / \partial \mathbf{x} = \mathbf{L}\mathbf{L}^T$.

8.4 References for Chapter 8

Aster, R.C., Borchers, B. and Thurber, C.H., 2005: *Parameter estimation and inverse problems*, Elsevier Academic Press, London.

Baird, J. and Dawson, C., 2005: The representer method for data assimilation in single-phase Darcy flow in porous media. *Computational Geosciences* **9** (4) 247-271. DOI: 10.1007/s10596-005-9006-2.

Baird, J., Dawson, C., 2007: The representer method for two-phase flow in porous media. *Computational Geosciences* **11** (3) 235-248. DOI: 10.1007/s10596-007-9048-8.

Bennett, A.F., 1992: *Inverse methods in physical oceanography*, Cambridge University Press, Cambridge.

Bennett, A.F., 2002: *Inverse modeling of the ocean and the atmosphere*, Cambridge University Press, Cambridge.

Evensen, G., 2009: *Data assimilation – The ensemble Kalman filter*, 2nd ed., Springer, Berlin.

Lewis, J.M., Lakshmivarahan, S. and Dhall, S.K., 2006: *Dynamic data assimilation: a least-squares approach*, Cambridge University Press, Cambridge.

Ljung, L., 1999: *System identification – Theory for the user, 2nd ed.*, Prentice-Hall PTR, Upper Saddle River.

Oliver, D.S., Reynolds, A.C. and Liu, N., 2008: *Inverse theory for petroleum reservoir characterization and history matching*, Cambridge University Press, Cambridge.

Oliver, D.S. and Chen, Y., 2011: Recent progress on reservoir history matching: a review. *Computational Geosciences*, **15** (1) 185-221. DOI: 10.1007/s10596-010-9194-2.

Przybysz-Jarnut, J.K., Hanea, R.G., Jansen, J.D. and Heemink, A.W., 2007: Application of the representer method for parameter estimation in numerical reservoir models. *Computational Geosciences* **11** (1) 73-85. DOI: 10.1007/s10596-006-9035-5.

Rommelse, J.R., Jansen, J.D. and Heemink, A.W., 2010: An efficient weak-constraint gradient-based parameter estimation algorithm using representer expansions. *SPE Journal* **15** (1) 18-30. DOI: 10.2118/120120-PA.

Tarantola, A., 2005: *Inverse problem theory and methods for model parameter estimation*, SIAM, Philadelphia.

Valstar, J.R., McLaughlin, D.B., Te Stroet, C.B.M., and Van Geer, F.C., 2004: A representer-based inverse method for groundwater flow and transport applications. *Water Resources Research* **40** W05116. DOI: 10.1029/2003WR002922.

9 Closed-loop reservoir management

To be written.

Appendix A – Elements of linear algebra

This appendix briefly covers some elements of linear algebra of which we make use in the body of the text. It is meant to serve as a refresher. Here we focus on the geometric aspects and mention some of the algebraic aspect in passing. Textbooks that treat these topics at an introductory level and that emphasize the geometric point of view are, e.g., Strang (2003, 2006), Poole (2003) or Lay (2003). For an in-depth treatment of the computational aspects of linear algebra, see Golub and Van Loan (1996).

A.1 Vectors and matrices

A.1.1 A warning

In this text we use the word *vector* in different meanings. Sometimes rather loosely, as a convenient abbreviation for a one-dimensional array of numbers which may or may not have the same physical dimension. E.g. we use the concept of a state vector as the set of pressure and saturation values in a reservoir simulation model. In that case the pressures are elements of the set of positive real numbers, while the saturations are elements of the set of real numbers between zero and one. Usually, however, we refer to vectors more strictly. In particular the concepts of linear algebra, as reviewed in this appendix, are defined for vectors and matrices with elements that belong to the entire set of real numbers $\mathbb{R}$. Blind application of the machinery of linear algebra to the equations for porous media flow may therefore lead to results that violate physical constraints.

A.1.2 Notation

We indicate matrices with boldface capitals and column vectors with boldface lower case letters. The elements of vectors and matrices may be real or complex numbers, or even functions, but in this text we will only consider real elements, i.e. elements belonging to the set $\mathbb{R}$. The number of elements of a vector is called its *dimension*, and the notation $\mathbf{x} \in \mathbb{R}^n$ means that $\mathbf{x}$ belongs to the set of all n-dimensional column vectors. Row vectors are denoted as transposed column vectors, where we use the superscript T to indicate the transpose. For example:

$$\mathbf{A} = \begin{bmatrix} A_{11} & A_{12} & \cdots & A_{1n} \\ A_{21} & A_{22} & \cdots & A_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ A_{m1} & A_{m2} & \cdots & A_{mn} \end{bmatrix}, \quad \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, \quad \mathbf{y}^T = \begin{bmatrix} y_1 & y_2 & \cdots & y_m \end{bmatrix}. \quad \text{(A.1, A.2, A.3)}$$

Here, $\mathbf{A}$ is referred to an $m \times n$ matrix, i.e. a matrix with m rows and n columns. This may also be indicated as $\mathbf{A} \in \mathbb{R}^{m \times n}$, which, in analogy to the definition used for vectors, formally means that $\mathbf{A}$ belongs to the set of all $m \times n$ matrices. Rows of a matrix will be indicated with superscripts and columns with subscripts:

$$\mathbf{A} = \begin{bmatrix} \mathbf{a}^1 \\ \mathbf{a}^2 \\ \vdots \\ \mathbf{a}^m \end{bmatrix} = \begin{bmatrix} \mathbf{a}_1 & \mathbf{a}_2 & \cdots & \mathbf{a}_n \end{bmatrix}. \quad \text{(A.4)}$$

In addition, we use colon notation to compactly refer to a set of vectors, e.g.,

$$\mathbf{a}_{1:p} \triangleq \{\mathbf{a}_i, i = 1,2,\ldots,p\} \text{ or } \mathbf{a}^{1:q} \triangleq \{\mathbf{a}^i, i = 1,2,\ldots,q\} \ , \tag{A.5}$$

such that we can express (A.4) also as

$$\mathbf{A} = \mathbf{a}^{1:m} = \mathbf{a}_{1:n} \ . \tag{A.6}$$

Zero vectors or *zero matrices* are indicated as $\mathbf{0}_{n\times m}$ where the subscripts indicate the number of rows and columns respectively, e.g.

$$\mathbf{0}_{2\times 3} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \tag{A.7}$$

or simply as $\mathbf{0}$ if the dimensions are clear from the context. Similarly a *unit matrix* is indicated as

$$\mathbf{I}_{3\times 3} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{A.8}$$

or just as $\mathbf{I}$, as appropriate[†]. The product of two vectors may be indicated in one of the following forms:

$$\mathbf{x}^T\mathbf{z} = \sum_{i=1}^{n} x_i z_i = \mathbf{x}\cdot\mathbf{z} = <\mathbf{x},\mathbf{z}> . \tag{A.9}$$

where, in this case, $\mathbf{x}$ and $\mathbf{z}$ are both $n \times 1$ vectors. The last representation, $<\mathbf{x},\mathbf{z}>$, is often referred to as an *inner produc*[*]*t*.

A.1.3 Matrix-vector multiplication

Equation (A.9) also forms the definition of vector multiplication; matrix-vector multiplication is defined as:

$$\mathbf{A}\mathbf{x} \triangleq \sum_{j=1}^{n} a_{ij} x_j \ . \tag{A.10}$$

We note that equation (A.10) can be interpreted as a multiplication of the individual rows of matrix $\mathbf{A}$ with vector $\mathbf{x}$:

$$\mathbf{A}\mathbf{x} = \begin{bmatrix} \mathbf{a}^1\mathbf{x} \\ \mathbf{a}^2\mathbf{x} \\ \vdots \\ \mathbf{a}^m\mathbf{x} \end{bmatrix}, \tag{A.11}$$

or, alternatively as a multiplication of the individual matrix columns with the individual vector elements:

$$\mathbf{A}\mathbf{x} = \begin{bmatrix} \mathbf{a}_1 x_1 & \mathbf{a}_2 x_2 & \cdots & \mathbf{a}_n x_n \end{bmatrix} . \tag{A.12}$$

[†] A unit matrix always has the same number of rows and columns, although occasionally we will use non-square matrices with elements that consist of only zeros and ones, which will be referred to as *generalized unit matrices*.

[*] More in general an inner product is defined with respect to a matrix, e.g. $<\mathbf{x},\mathbf{z}>_{\mathbf{A}} = \mathbf{x}^T\mathbf{A}\mathbf{z}$.

In the latter case we can interpret the matrix-vector equation

$$\mathbf{A}\mathbf{x} = \mathbf{b}, \tag{A.13}$$

with unknown **x** and known **A** and **b**, as the search for the vector multipliers $x_1, x_2, \ldots, x_n$ that just express **b** as a linear combination of the columns of **A**.

A.2 Geometric aspects

A.2.1 Vector spaces

Vectors can be interpreted geometrically as elements in space, and may be visualized as lines starting at the origin and pointing towards particular points. This leads to the mathematical interpretation of a vector as an element of a *linear vector space*, defined as a nonempty set $\mathcal{V}$ of vectors in which two operations are defined, addition and multiplication with a scalar, which obey the following axioms. For vectors **x**, **y** and **z** that are elements of $\mathcal{V}$ and real scalars *a* and *b*:

- $\mathbf{x}+\mathbf{y} \in \mathcal{V}$, i.e. $\mathcal{V}$ is closed under addition
- $\mathbf{x}+\mathbf{y} = \mathbf{y}+\mathbf{x}$, i.e. addition is *commutative*
- $(\mathbf{x}+\mathbf{y})+\mathbf{z} = \mathbf{x}+(\mathbf{y}+\mathbf{z})$, i.e. addition is *associative*
- There exists a unique *zero vector* $\mathbf{0}$ in $\mathcal{V}$ such that $\mathbf{x}+\mathbf{0} = \mathbf{x}$
- For each vector **x** there exists a unique vector $-\mathbf{x}$ in $\mathcal{V}$ such that $\mathbf{x}+(-\mathbf{x}) = \mathbf{0}$
- $a\mathbf{x} \in \mathcal{V}$, i.e. $\mathcal{V}$ is closed under scalar multiplication
- $a(\mathbf{x}+\mathbf{y}) = a\mathbf{x}+a\mathbf{y}$, and
- $(a+b)\mathbf{x} = a\mathbf{x}+b\mathbf{x}$, i.e. scalar multiplication is *distributive*
- $a(b\mathbf{x}) = (ab)\mathbf{x}$
- $1\mathbf{x} = \mathbf{x}$

A typical example of a linear vector space is $\mathbb{R}^n$, called a finite-dimensional vector space of dimension *n*. We do not consider infinite-dimensional vector spaces, or vector spaces with elements other than real numbers.

A.2.2 Subspaces

A subspace of a vector space $\mathcal{V}$ is a subset $\mathcal{W}$ of $\mathcal{V}$ that has the following properties:

- $\mathcal{W}$ is closed under addition
- $\mathcal{W}$ is closed under multiplication with scalars

A consequence of the second property is that the zero vector of $\mathcal{V}$ is in $\mathcal{W}$, as follows from choosing the scalar equal to 0. A classic example of the geometric interpretation of vector spaces is the representation of a two-dimensional subspace in $\mathbb{R}^3$ as a plane through the origin[†].

A.2.3 Norms

The *length* of an $n \times 1$ vector, also known as the *two-norm*, or the *Euclidian norm*, is defined as

[†] Note that a plane that does not pass through the origin is not a subspace because it does not contain the zero vector.

$$\|\mathbf{x}\| = \|\mathbf{x}\|_2 \triangleq \sqrt{\sum_{i=1}^{n} x_i^2} = \sqrt{\mathbf{x}^T\mathbf{x}} \ . \tag{A.14}$$

More in general, the *p-norm* of a vector is defined as

$$\|\mathbf{x}\|_p \triangleq \left(\sum_{i=1}^{n} x_i^p \right)^{\frac{1}{p}} \ . \tag{A.15}$$

A special case is obtained when p approaches infinity, which leads to the *infinity norm*

$$\|\mathbf{x}\|_\infty \triangleq \max_i (x_i), \quad i = 1,\ldots,n \ . \tag{A.16}$$

Yet another vector norm is sometimes used, defined with respect to a matrix:

$$\|\mathbf{x}\|_\mathbf{A} \triangleq \mathbf{x}^T\mathbf{A}\mathbf{x} \ . \tag{A.17}$$

A vector that has a length (i.e. a two-norm) equal to one is known as a *unit vector*[†]. Specific unit vectors with just one element equal to one and all others equal to zero are called *canonical unit vectors* and we will indicate them with the letter **i**. E.g. the third canonical unit vector in $\mathbb{R}^4$ is defined as

$$\mathbf{i}_3 \triangleq \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} \ . \tag{A.18}$$

A.2.4 Orthogonality

Two vectors are *orthogonal* if their vector product equals zero:

$$\mathbf{x}^T\mathbf{y} = 0 \ . \tag{A.19}$$

Geometrically this implies that the two vectors are perpendicular to each other. If, in addition, the vectors have length one, they are called *orthonormal*. A square matrix with orthonormal columns is called an *orthogonal matrix*[‡]. Orthogonal matrices have the algebraic property that their inverse is equal to their transpose, i.e. if **U** is an $n \times n$ orthogonal matrix we have

$$\mathbf{U}\mathbf{U}^{-1} = \mathbf{U}\mathbf{U}^T = \mathbf{I}_{n\times n} \ . \tag{A.20}$$

A special form of orthogonality is defined with respect to a matrix and is known as *conjugate orthogonality*:

$$\mathbf{x}^T\mathbf{A}\mathbf{y} = 0 \ . \tag{A.21}$$

Note that two conjugate orthogonal vectors are, in general, not perpendicular to each other.

A.2.5 Linear dependence

Two vectors $\mathbf{a}_1$ and $\mathbf{a}_2$ are linearly dependent if there exists a nonzero constant x such that

[†] Note that a unit *matrix*, as e.g. the one defined in equation (A.8), contains only zeros and ones, whereas the elements of a unit *vector* may have any value as long the vector has length one.

[‡] This is an unfortunate naming convention, and *orthonormal matrix* would have been a better choice.

$$\mathbf{a}_1 = \mathbf{a}_2 x \,. \tag{A.22}$$

The geometrical interpretation is that the vectors are lying on the same line. Similarly, a set of vectors $\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_n$ is linearly dependent if there exist constants $x_1, x_2, \ldots, x_n$, at least one of which is not zero, such that

$$\mathbf{a}_1 x_1 + \mathbf{a}_2 x_2 + \cdots + \mathbf{a}_n x_n = \mathbf{0} \,. \tag{A.23}$$

The geometrical interpretation is that at least three of the vectors are lying in the same plane.

A.2.6 Span

The *span* of a set of vectors is defined as the set of all linear combinations of these vectors. That is $\text{span}\{\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_n\}$ is the set of all vectors that can be represented as

$$\mathbf{a}_1 x_1 + \mathbf{a}_2 x_2 + \ldots + \mathbf{a}_n x_n = \mathbf{A}\mathbf{x} \;. \tag{A.24}$$

Geometrically, the span of a single vector, $\text{span}\{\mathbf{a}\}$, can be interpreted as a line, and the span of two vectors, $\text{span}\{\mathbf{a}_1, \mathbf{a}_2\}$, as a line or a plane, depending on whether the vectors are linearly dependent or not. More in general $\text{span}\{\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_n\}$ can be interpreted as an m-dimensional vector space $\mathcal{W}$ with $m \le n$, where the equality sign holds if all vectors $\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_n$ are linearly independent. It is also said that 'the vector space $\mathcal{W}$ is spanned by the vectors $\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_n$'. Note that $\mathcal{W}$ may be a subspace of a larger vector space, in which case the number of elements of each of the vectors $\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_n$ is larger than the dimension of the subspace.

A.2.7 Basis and coordinates

If an n-dimensional subspace is spanned by exactly n independent vectors these are called *basis vectors* or *coordinate vectors.* The set of basis vectors is called the *basis* of the vector space, and the elements $a_1, a_2, \ldots, a_n$ of a basis vector are referred to as its *coordinates*. A basis can be interpreted as an efficient span in the sense that it is a span with the minimum number of vectors that are needed to span the corresponding subspace[†]. In the particular case that the basis vectors of an n-dimensional vector space are just n canonical unit vectors of length n each, we have

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} x_1 + \begin{bmatrix} 0 \\ 1 \\ \vdots \\ 0 \end{bmatrix} x_2 + \ldots + \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix} x_n = \mathbf{i}_1 x_1 + \mathbf{i}_2 x_2 + \ldots + \mathbf{i}_n x_n = \mathbf{I}\mathbf{x} \;. \tag{A.25}$$

It is easily verified that the unit vectors $\mathbf{i}_1, \mathbf{i}_2, \ldots, \mathbf{i}_n$ are mutually orthogonal such that the unit matrix $\mathbf{I}$ is orthogonal. Such a particular basis is known as an *orthonormal basis* or *standard basis*. Note that because the basis consists of n vectors of n elements each they span the vector space $\mathbb{R}^n$.

A.2.8 Fundamental subspaces

The subspace spanned by the column vectors of a matrix is know as the *column space* of that matrix. Similarly, the subspace spanned by its rows is the *row space*. We will use the

[†] In fact, the minimum number of vectors required to span a subspace may be considered as the definition of the dimension of that subspace.

abbreviations col(•) and row(•) to indicate these subspaces. Note that the column space of a matrix **A** is identical to the row space of $\mathbf{A}^T$. Now, consider equation (A.13) again:

$$\mathbf{Ax} = \mathbf{b}\text{, or, in full, } \begin{bmatrix} A_{11} & A_{12} & \cdots & A_{1n} \\ A_{21} & A_{22} & \cdots & A_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ A_{m1} & A_{m2} & \cdots & A_{mn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}. \tag{A.26}$$

For known **A** and **b** this equation can only be solved for **x** if **b** is in col(**A**), that is, if **b** can be expressed as a linear combination of the columns of **A**. Similarly, the equation

$$\mathbf{A}^T\mathbf{y} = \mathbf{b}, \tag{A.27}$$

can only be solved for **y** if **b** is in row(**A**). We will review the implications of these conditions in more detail in Section A.3 below. A different situation occurs if we attempt solve the equation

$$\mathbf{Ax} = \mathbf{0}. \tag{A.28}$$

for unknown **x**, given **A**. In this case there will be a trivial solution, namely $\mathbf{x} = \mathbf{0}$, but there may also be non-trivial solutions if it is possible to linearly combine the columns of **A** such that they add up to zero, in other words if the columns are linearly dependent. The corresponding values of **x** then form elements of the *null space* of **A**, which is defined as the set of all possible vectors **x** that fulfill equation (A.28). We will use the abbreviation null(•) to indicate this subspace. The corresponding set of vectors **y** that fulfill the equation

$$\mathbf{A}^T\mathbf{y} = \mathbf{0}. \tag{A.29}$$

is known as the *left null space*. These four subspaces, i.e. the column space, the row space, the null space and the left null space of a matrix, are often referred to as the *fundamental subspaces* of linear algebra.

A.2.9 Rank and nullity

The *column rank* of a matrix is the number of independent columns. In other words, it is the dimension of the column space, or the dimension of the range. Similarly, the *row rank* of a matrix is the number of independent rows. It can be proved that the column rank is always equal to the row rank, and we can therefore simply speak of the *rank* of a matrix. For the proof of this non-trivial equality we refer to one of the texts mentioned at the beginning of Section A. It implies that the number of independent rows is always equal to the number of independent columns, and an $m \times n$ matrix can therefore not be of higher rank than the smaller of the two integers *n* and *m*. If a matrix has no dependent columns, and therefore also no dependent rows, it is said that the matrix has *full rank*. Note that this implies that a full-rank matrix is square. Rectangular matrices can be full column rank or full row rank, but not both. The *nullity* of a matrix is the number of dependent columns, i.e. the total number of columns minus the range, or in other words, the dimension of the null space. The term *rank deficiency* is used to indicate the number of dependent columns, i.e. the nullity. A rank-deficient square matrix is called *singular*; a full-rank square matrix *regular*. In summary, a matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$ with rank *r* has a column space and a row space which are both of dimension *r*, a null space of dimension $n - r$, and a left null space of dimension $m - r$.

A.2.10 Orthogonal complements

The *orthogonal complement* $\mathcal{V}^{\perp}$ of a subspace $\mathcal{V}$ consists of the space of all vectors that are orthogonal to all vectors in $\mathcal{V}$. A classic example is the orthogonal complement of a two-dimensional subspace (i.e. of a plane through the origin) in $\mathbb{R}^3$ which is simply a line through the origin perpendicular to the plane. The orthogonal complement is also a subspace, which implies that the origin forms part of both subspaces. The four fundamental subspaces of a matrix $\mathbf{A}$ form two pairs of orthogonal complements. In particular, the orthogonal complement of col($\mathbf{A}$) is equal to null($\mathbf{A}^T$), i.e. to the left null space of $\mathbf{A}$. Similarly, the orthogonal complement of null($\mathbf{A}$) is equal to row($\mathbf{A}$). More generally, two subspaces are orthogonal to each other if all vectors in one subspace are orthogonal to all vectors in the other subspace. If two subspaces are each others orthogonal complement they are therefore also orthogonal to each other. However, the reverse is not always true, i.e. two subspaces of the same space may be orthogonal but be of too low dimension to span the entire space. E.g. two perpendicular lines through the origin in $\mathbb{R}^3$ are orthogonal subspaces but since they only span a plane, they do not form orthogonal complements.

A.2.11 Transformations

The multiplication of a vector $\mathbf{x} \in \mathbb{R}^{n\times 1}$ with a square matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$,

$$\mathbf{y} = \mathbf{A}\mathbf{x} \ , \tag{A.30}$$

where $\mathbf{A}$ has full rank, may be considered a *transformation* of $\mathbf{x}$ to another vector $\mathbf{y} \in \mathbb{R}^{n\times 1}$. Mathematically, matrix multiplication is just one form of a much wider class of *linear transformations* between arbitrary vector spaces[†]. Geometrically, a linear transformation can be interpreted as an operation that takes all points of a vector space into either itself or into another vector space. E.g. it is easily verified that the operation $\mathbf{y} = \mathbf{A}\mathbf{x}$ on a vector $\mathbf{x} \in \mathbb{R}^2$, where

$$\mathbf{A} = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \text{ or } \mathbf{A} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}, \tag{A.31, A.32}$$

takes all vectors to $\mathbb{R}^2$ again, and represents a reflection through the *x*-axis, or a 90° anti-clockwise rotation respectively; see Figure A.1.

[†] Matrix multiplication is indeed a linear operation because it obeys the conditions $\mathbf{A}(\mathbf{x}+\mathbf{y}) = \mathbf{A}\mathbf{x} + \mathbf{A}\mathbf{y}$ and $\mathbf{A}(a\mathbf{x}) = a\mathbf{A}\mathbf{x}$ for all vectors $\mathbf{x}$ and $\mathbf{y}$ and scalars a.

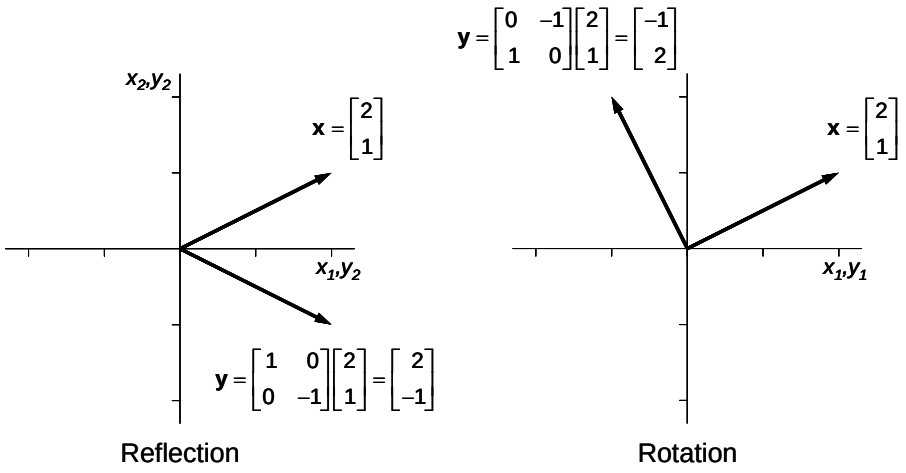


Figure A.1: Left: Reflection through the x-axis. Right: Anti-clockwise rotation over 90° around the origin.

A.2.12 Range and kernel

Above we discussed the column space and the null space of a matrix **A**. For linear transformations the analogous concepts are the *range* and the *kernel* of the transformation, which, in turn, are analogous to the range and the domain of a function *f*. Recall that the range of a function is defined as the set of all possible values $f(x)$, while the domain is the set of all possible values *x*. Therefore, the linear transformation **Ax**, with $\mathbf{x} \in \mathbb{R}^n$ and $\mathbf{A} \in \mathbb{R}^{m \times n}$, may be considered as a function taking vectors from the kernel, consisting of all vectors in $\mathbb{R}^n$, to the range, consisting of those vectors in $\mathbb{R}^m$ that can be obtained by applying the transformation **Ax**. Some textbooks use the notation range(**A**) and ker(**A**). Alternatively the range is referred to as the *image*, with the corresponding notation im(**A**).

A.2.13 Projections

If the matrix **A** is square but not of full rank, the vector multiplication (A.30) is still a linear transformation, but no longer maps vectors such that they fully remain within the same vector space. Instead, **A** now represents a *projection* on a lower-dimensional vector space which is a subspace of the kernel. E.g., the matrices

$$\mathbf{A} = \begin{bmatrix} 1 & -1 \\ 0 & 0 \end{bmatrix} \text{ and } \mathbf{A} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, \qquad \text{(A.33, A.34)}$$

represent projections on a one-dimensional subspace (a line), as illustrated in Figure A.2. A *projection matrix* is singular, which implies that the inverse operation of a projection on a subspace is not defined. A projection matrix is also *idempotent*, meaning that we can only project a vector once, and repeated application of a projection operator does not change the situation: $\mathbf{A}^2 = \mathbf{A}$. If **A** is symmetric, the corresponding projection is orthogonal, otherwise it is *oblique*. Somewhat confusingly an orthogonal-projection matrix produces an orthogonal projection but is not necessarily orthogonal itself.

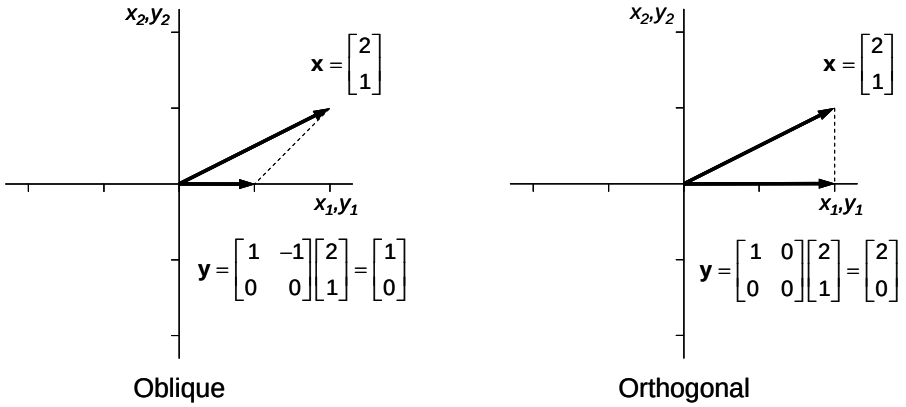


Figure A.2: Projections from $\mathbb{R}^2$ to a one-dimensional subspace in $\mathbb{R}^2$. Left: Oblique projection with a non-orthogonal-projection matrix. Right: Orthogonal projection with an orthogonal-projection matrix.

A.3 Linear equations

A.3.1 Geometry

The geometric interpretation of systems of linear equations $\mathbf{Ax} = \mathbf{b}$ is beautifully described in the textbooks of Strang and various associated papers; see e.g. Strang (1984, 1993, 2003, 2006), and you are encouraged to study the original texts to obtain a full appreciation of this topic.

A.3.2 Regular system matrix

For every square matrix **A** that is full rank we can define an inverse $\mathbf{A}^{-1}$ such that

$$\mathbf{A}\mathbf{A}^{-1} = \mathbf{A}^{-1}\mathbf{A} = \mathbf{I} \ . \tag{A.35}$$

For a rank-deficient square matrix, or for a rectangular matrix, the inverse does not exist. Now consider yet again the matrix-vector equation (A.13)

$$\mathbf{Ax} = \mathbf{b} \ , \tag{A.36}$$

where $\mathbf{A} \in \mathbb{R}^{m \times n}$, $\mathbf{b} \in \mathbb{R}^m$ and $\mathbf{x} \in \mathbb{R}^n$, and where **A** and **b** are known while **x** is to be determined. It was already discussed in the paragraph after equation (A.26) that this equation can be solved if **A** is regular, i.e. if **A** is not singular, i.e. if **A** is square and of full rank. In that case **b** is in col(**A**) **A**, i.e. **b** can be expressed as a linear combination of the columns of **A** with weight factors that form the elements of **x**. We can then formally write the solution of equation (A.36) as

$$\mathbf{x} = \mathbf{A}^{-1}\mathbf{b} \ . \tag{A.37}$$

In an actual computation of **x**, the inverse is seldom computed. Instead the elements of **x** are obtained through a computationally much more efficient process of adding, subtracting, scalar-multiplying, and reordering of the rows of **A**, known as *Gaussian elimination*. For details, see any of the references mentioned at the beginning of this appendix. For very large systems it is more efficient to use iterative solution procedures, but we will not discuss these computational aspects any further in this text.

A.3.3 Singular system matrix – overdetermined case

If **A** is rectangular, we have to distinguish two cases. 1) The overdetermined case where **A** has more rows than columns, i.e. where we have more equations than unknown elements

of **x**. 2) The underdetermined case where **A** has more columns than rows, i.e. where we have more unknowns than equations. In the first case, where $m > n$, the column space of the 'tall' matrix **A** is of a lower dimension than the space where **b** lives, such that in general it will not be possible express **b** exactly as linear combination **x** of the columns of **A**. However, we could try to find an approximate solution $\bar{\mathbf{x}}$ as a linear combination that projects **b** on the span of **A** in some optimal sense. To do so we define an error vector **e** as the difference between the projected vector $\bar{\mathbf{b}} = \mathbf{A}\bar{\mathbf{x}}$ and the original vector **b**:

$$\mathbf{e} \triangleq \mathbf{b} - \bar{\mathbf{b}} \ . \tag{A.38}$$

We can now define optimality as the requirement that **e** is as small as possible. This leads to a minimization problem that can be solved using a least squares approach. However, we can also obtain the solution using geometric arguments. Recall that the shortest distance between a point (or the end of a vector) and a line is given by a vector perpendicular to that line. Indeed the error vector **e** between a vector **b** and its projection $\bar{\mathbf{b}}$ on a line is shortest if the projection is orthogonal. Similarly, the shortest vector **e** in equation (A.38) will be the one that is perpendicular to the column space of **A**, i.e. perpendicular to each column of **A**:

$$\begin{bmatrix} \mathbf{a}_1^T \\ \mathbf{a}_2^T \\ \vdots \\ \mathbf{a}_n^T \end{bmatrix} \mathbf{e} = \begin{bmatrix} \mathbf{a}_1^T \\ \mathbf{a}_2^T \\ \vdots \\ \mathbf{a}_n^T \end{bmatrix} \left(\mathbf{b} - \mathbf{A}\bar{\mathbf{x}} \right) = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} , \tag{A.39}$$

which can also be written as

$$\mathbf{A}^T \left(\mathbf{b} - \mathbf{A}\bar{\mathbf{x}} \right) = \mathbf{A}^T\mathbf{b} - \mathbf{A}^T\mathbf{A}\bar{\mathbf{x}} = \mathbf{0} . \tag{A.40}$$

If **A** is full rank, $\mathbf{A}^T\mathbf{A}$ will be regular and we can solve from equation (A.40) for $\bar{\mathbf{x}}$ according to

$$\bar{\mathbf{x}} = \left(\mathbf{A}^T\mathbf{A} \right)^{-1} \mathbf{A}^T\mathbf{b} \ . \tag{A.41}$$

Equation (A.41) is often written as[‡]

$$\bar{\mathbf{x}} = \mathbf{A}^+\mathbf{b} \ . \tag{A.42}$$

where the *pseudo-inverse* $\mathbf{A}^+$ is defined as

$$\mathbf{A}^+ \triangleq \left(\mathbf{A}^T\mathbf{A} \right)^{-1} \mathbf{A}^T \ . \tag{A.43}$$

Note that the Gramian matrix $\mathbf{A}^T\mathbf{A}$ is square and symmetric (because $(\mathbf{A}^T\mathbf{A})^T = \mathbf{A}^T\mathbf{A}$) and has the 'small' dimension $n \times n$. The projected vector $\bar{\mathbf{b}} = \mathbf{A}\bar{\mathbf{x}}$ can be written as

$$\bar{\mathbf{b}} = \bar{\mathbf{P}}_b\mathbf{b} \ , \tag{A.44}$$

where

$$\bar{\mathbf{P}}_b \triangleq \mathbf{A}\left(\mathbf{A}^T\mathbf{A} \right)^{-1} \mathbf{A}^T \ , \tag{A.45}$$

[‡] Written as $(\mathbf{A}^T\mathbf{A})\mathbf{x} = \mathbf{A}^T\mathbf{b}$ equation (A.41) is often referred to as the system of *normal equations*.

is an orthogonal-projection matrix which has the 'large' dimension $m \times m$, and which is singular with rank deficiency $m-1$ (i.e. it has rank 1). Because the error **e** is perpendicular to $\text{col}(\mathbf{A})$ it is in the orthogonal complement of $\text{col}(\mathbf{A})$ which is $\text{null}(\mathbf{A}^T)$. We therefore define the complementary orthogonal-projection matrix

$$\mathbf{P}_b^{\perp} \triangleq \mathbf{I} - \overline{\mathbf{P}}_b = \mathbf{I} - \mathbf{A}\left(\mathbf{A}^T\mathbf{A}\right)^{-1}\mathbf{A}^T \ , \tag{A.46}$$

which is also of size $m \times m$, and which is singular with rank $n-m$. Every vector **b** can now be written as the sum of two projections, one on $\text{col}(\mathbf{A})$, and one on $\text{null}(\mathbf{A}^T)$:

$$\mathbf{b} = \overline{\mathbf{b}} + \mathbf{e} = \overline{\mathbf{b}} + \mathbf{b}^{\perp} = \overline{\mathbf{P}}_b\mathbf{b} + \mathbf{P}_b^{\perp}\mathbf{b} \ . \tag{A.47}$$

Figure A.3 gives a graphical interpretation of the various relationships that play a role in solving an overdetermined system of equations.

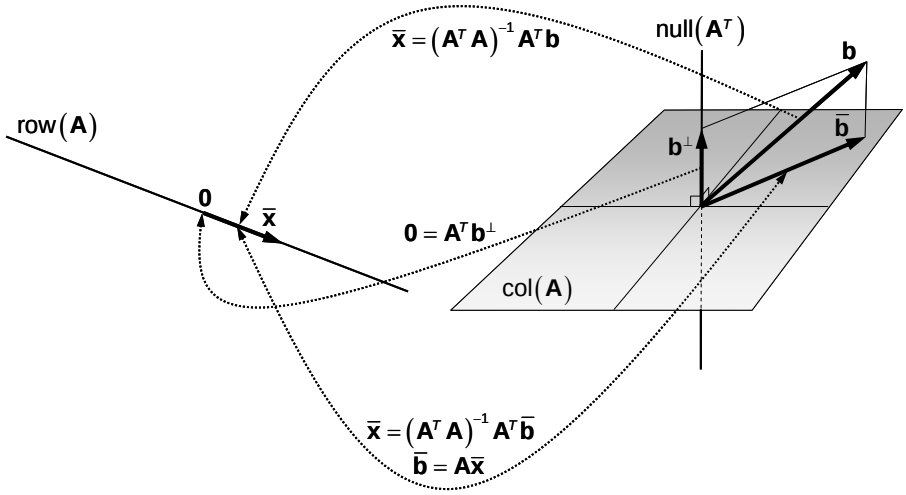


Figure A.3: The geometry of linear equations – overdetermined case. In this example $\text{col}(\mathbf{A})$ *of the 'tall' matrix* **A** *is two-dimensional,* $\text{row}(\mathbf{A})$ *and* $\text{null}(\mathbf{A}^T)$ *are both one-dimensional, and* $\text{null}(\mathbf{A})$ *is zero-dimensional, i.e. it only contains the zero vector, which implies that* **A** *is full rank. Because* **A** *has more rows than columns,* **b** *can not be in* $\text{col}(\mathbf{A})$ *(except for special cases). We can, however, project* **b** *on* $\text{col}(\mathbf{A})$*, and we look for the 'best' projection* $\overline{\mathbf{b}}$ *which is obtained when the error* $\mathbf{b}^{\perp}$ *is as small as possible, i.e. when it is perpendicular to* $\text{col}(\mathbf{A})$.

A.3.4 Singular system matrix – underdetermined case

For underdetermined case, where $m < n$, we have $\text{rank}(\mathbf{A}) = \min(n,m) = m$, which means that the 'flat' matrix **A** has only *m* independent columns. We can therefore find an infinite amount of linear combinations **x** that produce **b**, and it is not directly obvious what would be the best choice. Just like we could write any vector **b** as the sum of projections on $\text{col}(\mathbf{A})$ and $\text{null}(\mathbf{A}^T)$, we can write any vector **x** as the sum of projections on $\text{row}(\mathbf{A})$ and $\text{null}(\mathbf{A})$ (see Figure A.4):

$$\mathbf{x} = \overline{\mathbf{x}} + \mathbf{x}^{\perp} = \overline{\mathbf{P}}_x\mathbf{x} + \mathbf{P}_x^{\perp}\mathbf{x} \ , \tag{A.48}$$

where $\overline{\mathbf{P}}_x$ and $\mathbf{P}_x^{\perp}$ are projection matrices that are obtained as follows. The requirement that $\overline{\mathbf{x}}$ is in $\text{row}(\mathbf{A})$ can be expressed as

$$\overline{\mathbf{x}} = \mathbf{A}^T\overline{\mathbf{f}} \ , \tag{A.49}$$

where $\overline{\mathbf{f}}$ is an, as yet, arbitrary vector multiplying the columns of $\mathbf{A}^T$ and thus the rows of **A**. Furthermore, the requirement that $\mathbf{x}^\perp$ is in null(**A**) leads to

$$\mathbf{A}\mathbf{x}^\perp = \mathbf{A}\left(\mathbf{x} - \overline{\mathbf{x}}\right) = \mathbf{A}\left(\mathbf{x} - \mathbf{A}^T\overline{\mathbf{f}}\right) = \mathbf{A}\mathbf{x} - \mathbf{A}\mathbf{A}^T\overline{\mathbf{f}} = \mathbf{0} \ , \tag{A.50}$$

and provided that **A** is full rank we can solve for $\overline{\mathbf{f}}$ according to

$$\overline{\mathbf{f}} = \left(\mathbf{A}\mathbf{A}^T\right)^{-1}\mathbf{A}\mathbf{x} \ . \tag{A.51}$$

Multiplying both sides with $\mathbf{A}^T$ and using equation (A.49) we find that

$$\overline{\mathbf{x}} = \mathbf{A}^T\left(\mathbf{A}\mathbf{A}^T\right)^{-1}\mathbf{A}\mathbf{x} \ , \tag{A.52}$$

and therefore

$$\overline{\mathbf{P}}_x = \mathbf{A}^T\left(\mathbf{A}\mathbf{A}^T\right)^{-1}\mathbf{A} \ , \tag{A.53}$$

and

$$\mathbf{P}_x^\perp \triangleq \mathbf{I} - \overline{\mathbf{P}}_x = \mathbf{I} - \mathbf{A}^T\left(\mathbf{A}\mathbf{A}^T\right)^{-1}\mathbf{A} \ . \tag{A.54}$$

In the case of a regular system matrix **A** or in the case of an overdetermined system with full-rank **A**, **x** is exactly in row(**A**), and we could therefore require that also for the underdetermined case **x** is just in row(**A**), i.e. that $\mathbf{x} = \overline{\mathbf{x}}$ such that $\mathbf{x}^\perp = \mathbf{0}$ [†]. Using equation (A.52) we therefore find the solution for the underdetermined case as

$$\overline{\mathbf{x}} = \mathbf{A}^T\left(\mathbf{A}\mathbf{A}^T\right)^{-1}\mathbf{b} \ , \tag{A.55}$$

such that the pseudo-inverse for the underdetermined case becomes

$$\mathbf{A}^+ \triangleq \mathbf{A}^T\left(\mathbf{A}\mathbf{A}^T\right)^{-1} \ . \tag{A.56}$$

Note that, as before, the Gramian $\mathbf{A}\mathbf{A}^T$ is square and symmetric and has the 'small' dimension $n \times n$. Figure A.4 gives a graphical interpretation of the various relationships that play a role in solving an underdetermined system of equations using the requirement that **x** is as short as possible[‡].

[†] This is equivalent to requiring that **x** is as short as possible.

[‡] In practical applications there may be other requirements to select the most appropriate value of **x**. E.g. in Chapter 8, frequent use is made of underdetermined systems of equations to estimate large numbers of reservoir parameter values from small numbers of measurements. The most likely values of the uncertain parameter vectors are then obtained with the aid of *prior* information based on geological insight.

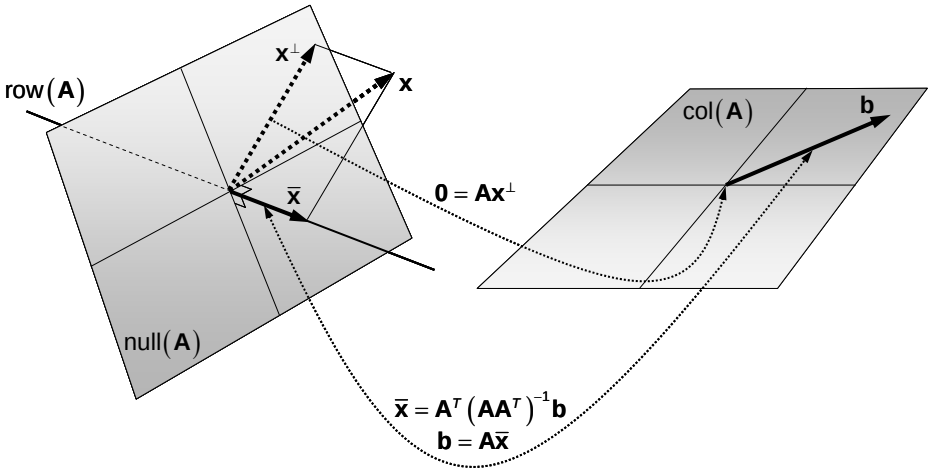


Figure A.4: The geometry of linear equations – underdetermined case. In this example col(**A**) *and* null(**A**) *of the 'flat' matrix* **A** *are two-dimensional,* row(**A**) *is one-dimensional, and* null($\mathbf{A}^T$) *is zero-dimensional, which means that* **A** *is full rank. Because* **A** *has more columns than rows,* **b** *is always in* col(**A**) *but* **x** *may be anywhere. The 'best' solution is obtained by requiring that* **x** *is as short as possible. In that case we have* $\mathbf{x}^\perp = \mathbf{0}$ *and* $\mathbf{x} = \bar{\mathbf{x}}$*, which is the reason that* **x** *and* $\mathbf{x}^\perp$ *have been indicated with dotted lines.*

A.4 Eigenvalues and eigenvectors

A.4.1 Determinant

A *determinant* is a single number related to a square matrix that determines if that matrix is invertible. The determinant of a matrix **A** is indicated as $|\mathbf{A}|$. For a detailed description of the properties and the use of determinants see e.g. Strang (2006). Here we will only mention the classic explicit formula for their computation. Starting simple, the determinant of a 2×2 matrix **A** is defined as[†]

$$|\mathbf{A}| = \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} \triangleq a_{11}a_{22} - a_{12}a_{21} \ . \tag{A.57}$$

Next, the determinant of a 3×3 matrix **A** is defined as

$$|\mathbf{A}| = \begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix} \triangleq a_{11}\begin{vmatrix} a_{22} & a_{23} \\ a_{32} & a_{33} \end{vmatrix} - a_{12}\begin{vmatrix} a_{21} & a_{23} \\ a_{31} & a_{33} \end{vmatrix} + a_{13}\begin{vmatrix} a_{21} & a_{22} \\ a_{31} & a_{32} \end{vmatrix} \ . \tag{A.58}$$

The determinant is thus computed as a summation of products, where each product consists of a term a_{1i} in the first row of **A** multiplying the determinant of a sub matrix $|\mathbf{A}_{1i}|$ obtained by deleting row 1 and column i of **A**. In addition the signs of the terms a_{1i} alternate from plus to minus. This definition can be extended to arbitrary $n\times n$ matrices with $n \geq 2$:

$$|\mathbf{A}| \triangleq \sum_{i=1}^{n} a_{1i}C_{1i} \ , \tag{A.59}$$

where the terms C_{1i} are the *cofactors* defined as

[†] The determinant of the even simpler 1×1 matrix **A** can be written as $|a_{11}|$ but we will not use this expression to avoid confusion with the absolute value of a_{11} which is also written as $|a_{11}|$.

$$C_{1i} \triangleq (-1)^{1+i} |\mathbf{A}_{1i}| \ . \tag{A.60}$$

Note that the determinants in the cofactors can be determined recursively using the same expression (A.59). An important property of the determinant of any square matrix **A** is that it signals singularity of **A** by being equal to zero. For any nonzero value of the determinant, **A** is regular and its inverse exists. In addition, the determinant can then be used to compute an explicit value for the inverse, although in a computationally inefficient way. In particular the expression for the inverse of a 2×2 matrix **A** is

$$\mathbf{A}^{-1} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}^{-1} = \frac{1}{|\mathbf{A}|} \begin{bmatrix} a_{22} & -a_{12} \\ -a_{21} & a_{11} \end{bmatrix} \ . \tag{A.61}$$

A.4.2 Eigenvalues and eigenvectors

Consider the multiplication of an $n \times 1$ vector **x** with an $n \times n$ matrix **A**:

$$\mathbf{y} = \mathbf{A}\mathbf{x} \ , \tag{A.62}$$

which may be considered as a transformation of **x** to another $n \times 1$ vector **y**. Often, it is possible to find scalars λ, called *eigenvalues*, and particular values **m** of **x**, called *eigenvectors*, such that[‡]

$$\mathbf{y} = \mathbf{A}\mathbf{m} = \mathbf{m}\lambda \ . \tag{A.63}$$

Every $n \times n$ matrix **A** has n eigenvalues and up to n eigenvectors, which can be computed from the matrix equation

$$(\mathbf{A} - \mathbf{I}\lambda_i)\mathbf{m}_i = \mathbf{0} \ , \ \ i = 1, \ldots, n \ , \tag{A.64}$$

which shows that the eigenvectors $\mathbf{m}_i$ are in null$(\mathbf{A} - \mathbf{I}\lambda_i)$. For equation (A.64) to have solutions other than the trivial solution $\mathbf{m} = \mathbf{0}$, it is required that the term in between brackets is singular, i.e. that its determinant is equal to zero:

$$|\mathbf{A} - \mathbf{I}\lambda_i| = 0 \, . \tag{A.65}$$

Note that equation (A.64) remains valid if the eigenvalues are multiplied with an arbitrary constant which shows that the eigenvalues can only be determined up to an arbitrary constant. The eigenvectors of a matrix can be interpreted as defining directions in n-dimensional space, referred to as the *principal directions,* which are often of special importance depending on the physics described by the equations in which the matrix plays a role. In general we need to perform the eigenvalue computation numerically, and a large body of research has been devoted to develop efficient algorithms for this purpose. In case of a 2×2 matrix we can perform the computation analytically by working out the determinant, leading to the *characteristic equation*

$$\begin{vmatrix} A_{11} - \lambda_i & A_{12} \\ A_{21} & A_{22} - \lambda_i \end{vmatrix} = \lambda_i^2 - (A_{11} + A_{22})\lambda_i + A_{11}A_{22} - A_{12}A_{21} = 0 \ , \tag{A.66}$$

which, because it is quadratic in λ_i, can be solved to give the two eigenvalues

[‡] Eigenvalues and eigenvectors are occasionally referred to as *characteristic* values and vectors respectively.

$$\lambda_{1,2} = \frac{\left(A_{11} + A_{22}\right) \pm \sqrt{\left(A_{11} + A_{22}\right)^2 - 4\left(A_{11}A_{22} - A_{12}A_{21}\right)}}{2} \ . \tag{A.67}$$

More in general, expansion of the determinant in equation (A.65) leads to a *characteristic polynomial* if degree n in terms of λ. If an $n \times n$ matrix **A** has n distinct eigenvalues, i.e. n eigenvalues that have a different numerical value, it also has n distinct eigenvectors. However, if **A** has repeated eigenvalues, i.e. eigenvalues with numerical values that occur twice or more, such that there are only m distinct eigenvalues, where $m < n$, the matrix has a minimum of m and a maximum of n eigenvectors[†]. If **A** is real and symmetric, the eigenvalues and eigenvectors are also real. If **A** is real but not symmetric, some or all of the eigenvalues and eigenvectors may become complex. For a real matrix **A** with n distinct eigenvalues, the n eigenvectors are independent. Moreover, if **A** is symmetric, the eigenvectors are real and orthogonal to each other, and thus form an orthogonal basis for $\mathbb{R}^n$.

A.4.3 Positive definiteness

A real symmetric matrix **A** is called *positive definite* if

$$\mathbf{x}^T \mathbf{A} \mathbf{x} > 0 \ , \tag{A.68}$$

for any $\mathbf{x} \neq \mathbf{0}$. It is called *positive semi-definite* if[‡]

$$\mathbf{x}^T \mathbf{A} \mathbf{x} \geq 0 \ . \tag{A.69}$$

If λ is an eigenvalue of **A** we can write

$$\mathbf{x}^T \mathbf{A} \mathbf{x} = \mathbf{x}^T \lambda \mathbf{x} = \lambda \left\| \mathbf{x} \right\|_2^2 \ , \tag{A.70}$$

which suggests that another condition for positive definiteness of **A** is that all its eigenvalues are positive, and it can be proved that this is indeed the case. Similarly, another condition for semi-positive definiteness of **A** is that all its eigenvalues are positive or zero. Moreover, a real symmetric matrix with all eigenvalues smaller than (or equal to) zero is called *negative (semi-) definite*, while the case with some eigenvalues positive and some others negative is referred to as *indefinite*.

A.4.4 Diagonalization

In the transformation $\mathbf{y} = \mathbf{A}\mathbf{x}$ as represented by equation (A.62), both vectors **x** and **y** can be expressed in terms of an arbitrary set of basis vectors. If we use the standard basis we have, following equation (A.25),

$$\mathbf{x} = \mathbf{i}_1 x_1 + \mathbf{i}_2 x_2 + \ldots + \mathbf{i}_n x_n \quad \text{and} \quad \mathbf{y} = \mathbf{i}_1 y_1 + \mathbf{i}_2 y_2 + \ldots + \mathbf{i}_n y_n \ . \tag{A.71, A.72}$$

However, if we choose any other basis, say the set of independent vectors $\{\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_n\}$, we can write **x** and **y** as linear combinations

$$\mathbf{x} = \mathbf{p}_1 \overline{x}_1 + \mathbf{p}_2 \overline{x}_2 + \ldots + \mathbf{p}_n \overline{x}_n = \mathbf{P}\overline{\mathbf{x}} \quad \text{and} \quad \mathbf{y} = \mathbf{p}_1 \overline{y}_1 + \mathbf{p}_2 \overline{y}_2 + \ldots + \mathbf{p}_n \overline{y}_n = \mathbf{P}\overline{\mathbf{y}} \ , \tag{A.73, A.74}$$

[†] The number of times that an eigenvalue is repeated is known as the *algebraic multiplicity*, or simply the multiplicity, of that eigenvalue. Under some conditions, there are multiple independent eigenvectors that correspond to the repeated eigenvalues, and their number is known as the *geometric multiplicity*.

[‡] Some texts use the terms '*strictly* positive definite' and 'positive definite' instead of 'positive definite' and 'positive semi-definite' respectively. Note that this implies that 'positive definite' may have a different meaning in different texts.

where **P** is the matrix

$$\mathbf{P} = \begin{bmatrix} \mathbf{p}_1 & \mathbf{p}_2 & \cdots & \mathbf{p}_n \end{bmatrix} , \tag{A.75}$$

and where $\bar{\mathbf{x}}$ and $\bar{\mathbf{y}}$ are the vectors of components corresponding to the new basis. The transformation (A.62) can then be rewritten as

$$\mathbf{P}\bar{\mathbf{y}} = \mathbf{A}\mathbf{P}\bar{\mathbf{x}} \ , \tag{A.76}$$

and because the vectors $\{\mathbf{p}_1, \mathbf{p}_2, \ldots \mathbf{p}_n\}$ are independent such that **P** is regular, we can define the inverse of **P**, and write

$$\bar{\mathbf{y}} = \mathbf{P}^{-1}\mathbf{A}\mathbf{P}\bar{\mathbf{x}} \ . \tag{A.77}$$

This gives us the formula for the matrix components of **A** as expressed in the new basis:

$$\bar{\mathbf{A}} = \mathbf{P}^{-1}\mathbf{A}\mathbf{P} \ , \tag{A.78}$$

and we can therefore rewrite the transformation (A.62) in terms of the new basis vectors as

$$\bar{\mathbf{y}} = \bar{\mathbf{A}}\bar{\mathbf{x}} \ . \tag{A.79}$$

If **A** is an $n \times n$ matrix with n distinct eigenvalues, a special basis is formed by the set of eigenvectors $\{\mathbf{m}_1, \mathbf{m}_2, \ldots, \mathbf{m}_n\}$ which can be collected in a matrix **M** according to[†]

$$\mathbf{M} = \begin{bmatrix} \mathbf{m}_1 & \mathbf{m}_2 & \cdots & \mathbf{m}_n \end{bmatrix}. \tag{A.80}$$

Applying equation (A.63) to all eigenvalues in **M** leads to

$$\mathbf{A}\mathbf{M} = \mathbf{M}\boldsymbol{\Lambda} \ , \tag{A.81}$$

where $\boldsymbol{\Lambda}$ is a diagonal matrix containing the n eigenvalues,

$$\boldsymbol{\Lambda} = \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_n \end{bmatrix} , \tag{A.82}$$

such that **A** can be rewritten as

$$\mathbf{A} = \mathbf{M}\boldsymbol{\Lambda}\mathbf{M}^{-1} \ , \tag{A.83}$$

with the inverse relationship given by

$$\boldsymbol{\Lambda} = \mathbf{M}^{-1}\mathbf{A}\mathbf{M} \ . \tag{A.84}$$

If **A** is real and symmetric, **M** is orthogonal, and the inverse relationship is given by

$$\boldsymbol{\Lambda} = \mathbf{M}^{T}\mathbf{A}\mathbf{M} \ . \tag{A.85}$$

Comparison of equations (A.78) and (A.84) shows that $\boldsymbol{\Lambda}$ can be interpreted as a special representation $\bar{\mathbf{A}}$ of transformation **A** in a new basis that has been chosen just so that $\bar{\mathbf{A}}$ becomes a diagonal matrix. We can therefore rewrite transformation (A.62) in terms of these special new basis vectors as

$$\bar{\mathbf{y}} = \boldsymbol{\Lambda}\bar{\mathbf{x}} \ , \tag{A.86}$$

[†] Such a matrix of eigenvectors is sometimes referred to as a *modal matrix*, and its columns as *modes*.

where $\bar{\mathbf{x}}$ and $\bar{\mathbf{y}}$ are now given by

$$\mathbf{x} = \mathbf{M}\bar{\mathbf{x}} \quad \text{and} \quad \mathbf{y} = \mathbf{M}\bar{\mathbf{y}} \ . \tag{A.87, A.88}$$

The major benefit of expressing the transformation in this special new basis is that the equations are *decoupled* such that the matrix vector equation (A.86) can computed as n separate scalar equations

$$\bar{y}_i = \lambda_i \bar{x}_i \ , \ i = 1,\ldots,n \ . \tag{A.89}$$

The same change of basis can be used to diagonalize systems of coupled ordinary differential equations, as described in Section 4.1.2 in the body of the text. Note that a change of basis is in fact a transformation itself, namely of the vectors of coordinates, and diagonalization is a special transformation known as a *similarity transformation*. This name stems from the fact that a dynamic system with a system matrix $\mathbf{\Lambda}$ has similar properties as one with a system matrix $\mathbf{A}$ because both matrices have the same eigenvalues. Not all square matrices can be diagonalized. However it will always be possible to transform a square matrix to a *block-diagonal* form, known as the *Jordan form*. Such a block-diagonal matrix contains as many blocks as there are distinct eigenvalues, while the number of diagonal elements of each block is equal to the multiplicity of the corresponding eigenvalue. We refer to Appendix B of Strang (2006) for further details.

A.5 Singular value decomposition

Equation (A.83) gave a matrix decomposition to diagonalize any square matrix with distinct eigenvalues. A somewhat similar, but often more powerful decomposition is possible for any matrix, irrespective of the multiplicity of its eigenvalues or even its dimensions or rank. It is known as the *singular value decomposition* (SVD) which allows every $m \times n$ matrix $\mathbf{A}$ to be written as

$$\mathbf{A} = \mathbf{\Phi}\mathbf{\Sigma}\mathbf{\Psi}^T \ , \tag{A.90}$$

where the columns of the $m \times m$ matrix $\mathbf{\Phi}$, known as the *left singular vectors,* are the eigenvectors of $\mathbf{A}\mathbf{A}^T$, the columns of the $n \times n$ matrix $\mathbf{\Psi}$, known as the *right singular vectors*, are the eigenvectors of $\mathbf{A}^T\mathbf{A}$, while $\mathbf{\Sigma}$ is an $m \times n$ matrix with on its main diagonal the *singular values* $\sigma_i, i = 1,\ldots,r$ of $\mathbf{A}$, which are the square roots of the non-zero eigenvalues of both $\mathbf{A}\mathbf{A}^T$ and $\mathbf{A}^T\mathbf{A}$. E.g., for the case that $m > n$ we have

$$\mathbf{\Sigma} = \begin{bmatrix} \sigma_1 & 0 & \cdots & 0 & 0 & \cdots & 0 \\ 0 & \sigma_2 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_r & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & 0 & \cdots & 0 \end{bmatrix} , \tag{A.91}$$

whereas for the case that $m < n$ we obtain a transposed version of this matrix. We refer to Antoulas (2005) and Strang (2006) for a discussion of the properties and the applications of

the SVD, and to Golub and van Loan (1996) for a treatment of the computational aspects. Here we will mention only some of the many properties and applications:

- The singular values are always larger or equal to zero. Usually they are ordered such that $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_r \geq 0$. Sometimes a short version of the SVD is applied in which case $\mathbf{\Sigma}$ is an $n \times n$ matrix, i.e. without the $m-n$ bottom rows of zeros for the case where $m > n$, or the $m-n$ right columns of zeros for the case that $m < n$. The number of columns in $\mathbf{\Phi}$ or in $\mathbf{\Psi}$ is then adjusted accordingly.
- The SVD gives orthonormal bases for all four fundamental subspaces:
  - the first *r* columns of $\mathbf{\Phi}$ for col(**A**);
  - the last $m-r$ columns of $\mathbf{\Phi}$ for null($\mathbf{A}^T$);
  - the first *r* columns of $\mathbf{\Psi}$ for row(**A**);
  - the last $n-r$ columns of $\mathbf{\Psi}$ for null(**A**).
- The SVD provides a robust way to compute the *effective* rank of a matrix, that is the rank that can be determined numerically up to machine precision. The effective rank is simply equal to the number of nonzero singular values, which is always smaller than or equal to the theoretical rank. In practical applications the theoretical rank of a matrix is usually of no value, and it is the effective rank that determines properties like e.g. the regularity.
- The pseudo inverse of an $m \times n$ matrix **A** was defined in equation (A.43) for the case that $m > n$ and in equation (A.56) for the case that $m < n$. The former case involved the inverse of $\mathbf{A}^T\mathbf{A}$, and the latter the inverse of $\mathbf{A}\mathbf{A}^T$, and both matrix products therefore had to be full rank. This restriction can be alleviated by computing the pseudo inverse with the aid of the SVD according to

$$\mathbf{A}^+ \triangleq \mathbf{\Psi}\mathbf{\Sigma}^{-1}\mathbf{\Phi}^T \quad , \tag{A.92}$$

  where $\mathbf{\Phi}$ and $\mathbf{\Psi}$ have been swapped compared to the SVD of **A**, and where $\mathbf{\Sigma}^{-1}$ is the inverted form of $\mathbf{\Sigma}$ with reciprocal values $1/\sigma_i$, $i = 1, \ldots, r$ of the non-zero singular values on its main diagonal and zeros otherwise[†]. The pseudo inverse as defined in equation (A.92) is known as the *Moore-Penrose* pseudo inverse.

With the aid of the Moore-Penrose pseudo inverse we can now generalize the geometric interpretation of linear systems of equations to systems with any matrix **A**, whether overdetermined or underdetermined, or whether with a full rank matrix **A** or not. Figures A.3 and A.4 can then be combined to Figure A.5 as shown below.

[†] With the aid of the Moore-Penrose pseudo inverse all systems of equations can be solved using formulation (A.43), whether over- or underdetermined, and the separate form (A.56) is no longer required.

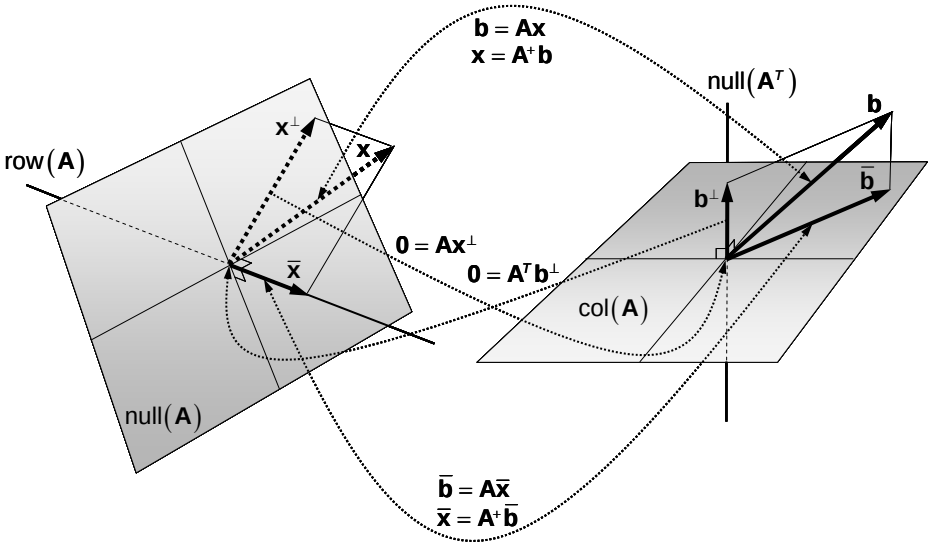


Figure A.5: The geometry of linear equations – general case. In this example col(**A**) *and* null(**A**) *of matrix* **A** *are two-dimensional, while* row(**A**) *and* null($\mathbf{A}^T$) *are one-dimensional. Matrix* **A** *may have more columns than rows or the other way round. In addition* **A** *may be rank-deficient. In any case the Moore-Penrose pseudo inverse* $\mathbf{A}^+$ *can be computed with the aid of the SVD.*

A.6 Vector derivatives of scalars, vectors and matrix-vector products

1. Let **x** be an $m\times 1$ vector and a a scalar that is a function of **x**. Then

$$\frac{\partial a}{\partial \mathbf{x}} = \begin{bmatrix} \frac{\partial a}{\partial x_1} & \frac{\partial a}{\partial x_2} & \cdots & \frac{\partial a}{\partial x_m} \end{bmatrix}. \tag{A.93}$$

   Note that $\partial a/\partial \mathbf{x}$ is a row vector. Using the gradient operator, which is defined as the transpose of the derivative vector, equation (A.93) can also be written as

$$\nabla a = \begin{bmatrix} \frac{\partial a}{\partial x_1} \\ \frac{\partial a}{\partial x_2} \\ \vdots \\ \frac{\partial a}{\partial x_m} \end{bmatrix}. \tag{A.94}$$

2. Let **x** be an $m\times 1$ vector and **a** an $n\times 1$ vector that is a function of **x**. Then $\partial \mathbf{a}/\partial \mathbf{x}$ is an $n\times m$ Jacobian defined as

$$\frac{\partial \mathbf{a}}{\partial \mathbf{x}} = \begin{bmatrix} \frac{\partial a_1}{\partial x_1} & \frac{\partial a_1}{\partial x_2} & \cdots & \frac{\partial a_1}{\partial x_m} \\ \frac{\partial a_2}{\partial x_1} & \frac{\partial a_2}{\partial x_2} & \cdots & \frac{\partial a_2}{\partial x_m} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial a_n}{\partial x_1} & \frac{\partial a_n}{\partial x_2} & \cdots & \frac{\partial a_n}{\partial x_m} \end{bmatrix} . \tag{A.95}$$

Note that the elements $\partial a_i/\partial \mathbf{x}$, $i = 1,2, \ldots, n$, of matrix $\partial \mathbf{a}/\partial \mathbf{x}$ are row vectors. Using the gradient operator, equation (A.95) can also be written in transposed form as

$$\nabla \cdot \mathbf{a} = \begin{bmatrix} \frac{\partial a_1}{\partial x_1} & \frac{\partial a_2}{\partial x_1} & \cdots & \frac{\partial a_n}{\partial x_1} \\ \frac{\partial a_1}{\partial x_2} & \frac{\partial a_2}{\partial x_2} & \cdots & \frac{\partial a_n}{\partial x_2} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial a_1}{\partial x_m} & \frac{\partial a_2}{\partial x_m} & \cdots & \frac{\partial a_n}{\partial x_m} \end{bmatrix} . \tag{A.96}$$

3. Let $\mathbf{x}$ be an $m\times 1$ vector and $\mathbf{A}$ an $n\times m$ matrix with elements that are a function of $\mathbf{x}$. Then $\mathbf{Ax}$ is an $n\times 1$ vector, and $\partial(\mathbf{Ax})/\partial \mathbf{x}$ is an $n\times m$ Jacobian that can be computed as

$$\frac{\partial(\mathbf{Ax})}{\partial \mathbf{x}} = \frac{\partial \mathbf{A}}{\partial \mathbf{x}}\mathbf{x} + \mathbf{A} = \frac{\partial \mathbf{A}}{\partial \begin{bmatrix} x_1 & x_2 & \cdots & x_m \end{bmatrix}^T}\mathbf{x} + \mathbf{A} = \begin{bmatrix} \frac{\partial \mathbf{A}}{\partial x_1}\mathbf{x} & \frac{\partial \mathbf{A}}{\partial x_2}\mathbf{x} & \cdots & \frac{\partial \mathbf{A}}{\partial x_m}\mathbf{x} \end{bmatrix} + \mathbf{A} . \tag{A.97}$$

4. Let $\mathbf{x}$ be an $m\times 1$ vector, and $\mathbf{A}$ and $\mathbf{B}$ matrices of size $p\times n$ and $n\times m$ respectively with elements that are functions of $\mathbf{x}$. Then $\mathbf{ABx}$ is a $p\times 1$ vector, and $\partial(\mathbf{ABx})/\partial \mathbf{x}$ is a $p\times m$ Jacobian that can be computed as

$$\begin{aligned} \frac{\partial(\mathbf{ABx})}{\partial \mathbf{x}} &= \frac{\partial(\mathbf{AB})}{\partial \mathbf{x}}\mathbf{x} + \mathbf{AB} = \frac{\partial(\mathbf{AB})}{\partial \begin{bmatrix} x_1 & x_2 & \cdots & x_m \end{bmatrix}^T}\mathbf{x} + \mathbf{AB} \\ &= \begin{bmatrix} \left(\frac{\partial \mathbf{A}}{\partial x_1}\mathbf{B} + \frac{\partial \mathbf{B}}{\partial x_1}\mathbf{A}\right)\mathbf{x} & \left(\frac{\partial \mathbf{A}}{\partial x_2}\mathbf{B} + \frac{\partial \mathbf{B}}{\partial x_2}\mathbf{A}\right)\mathbf{x} & \cdots & \left(\frac{\partial \mathbf{A}}{\partial x_m}\mathbf{B} + \frac{\partial \mathbf{B}}{\partial x_m}\mathbf{A}\right)\mathbf{x} \end{bmatrix} + \mathbf{AB} . \end{aligned} \tag{A.98}$$

5. Let $\mathbf{x}$ be an $m\times 1$ vector and $\mathbf{A}$ an $n\times m$ matrix with elements that are a function of $\mathbf{x}$. If the inverse $\mathbf{A}^{-1}$ exists, then $\mathbf{A}^{-1}\mathbf{x}$ is an $n\times 1$ vector, and $\partial(\mathbf{A}^{-1}\mathbf{x})/\partial \mathbf{x}$ is an $n\times m$ Jacobian that can be computed as

$$\begin{aligned} \frac{\partial\left(\mathbf{A}^{-1}\mathbf{x}\right)}{\partial \mathbf{x}} &= \frac{\partial \mathbf{A}^{-1}}{\partial \mathbf{x}}\mathbf{x} + \mathbf{A}^{-1} = -\mathbf{A}^{-1}\frac{\partial \mathbf{A}}{\partial \begin{bmatrix} x_1 & x_2 & \cdots & x_m \end{bmatrix}^T}\mathbf{A}^{-1}\mathbf{x} + \mathbf{A}^{-1} \\ &= -\begin{bmatrix} \mathbf{A}^{-1}\frac{\partial \mathbf{A}}{\partial x_1}\mathbf{A}^{-1}\mathbf{x} & \mathbf{A}^{-1}\frac{\partial \mathbf{A}}{\partial x_2}\mathbf{A}^{-1}\mathbf{x} & \cdots & \mathbf{A}^{-1}\frac{\partial \mathbf{A}}{\partial x_m}\mathbf{A}^{-1}\mathbf{x} \end{bmatrix} + \mathbf{A}^{-1} \end{aligned} . \tag{A.99}$$

6. Let $\mathbf{x}$ be an $m\times 1$ vector, and $\mathbf{A}$ and $\mathbf{B}$ matrices of size $p\times n$ and $n\times m$ respectively with elements that are functions of $\mathbf{x}$. If the inverse $\mathbf{A}^{-1}$ exists, then $\mathbf{A}^{-1}\mathbf{Bx}$ is a $p\times 1$ vector, and $\partial(\mathbf{A}^{-1}\mathbf{Bx})/\partial \mathbf{x}$ is a $p\times m$ Jacobian that can be computed as

$$\frac{\partial\left(\mathbf{A}^{-1}\mathbf{B}\mathbf{x}\right)}{\partial\mathbf{x}} = \frac{\partial\left(\mathbf{A}^{-1}\mathbf{B}\right)}{\partial\mathbf{x}}\mathbf{x} + \mathbf{A}^{-1}\mathbf{B} = \frac{\partial\left(\mathbf{A}^{-1}\mathbf{B}\right)}{\partial\left[x_1 \quad x_2 \quad \cdots \quad x_m\right]^T}\mathbf{x} + \mathbf{A}^{-1}\mathbf{B}$$
$$= \left[\left(\mathbf{A}^{-1}\frac{\partial\mathbf{A}}{\partial x_1}\mathbf{A}^{-1}\mathbf{B} + \frac{\partial\mathbf{B}}{\partial x_1}\mathbf{A}^{-1}\right)\mathbf{x} \quad \left(\mathbf{A}^{-1}\frac{\partial\mathbf{A}}{\partial x_2}\mathbf{A}^{-1}\mathbf{B} + \frac{\partial\mathbf{B}}{\partial x_2}\mathbf{A}^{-1}\right)\mathbf{x}\right.$$
$$\left. \cdots \quad \left(\mathbf{A}^{-1}\frac{\partial\mathbf{A}}{\partial x_m}\mathbf{A}^{-1}\mathbf{B} + \frac{\partial\mathbf{B}}{\partial x_m}\mathbf{A}^{-1}\right)\mathbf{x}\right] + \mathbf{A}^{-1}\mathbf{B}. \tag{A.100}$$

A.7 References for Appendix A

Golub, G.H. and Van Loan, C.F., 1996: *Matrix computations, 3rd ed.*, John Hopkins University Press, Baltimore.

Lay, D.C., 2003: *Linear algebra and its applications, 3rd ed.*, Addison Wesley, Boston.

Poole, D., 2003: *Linear algebra – a modern introduction*, Brooks/Cole, Pacific Grove.

Strang, G., 1984: Duality in the classroom. *The American Mathematical Monthly* **100** 848-855.

Strang, G., 1993: The fundamental theorem of linear algebra. *The American Mathematical Monthly* **91** 250-254.

Strang, G., 2003: *Introduction to linear algebra, 3rd ed.,* Wellesley-Cambridge Press, Wellesley.

Strang, G., 2006: *Linear algebra and its applications, 4th ed.,* Thomson Brooks/Cole, Pacific Grove.

Appendix B – Simple simulator `simsim`

B.1 Formulation

B.1.1 System equations

The system, input and accumulation matrices used in `simsim` have been defined in generalized state space form as†

$$\hat{\mathbf{A}}\left(\mathbf{x}\right) = -\left[\mathbf{T}\left(\mathbf{s}\right) + \mathbf{F}\left(\mathbf{s}\right)\mathbf{J}\left(\mathbf{s}\right)\right], \quad \hat{\mathbf{B}}\left(\mathbf{x}\right) = \begin{bmatrix}\mathbf{F}_w\left(\mathbf{s}\right)\\ \mathbf{F}_o\left(\mathbf{s}\right)\end{bmatrix}\left[\mathbf{I}_q + \mathbf{J}_p\left(\mathbf{s}\right)\right]\mathbf{L}_{qu}, \quad \hat{\mathbf{E}}\left(\mathbf{x}\right) = \mathbf{V}\left(\mathbf{s}\right) \tag{B.1, B.2, B.3}$$

where

† Here we indicate the dependence of **T** on **s** only, whereas actually **T** is also a function of **p** because of the upstream weighting of the relative permeabilities; see Section 2.4.9. However, because the dependence on **p** is non-differentiable it does not play a role in deriving the Jacobian matrix and we have disregarded it in the notation in this Appendix.

$$\mathbf{T}(\mathbf{s})=\begin{bmatrix}\mathbf{T}_w(\mathbf{s}) & \mathbf{0}\\ \mathbf{T}_o(\mathbf{s}) & \mathbf{0}\end{bmatrix},\ \mathbf{F}(\mathbf{s})=\begin{bmatrix}\mathbf{F}_w(\mathbf{s}) & \mathbf{0}\\ \mathbf{F}_o(\mathbf{s}) & \mathbf{0}\end{bmatrix},\ \mathbf{J}(\mathbf{s})=\begin{bmatrix}\mathbf{J}_p(\mathbf{s}) & \mathbf{0}\\ \mathbf{0} & \mathbf{0}\end{bmatrix},$$

$$\mathbf{F}_w(s)=\begin{bmatrix}\mathbf{0} & \mathbf{0} & \mathbf{0}\\ \mathbf{0} & \mathbf{F}_{w,22}(\mathbf{s}) & \mathbf{0}\\ \mathbf{0} & \mathbf{0} & \mathbf{F}_{w,33}(\mathbf{s})\end{bmatrix},\ \mathbf{F}_o(s)=\begin{bmatrix}\mathbf{0} & \mathbf{0} & \mathbf{0}\\ \mathbf{0} & \mathbf{F}_{o,22}(\mathbf{s}) & \mathbf{0}\\ \mathbf{0} & \mathbf{0} & \mathbf{F}_{o,33}(\mathbf{s})\end{bmatrix},$$

$$\mathbf{I}_q=\begin{bmatrix}\mathbf{0} & \mathbf{0} & \mathbf{0}\\ \mathbf{0} & \mathbf{I} & \mathbf{0}\\ \mathbf{0} & \mathbf{0} & \mathbf{0}\end{bmatrix},\ \mathbf{J}_p(\mathbf{s})=\begin{bmatrix}\mathbf{0} & \mathbf{0} & \mathbf{0}\\ \mathbf{0} & \mathbf{0} & \mathbf{0}\\ \mathbf{0} & \mathbf{0} & \mathbf{J}_3(\mathbf{s})\end{bmatrix},\ \mathbf{L}_{qu}=\begin{bmatrix}\mathbf{0} & \mathbf{0}\\ \mathbf{I} & \mathbf{0}\\ \mathbf{0} & \mathbf{I}\end{bmatrix},\ \mathbf{V}(\mathbf{s})=\begin{bmatrix}\mathbf{V}_{wp}(\mathbf{s}) & \mathbf{V}_{ws}\\ \mathbf{V}_{op}(\mathbf{s}) & \mathbf{V}_{os}\end{bmatrix}.$$

(B.4, B.5, B.6, B.7, B.8, B.9, B.10, B.11, B.12)

In addition, use is made of the system and input matrices defined in regular state space form:

$$\mathbf{A}(\mathbf{x})=-\hat{\mathbf{E}}^{-1}(\mathbf{x})\hat{\mathbf{A}}(\mathbf{x}),\ \mathbf{B}(\mathbf{x})=\hat{\mathbf{E}}^{-1}(\mathbf{x})\hat{\mathbf{B}}(\mathbf{x}). \qquad \text{(B.13, B.14)}$$

The state, input and output vectors have in both cases been defined as

$$\mathbf{x}=\begin{bmatrix}\mathbf{p}\\ \mathbf{s}\end{bmatrix},\ \mathbf{u}=\begin{bmatrix}\breve{\mathbf{p}}_{well}\\ \breve{\mathbf{q}}_{well}\end{bmatrix},\ \mathbf{y}=\begin{bmatrix}\mathbf{y}_1\\ \mathbf{y}_2\\ \mathbf{y}_3\\ \mathbf{y}_4\\ \mathbf{y}_5\end{bmatrix}=\begin{bmatrix}\mathbf{p}_{well}\\ \mathbf{q}_{well,w}\\ \mathbf{q}_{well,o}\\ \mathbf{p}_{well\ grid\ blocks}\\ \mathbf{s}_{well\ grid\ blocks}\end{bmatrix}. \qquad \text{(B.15, B.16, B.17)}$$

The corresponding continuous-time system equations can be expressed as

$$\hat{\mathbf{E}}(\mathbf{x})\dot{\mathbf{x}}=\hat{\mathbf{f}}(\mathbf{x})\ , \qquad \text{(B.18)}$$

for the generalized state space form, where

$$\hat{\mathbf{f}}(\mathbf{x})=\hat{\mathbf{A}}(\mathbf{x})\mathbf{x}+\hat{\mathbf{B}}(\mathbf{x})\mathbf{u}\ , \qquad \text{(B.19)}$$

and as

$$\dot{\mathbf{x}}=\mathbf{f}(\mathbf{x})\ , \qquad \text{(B.20)}$$

for the regular state space form, where

$$\mathbf{f}(\mathbf{x})=\mathbf{A}(\mathbf{x})\mathbf{x}+\mathbf{B}(\mathbf{x})\mathbf{u}\ . \qquad \text{(B.21)}$$

In addition, use is made of the equivalent continuous-time system equations in implicit form, given by

$$\hat{\mathbf{g}}(\mathbf{x})=\hat{\mathbf{E}}(\mathbf{x})\dot{\mathbf{x}}-\hat{\mathbf{A}}(\mathbf{x})\mathbf{x}-\hat{\mathbf{B}}(\mathbf{x})\mathbf{u}\ , \qquad \text{(B.22)}$$

for the generalized state space case, and

$$\mathbf{g}(\mathbf{x})=\dot{\mathbf{x}}-\mathbf{A}(\mathbf{x})\mathbf{x}-\mathbf{B}(\mathbf{x})\mathbf{u}\ . \qquad \text{(B.23)}$$

for the regular case.

B.1.2 Jacobian $\mathbf{J}_{\hat{\mathbf{g}}}$

Four time integration routines are available in `simsim`: explicit, implicit with Picard iteration, implicit with Newton-Raphson iteration and IMPES. The implicit version with Newton-Raphson iteration has been defined in generalized state space form, with time-discretized system equations that can be expressed as:

$$\left(\hat{\mathbf{E}}_{k+1} - \Delta t \hat{\mathbf{A}}_{k+1}\right)\mathbf{x}_{k+1} = \hat{\mathbf{E}}_{k+1}\mathbf{x}_k + \Delta t \hat{\mathbf{B}}_{k+1}\mathbf{u}_{k+1} \quad , \tag{B.24}$$

where we use the short-cut notation $\mathbf{A}_{k+1}$ to indicate $\mathbf{A}(\mathbf{x}_{k+1})$ etc. The Newton-Raphson iteration scheme requires a Jacobian matrix $\partial\hat{\mathbf{g}}_{k+1}/\partial\mathbf{x}_{k+1}$ of the system equations in implicit form, i.e. of

$$\hat{\mathbf{g}}_{k+1} = \left(\hat{\mathbf{E}}_{k+1} - \Delta t \hat{\mathbf{A}}_{k+1}\right)\mathbf{x}_{k+1} - \hat{\mathbf{E}}_{k+1}\mathbf{x}_k - \Delta t \hat{\mathbf{B}}_{k+1}\mathbf{u}_{k+1} \ . \tag{B.25}$$

The Jacobian can be computed as

$$\mathbf{J}_{\hat{\mathbf{g}}} = \frac{\partial\hat{\mathbf{g}}_{k+1}}{\partial\mathbf{x}_{k+1}} = \frac{\partial\left(\hat{\mathbf{E}}_{k+1}\mathbf{x}_{k+1}\right)}{\partial\mathbf{x}_{k+1}} - \frac{\partial\left(\Delta t \hat{\mathbf{A}}_{k+1}\mathbf{x}_{k+1}\right)}{\partial\mathbf{x}_{k+1}} - \frac{\partial\left(\hat{\mathbf{E}}_{k+1}\mathbf{x}_k\right)}{\partial\mathbf{x}_{k+1}} - \frac{\partial\left(\Delta t \hat{\mathbf{B}}_{k+1}\mathbf{u}_{k+1}\right)}{\partial\mathbf{x}_{k+1}} \ . \tag{B.26}$$

See Section A.6 in Appendix A for a general overview of the computation of vector derivatives of matrix-vector products as appear in equation (B.26).

B.1.3 Time stepping

To be continued

Nomenclature

Notes:

- Several symbols occur more than once because they have a different meaning in different parts of the text.
- The dimensions of vectors and matrices have only been indicated when all elements have the same dimensions.

Symbol	*Description*	*Dimensions*	*SI units*
a	coefficient	-	-
A	area	L^2	m^2
$\mathbf{A}$	system matrix	-	-
$\mathbf{A}$	arbitrary matrix	-	-
$\mathbf{A}$	linear constraint matrix	-	-
$\overline{\mathbf{A}}$	Jacobian	-	-
b	coefficient	-	-
b	discount factor	-	-
$\mathbf{b}$	right-hand side vector	-	-
$\mathbf{b}$	vector of representer coefficients	-	-
$\mathbf{b}$	vector of constants in linear constraints	-	-
B	oil formation volume factor	-	-
$\mathbf{B}$	input matrix	-	-
$\overline{\mathbf{B}}$	Jacobian	-	-
c	compressibility	$t^2m^{-1}L^{-1}$	1/Pa
c	equality constraint	-	-
$\mathbf{c}$	vector of equality constraints	-	-
C	integration constant	$L^{-3}m$	kg/m^3
C_{1i}	cofactor	-	-
$\mathcal{C}$	controllability matrix	-	-
$\mathbf{C}$	output matrix	-	-
$\overline{\mathbf{C}}$	Jacobian	-	-
d	depth	L	m
d	inequality constraint	-	-
$\mathbf{d}$	data vector ('measured measurements')	-	-
$\mathbf{d}$	vector of inequality constraints	-	-
D	diffusion constant	L^2t^{-1}	m^2/s
$\mathbf{D}$	direct throughput matrix	-	-
$\overline{\mathbf{D}}$	Jacobian	-	-
$\mathcal{D}$	domain	-	-
E	energy	L^2mt^{-2}	J
$\mathbf{E}$	accumulation matrix	-	-
$\overline{\mathbf{E}}$	Jacobian	-	-
e	nonlinear function	-	-

$\mathbf{e}$	nonlinear vector-valued function	-	-
$\mathbf{e}$	error vector	-	-
f	nonlinear function	-	-
f	fractional flow	-	-
$\mathbf{f}$	nonlinear vector-valued function	-	-
$\mathbf{f}$	arbitrary vector	-	-
$\mathbf{F}$	fractional flow matrix	-	-
$\overline{\mathbf{F}}$	Jacobian	-	-
g	acceleration of gravity	Lt^{-2}	m/s^2
g	velocity gradient	t^{-1}	1/s
g	nonlinear scalar function	-	-
$\mathbf{g}$	nonlinear vector-valued function	-	-
$\mathbf{g}$	nonlinear super vector-valued function	-	-
$\mathbf{G}$	impulse response matrix	-	-
h	reservoir height	L	m
$\mathbf{h}$	nonlinear vector-valued output function	-	-
$\mathbf{H}$	Hessian matrix	-	-
$\mathcal{H}$	Hankel matrix	-	-
i	imaginary unit	-	-
i	counter	-	-
$\mathbf{i}$	unit vector	-	-
j	counter	-	-
$\mathbf{j}$	nonlinear vector-valued output function	-	-
$\mathbf{j}$	nonlinear super vector-valued output function	-	--
$\mathbf{I}$	identity matrix	-	-
J	well index, productivity index	L^2m^{-1}	$m^3/(Pa\ s)$
$\mathbf{J}$	well index matrix	L^2m^{-1}	$m^3/(Pa\ s)$
$\mathcal{J}$	objective function	-	-
$\overline{\mathcal{J}}$	modified objective function	-	-
$\overline{\overline{\mathcal{J}}}$	dual objective function	-	-
k	permeability	L^2	m^2
k	counter	-	-
k	discrete time	-	-
k	arbitrary constant	-	-
K	total number of time steps	-	-
$\vec{\mathbf{K}}$	permeability tensor	L^2	m^2
$\mathcal{K}$	secondary objective function	-	-
L	length	m	m
L	spatial differential operator	-	-
$\mathbf{L}$	location matrix (selection matrix)	-	-
$\mathbf{L}$	matrix of left singular vectors	-	-
$\mathcal{L}$	Lagrangian	-	-
m	number of elements in input vector $\mathbf{u}$	-	-

m	number (general, sometimes subscripted)	-	-
$\mathbf{m}$	eigenvector	-	-
$\mathbf{m}$	vector of generalized model parameters	-	-
M	number of elements in super input vector $\mathbf{u}$		
$\mathbf{M}$	matrix of eigenvectors	-	-
n	number of elements in state vector $\mathbf{x}$	-	-
n	number (general, sometimes subscripted)	-	-
n	Corey exponent	-	-
$\mathbf{n}$	unit vector normal to boundary	-	-
N	number of elements in super state vector $\mathbf{x}$	-	-
$\mathcal{O}$	observability matrix	-	-
p	number of elements in output vector $\mathbf{y}$	-	-
p	pressure	$L^{-1}mt^{-2}$	Pa
$\mathbf{p}$	pressure vector	$L^{-1}mt^{-2}$	Pa
$\mathbf{p}$	arbitrary basis vector	$L^{-1}mt^{-2}$	Pa
$\boldsymbol{p}$	vector of pressure differences	-	-
P	power	L^2mt^{-3}	W
$\mathbf{P}$	covariance matrix	-	-
$\mathbf{P}$	orthogonal matrix of arbitrary basis vectors	-	-
$\overline{\mathbf{P}}$	orthogonal-projection matrix	-	-
$\mathbf{P}^{\perp}$	complementary orthogonal-projection matrix	-	-
q	flow rate (source term)	L^3t^{-1}	m^3/s
$\tilde{q}$	flow rate over a grid block boundary	L^3t^{-1}	m^3/s
q''	flow rate per unit area (source term)	Lt^{-1}	m/s
q'''	flow rate per unit volume (source term)	t^{-1}	1/s
$\mathbf{q}$	vector of flow rates (source terms)	L^3t^{-1}	m^3/s
Q	scaled flow rate	$L^{-1}mt^{-3}$	Pa/s
Q	number of elements in super output vector $\mathbf{y}$	-	-
$\mathbf{Q}$	auxiliary matrix	-	-
r	radius, radial coordinate	L	m
r	residual	-	-
$\mathbf{r}$	residual vector	-	-
R	ratio	-	-
$\mathbf{R}$	representer matrix	-	-
s	coordinate along a curve	-	-
s	Laplace variable	t^{-1}	1/s
$\mathbf{s}$	saturation vector	-	-
$\mathbf{s}$	vector of slack variables	-	-
S	saturation	-	-
$\mathcal{S}$	set	-	-
$\mathbf{S}$	matrix to compute $\tilde{\boldsymbol{\nu}}$ from $\mathbf{p}$	$L^2M^{-1}t$	m/(Pa s)
t	time	t	s
T	transmissibility	$L^2M^{-1}t$	m^3/(Pa s)

$\mathbf{T}$	transmissibility matrix	$L^2M^{-1}t$	$m^3/(Pa\ s)$
u	input variable	-	-
$\mathbf{u}$	input vector	-	-
u	super vector of input variables	-	-
v	superficial velocity	Lt^{-1}	m/s
$\tilde{v}$	interstitial velocity	Lt^{-1}	m/s
$\vec{\mathbf{v}}$	superficial velocity vector in physical space	Lt^{-1}	m/s
$\mathbf{v}$	superficial velocity vector	Lt^{-1}	m/s
$\mathbf{v}$	arbitrary vector	-	-
$\mathbf{v}$	unit tangent vector	-	-
$\boldsymbol{v}$	Darcy velocity vector at grid block boundaries	Lt^{-1}	m/s
$\tilde{\boldsymbol{v}}$	interstitial velocity vector at grid bl. boundaries	Lt^{-1}	m/s
V	volume	L^3	m^3
$\mathbf{V}$	accumulation matrix	-	-
$\mathcal{V}$	vector space	-	-
$\mathbf{W}$	Gramian	-	-
$\mathcal{W}$	vector space	-	-
x	spatial coordinate	L	m
x	(state) variable	-	-
$\vec{\mathbf{x}}$	coordinate vector in physical space	L	m
$\mathbf{x}$	state vector	-	-
$\mathbf{x}$	arbitrary vector	-	-
x	super vector of state variables	-	-
y	spatial coordinate	L	m
y	output variable	-	-
$\mathbf{y}$	output vector	-	-
$\mathbf{y}$	arbitrary vector	-	-
y	super vector of output variables	-	-
z	spatial coordinate	L	m
z	transformed (state) variable	-	-
$\mathbf{z}$	transformed state vector	-	-
α	geometric factor	-, L, L^2	-, m, m^2
α	angle	-	-
α	dimensionless valve opening	-	-
$\boldsymbol{\alpha}$	vector of dimensionless valve openings	-	-
β	interpolation variable	-	-
$\boldsymbol{\beta}$	reduced-order parameter vector	-	-
γ	constant	-	-
$\boldsymbol{\gamma}$	vector of geometric factors	L^{-1}	1/m
Γ	boundary	L^2	m^2
δ	Dirac delta function	-	-
δ_k	Kronecker delta	-	-
ε	nonlinear function	-	-

ε	convergence criterion	-	-
ε	error	-	-
$\boldsymbol{\varepsilon}$	vector of model errors	-	-
$\boldsymbol{\eta}$	vector of measurement errors	-	-
ζ	diffusion constant	L^2t^{-1}	m^2/s
θ	parameter	-	-
θ	penalty parameter	-	-
$\boldsymbol{\theta}$	parameter vector	-	-
λ	mobility	$LM^{-1}t$	$m^2/(Pa\ s)$
λ	eigenvalue	-	-
λ	Lagrange multiplier	-	-
$\boldsymbol{\lambda}$	vector of mobilities at grid block boundaries	$LM^{-1}t$	$m^2/(Pa\ s)$
$\boldsymbol{\lambda}$	vector of Lagrange multipliers	-	-
$\boldsymbol{\lambda}$	super vector of Lagrange multipliers	-	-
$\boldsymbol{\Lambda}$	diagonal matrix of eigenvalues	-	-
μ	dynamic viscosity	$L^{-1}mt^{-1}$	Pa s
μ	Lagrange multiplier	-	-
$\boldsymbol{\mu}$	vector of Lagrange multipliers	-	-
$\boldsymbol{\mu}$	super vector of Lagrange multipliers	-	-
ν	Lagrange multiplier for equality constraint	-	-
$\boldsymbol{\nu}$	vector of Lagrange multipliers for eq. constr.	-	-
π	dummy variable	-	-
ρ	density	$L^{-3}m$	kg/m^3
σ	singular value	-	-
$\boldsymbol{\Sigma}$	diagonal matrix of singular values in SVD	-	-
$\Delta\tau$	grid block travel time along a stream line	t	s
τ	reference time interval for discounting	t	s
τ	time of flight along a streamline	t	s
ϕ	porosity	-	-
Φ	potential	$L^{-1}mt^{-2}$	Pa
$\boldsymbol{\Phi}$	state transition matrix	-	-
$\boldsymbol{\Phi}$	reduction matrix	-	-
$\boldsymbol{\Phi}$	matrix of left singular vectors in SVD	-	-
φ	nonlinear function	-	-
ψ	source term	-	-
$\boldsymbol{\Psi}$	matrix of right singular vectors in SVD	-	-
$\boldsymbol{\varphi}$	vector of averaged grid block porosities	-	-
Ω	domain	L^3	m^3
ω	frequency	t^{-1}	rad/s
ω	Lagrange multiplier for inequality constraint	-	-
$\boldsymbol{\omega}$	vector of Lagrange multipliers for ineq. constr.	-	-
$\nabla\mathbf{x}$	gradient vector	-	-

Subscripts

a	augmented
av	average
c	capillary
c	continuous
c	controllability
con	connectivity
d	discrete
dis	dissipation
D	dimensionless
e	exit
eq	equivalent
gb	grid block
i	initial
i	in (entry)
inj	injection
k	discrete time
l	liquid
lift	lift
m	mass
o	observability
o	oil
or	residual oil
p	pore
p	pressure
pot	potential
prod	production
q	flow rate
r	relative
r	rock
R	reservoir
s	saturation
sc	standard conditions
scal	scaling
sys	system
t	total
tf	flowing tubing head
w	water
wc	connate water
well	well
wf	flowing well bore
x	x-direction
y	y-direction
ε	model error

η measurement error
δ impulse response
θ parameter
λ mobility

Superscripts

0 end point saturation
i iteration counter
T transpose

Glossary

AIM	Adaptive Implicit Method
AIME	American Institute of Mining, Metallurgical, and Petroleum Engineers
BHP	Bottom Hole Pressure
CFL	Courant-Friedrichs-Lewy
FDP	Field Development Plan
GRG	Generalized Reduced Gradient
ICV	Inflow Control Valve
IMPES	IMplicit Pressure – Explicit Saturation
IOR	Improved Oil Recovery
KKT	Karush-Kuhn-Tucker
LTI	Linear Time-Invariant
LTV	Linear Time-Varying
NPV	Net Present Value
ODE	Ordinary Differential Equation
PDE	Partial Differential Equation
pdf	probability density function
PDG	Permanent Downhole Gauge
SPE	Society of Petroleum Engineers
SVD	Singular Value Decomposition

Index