Institut für Strömungsmechanik
und Elektron. Rechnen im Bauwesen
der Universität Hannover

Bericht Nr. 62/2001

Carsten Thorenz

Model Adaptive Simulation of Multiphase and Density Driven Flow in Fractured and Porous Media

ISSN 0177-9028

Institut für Strömungsmechanik
und Elektron. Rechnen im Bauwesen
Universität Hannover
Appelstr. 9A
D-30167 Hannover

Tel.: +49-511-762-3568
Fax.: +49-511-762-3777
URL: http://www.hydromech.uni-hannover.de

Vom Fachbereich Bauingenieur- und Vermessungswesen der Universität Hannover genehmigte Dissertationsschrift zur Erlangung des Grades Doktor-Ingenieur.

Referent:	Prof. Dr.-Ing. Werner Zielke
Korrefenten:	PD Dr.-Ing. habil. Olaf Kolditz Prof. Dr. Brian Berkowitz
Tag der Promotion:	17.08.2001

Abstract

In this work the fundamentals of a numerical model for flow processes in the subsurface are developed. Additionally, various illustrative examples are presented. The model is based on the finite element method and covers the simulation of saturated single-phase flow, unsaturated single-phase flow, density driven flow and multiphase flow in fractured and porous media. It is coupled to a single-phase, single-component tracer transport model, which is extended to the scope of multiphase, multi-component tracer transport and coupled to the flow model to take into account for transport processes. The numerical scheme is enhanced in its applicability for fracture networks by a scheme based on the method of characteristics, which is extended for the special case of transport in fracture networks.

The implementation was done in the framework of the development of the subsurface flow and transport simulator package *RockFlow 3*, which is developed by the groundwater modelling group at the Institute for Fluid Mechanics in Civil Engineering of the University of Hannover. The simulator works with grid adaptive strategies and is coded in the programming language C. The implementation uses object-oriented concepts for time uncritical parts of the simulator, thus making later enhancements of the code easier.

Due to the multitude of regarded physical processes within the flow model, a scalable approach is derived, which can treat flow processes of different complexity by up- and downsizing of a superset model. Thus, the chosen physical *model* is *adapted* to the relevant physical processes in the domain (*model adaptivity*). The downsizing methods are automatized and are suitable to adapt the underlying physical model dynamically.

A set of test applications for the developed model is regarded. Included are test cases for single-phase, density driven and multiphase flow and transport processes in porous and fractured media. Where analytical solutions are available, these are compared to the results of the model in order to quantify the accuracy.

Because of the lack of available benchmarks for density driven flow in partially saturated media, a laboratory scale experiment was performed. The laboratory experiments were designed according to initial simulations from the numerical model, thus underlining the predictive capacity of the model. Though the predicted behaviour of the system was quite unexpected, the experimental work proved the correctness of the prior numerical simulations. Simulations on the basis of the actual experimental data reproduce the behaviour of the experiments qualitatively and quantitatively. In future the experimental data can serve as a benchmark for other numerical models.

Keywords: fractured porous medium, adaptive numerical model, density driven and multiphase flow.

Kurzfassung

In der vorliegenden Arbeit werden die Grundlagen für ein numerisches Modell zur Simulation von Strömungsprozessen im geklüftet-porösen Untergrund entwickelt und zusammen mit Beispielen für die Anwendung präsentiert. Das Modell basiert auf der Finite-Elemente-Methode und beschreibt einphasige und mehrphasige sowie dichtebeeinflusste Strömungen. Durch die Erweiterung eines bereits vorhandenen Tracertransportmodell um den Transport mehrerer Komponenten in mehreren Fluidphasen und die Kopplung mit dem Strömungsmodell ergibt sich ein Mehrphasen-Mehrkomponenten-Modell mit optionalem Dichteeinfluss. Um die Effizienz der Simulation von Transportvorgängen in Kluftnetzwerken zu verbessern, wird ein speziell angepasstes Charakteristikenverfahren vorgestellt.

Die Implementierung wurde im Rahmen der von der Grundwassermodellierungsgruppe des Instituts für Strömungsmechanik der Universität Hannover betriebenen Entwicklung des Simulationspakets *RockFlow 3* durchgeführt. Der Simulator arbeitet mit gitteradaptiven Methoden und ist in der Programmiersprache C entwickelt worden. Die Implementierung verwendet objekt-orientierte Ansätze für zeitunkritische Teile des Simulators (z.B. Randbedingungen, Materialdatenverwaltung) und bietet so gute Voraussetzungen für spätere Erweiterungen.

Um der Vielzahl zu betrachtender physikalischer Prozesse gerecht zu werden, wird ein skalierbarer Ansatz entwickelt, der durch gezieltes Vereinfachen eines übergreifenden Modells das zu den abzubildenden Prozessen passende Modell auswählt. Dieses Vorgehen kann als *Adaption* des gewählten physikalischen *Modells* an die im Simulationsgebiet herrschenden Verhältnisse bezeichnet werden (*Modelladaption*). Es werden Möglichkeiten zur Automatisierung dieser Anpassung des zugrundegelegten physikalischen Modells vorgestellt, die zur dynamischen Adaption geeignet sind.

Zur Überprüfung des entwickelten Modells werden diverse Testbeispiele betrachtet. Darunter sind Beispiele für Ein- und Mehrphasenströmungen sowie für dichtebeeinflusste Strömungen und Transportprozesse in geklüfteten und porösen Medien. Soweit analytische Lösungen verfügbar sind, werden diese herangezogen um die Genauigkeit des numerischen Modells quantifizieren zu können. Da keine Testbeispiele für dichtegetriebene Strömungen in teilgesättigten Systemen bekannt sind, wurde zu diesem Zweck ein Laborexperiment durchgeführt. Der Laborversuch wurde auf der Basis von numerischen Vorabsimulationen entworfen. Obwohl das so vorhergesagte Systemverhalten höchst unerwartet war, haben die späteren Experimente die numerische Vorhersage bestätigt. Simulationen auf der Basis der tatsächlichen experimentellen Daten reproduzieren das Verhalten des Experiments sowohl qualitativ als auch quantitativ. Damit steht auch für andere numerische Modelle ein Testfall zur Modellüberprüfung zur Verfügung.

Schlagworte: geklüftet-poröses Medium, adaptives numerisches Modell, Dichte- und Mehrphasenströmung

Preface and acknowledgements

The largest part of this work was written between fall 2000 and spring 2001, mainly based on the work of the years 1997 to 2000. In this earlier period, I was working as a member of the Groundwater Modelling Group in the Institute for Fluid Mechanics and Computer Applications in Civil Engineering at the University of Hannover, Germany. I would like to thank Prof. Dr.-Ing. W. Zielke, the head of the institute, for his long-time support of my work (in total a little more than five years). Especially I'd like to thank Prof. Dr.-Ing. O Kolditz, at that time head of the Groundwater Modelling Group, for the intense scientific exchange and for the major effort to keep the group as a whole going. During spring 2000 I had the opportunity to join Prof. Brian Berkowitz' research group at the Weizmann Institute of Science (Israel) for a scientific exchange, which was a very pleasant and (scientifically) inspiring possibility to broaden my horizon. Furthermore, I'd like to thank all of the three mentioned above for reviewing my thesis, which for sure took substantial effort.

While working in the institute, I've experienced support from many sides, so that it is difficult to pick out single persons. Especially, I'd like to thank Dipl.-Math. R. Kaiser for many fruitful discussions and Dipl.-Ing. R. Ratke for his will to share his in-depth knowledge about the "nitty-gritty" of numerical algorithms.

The proofreaders of this document shall not be remained unnamed, because they are to be blamed for the remaining errors ☺. A big "thank you" for their effort goes to Dipl.-Math. H. Jakobs, Dipl.-Ing. M. Kohlmeier, Dipl.-Ökon. O. Siegert (and to Rob!) and especially to Dipl.-Phys. H. Pleteit for the feedback from the viewpoint of the "scientific outsider". Additionally, I'd like to thank Dipl.-Ing. F. Sellerhoff for his feedback while composing the talk, which was part of the thesis defense.

Furthermore, I have to thank the taxpayers of the Federal Republic of Germany, who have financed my work through the German Federal Ministry for Education and Research (BMBF), the German Federal Institute for Geosciences and Natural Resources (BGR) and the "Deutsche Forschungsgemeinschaft" (DFG).

And, last and most important, I have to thank my wife Meike and my daughter Lea for distracting me when the workload became a menace and my parents for supporting me all the way.

Carsten Thorenz

Hannover, 21.10.2001

There is a theory which states that if ever anyone discovers exactly
what the Universe is for and why it is here, it will instantly
disappear and be replaced by something even more bizarre and inexplicable.

There is another theory which states that this has already happened.

(*Douglas Adams*, †2001)

Contents

Table of symbols

Symbol	Quantity	Units
α	Used as index: Regarded fluid phase	[-]
α	In finite element method: Upwinding factor	[-]
α_l	Longitudinal dispersion length	[m]
α_t	Transverse dispersion length	[m]
A	Area of cross-section	[m^2]
$\mathbf{A}$	Matrix of a system of linear equations	[-]
$\mathbf{b}$	Right hand side vector of a system of linear equations	[-]
β	Used as index: Regarded component	[-]
c	Concentration of a tracer	[kg/m^3]
D_m	Molecular diffusion	[m^2/s]
e	An extensive quantity	[-]
ε	Error criterion	[-]
elements	Number of elements	[-]
f	Fractional flow	[-]
$\mathbf{g}$	Gravity acceleration (vector)	[m/s^2]
$\mathbf{h}$	Grid spacing	[m]
i, j, k	Counter variables	[-]
k_r	Relative permeability	[-]
$\mathbf{k}$	Permeability (written bold: tensor)	[m^2]
μ	Dynamic viscosity	[kg/(ms)]
n	Porosity	[-]
nodes	Number of nodes	[-]
ω	Basis function	[-]
$\overline{\boldsymbol{\omega}}^E$	Horizontal vector of basis function for element E	[-]
φ	Weighting function	[-]
$\boldsymbol{\varphi}^E$	Vector of weighting function for element E	[-]
p	Pressure	[Pa]
$p_{c_{a1,a2}}$	Capillary pressure between phases a1 and a2	[Pa]
phases	Number of regarded fluid phases	[-]
$\mathbf{q}$	Darcy velocity (vector)	[m/s]
Q_α	Source/sink rate of phase α	[$m^3/(m^3\,s)$]

Symbol	Quantity	Units
$Q_{\alpha,\beta}$	Source/sink rate of tracer β in phase α	[kg/(m^3 s)]
ρ	Density	[kg/m^3]
r, s, t	Local element coordinates	[-]
S_α	Saturation of phase α	[-]
S_{res}	Residual saturation	[-]
S_{full}	Full saturation	[-]
S_{eff}	Effective saturation	[-]
S	In integral: Surface of the domain	[m^2]
S_0	Soil storativity	[1/Pa]
θ	Time collocation factor	[-]
T	Time	[s]
T	Tortuosity	[-]
u	Regarded variable in FEM	[-]
$\mathbf{V}$	Velocity (vector)	[m/s]
V	Volume of the regarded domain	[m^3]
$\mathbf{x}$	Vector of unknowns of a system of linear equations	[-]
x, y, z	Carthesian coordinates	[m]

1 Introduction

In groundwater systems, fluid fluxes can be present in both saturated and partially saturated zones. In natural systems water and air are most common, in some places additionally crude oil or gas may be found. In areas of current or former human industrial activity the products of this activity may be found, e.g. liquid chemical waste. The waste forms together with the natural fluids a complex, interacting flow system.

In order to develop remediation strategies for contaminated areas, to give proposals about possible groundwater contamination, or to judge the behaviour of groundwater reservoirs, numerical simulators can be a worthwhile tool. One of the scopes of numerical simulation is to increase the understanding of those complex natural systems and eventually to produce predictions for the behaviour of these systems. This task largely depends on the quality and amount of the available data, the quality of the numerical model, the knowledge of the modeller and finally on the available computer power.

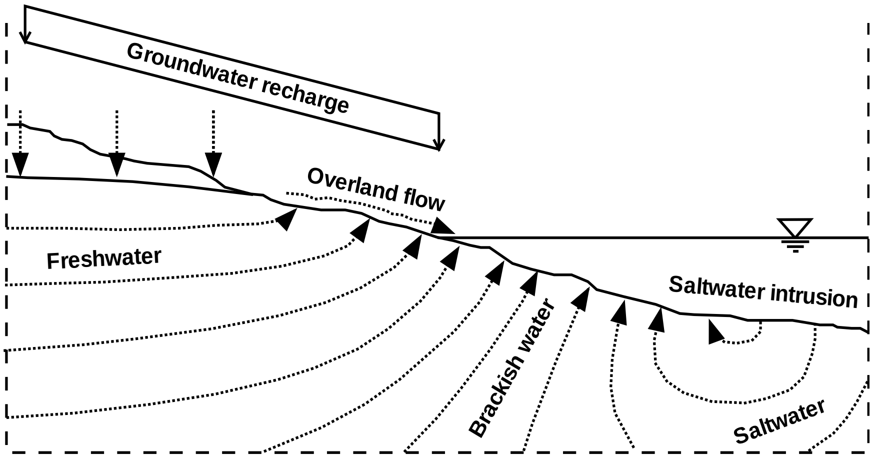


Fig. 1.1: Sketch of a natural groundwater system in a coastal region. The complex interaction between saturated and unsaturated zone, salt- and freshwater must be regarded.

For simple processes like single-phase groundwater flow the numerical simulation already has found its way into everyday engineering practice. For more complex processes, like saltwater intrusion in partially saturated systems (Figure 1.1) or multiphase flow simulation for remediation tasks, the numerical tools are not yet applicable for everyday use. On the one hand this is due to a lack in understanding of the physical processes and on the other because of the state of development of the numerical tools. This work is an attempt to provide an improvement to the latter branch of problems.

1.1 Description of the fractured porous medium

Flow and transport processes in a fractured porous medium depend on both the structure of the fracture network and on the porous medium. Actually, it many cases it will be difficult to differentiate a dense network of fissures from a porous medium. A hint for a differentiation can either be the genesis of the regarded material (fluvial materials are initially porous, but can solidify over time and fracture afterwards) or the bulk porosity (sic!). Furthermore, the differentiation must be related to the scale of observation.

1.1.1 The porous medium

For a porous medium the pore channels and their connection with each other determine the movement of fluids. From a microscopic point of view the material parameters change rapidly from grain to grain. The direct numerical simulation of the Navier-Stokes equation on this scale is possible, though it is not feasible for large scales, because of the numerical effort for this approach. Furthermore, the micro scale geometry of the porous medium is unknown, so that statistical approaches must be used to generate surrogate systems. Anyhow the numerical analysis on this scale can help to understand the physical behaviour of the system and to derive parameterised approaches for larger scales. On a larger scale, the small scale grain based material parameters are averaged over larger, typically geological, units. This scale is commonly used for engineering practice. It is assumed that the averaged physical parameters sufficiently describe the porous medium within each unit. Thus, a *representative elementary volume* (REV) would be suited to describe the system (Range for U_0 in Figure 1.2) on a local scale, while large scale heterogeneities must explicitly be described as different materials. If averaging approaches are used on even larger scales, the internal heterogeneities force the modeller to care for them by using an appropriate physical upscaling model (*Braun*, 2000).

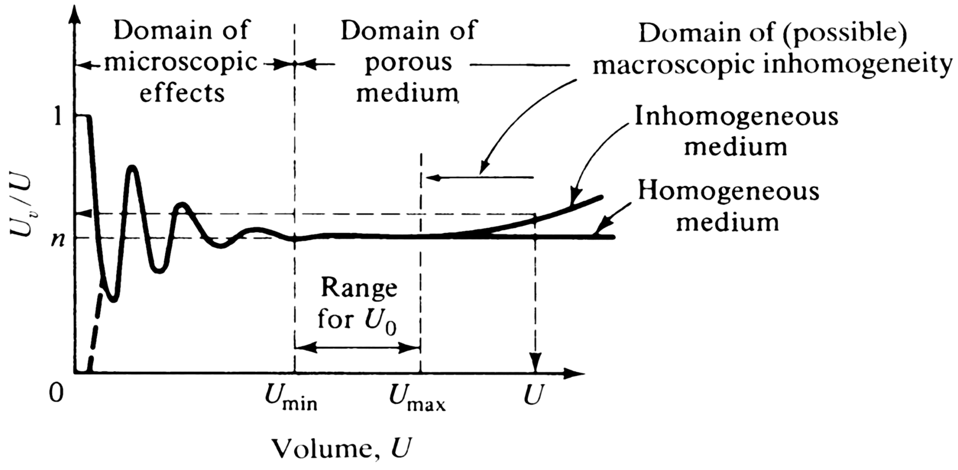


Fig. 1.2: Concept of a representative elementary volume (REV) for a porous medium (*Bear and Verruijt,* 1987). The range for U_0 denotes the scale for which a REV is suitable.

The porosity n of a porous medium on the macro scale is described by the relation between the volumetric amount of pore space in the REV and the volume of the REV:

$$n = \frac{Volume\ of\ \ pore\ space\ in\ REV}{Volume\ of\ \ REV} \tag{1.1}$$

For some special problems it can be useful to define different porosities for different physical processes, e.g. to use a limited porosity for the fluid flow, because it is assumed that the fluid in the smallest fissures is unmoveable due to surface effects. This approach is not followed here for simplicity reasons.

1.1.2 The fractured medium

Fluid flow in fractures or fracture networks can be characterized by various approaches. The simplest one describes the whole system as a continuum, for which bulk parameters are estimated. Though this approach is suitable for simple physical processes, such as steady-state single-phase flow, it fails to capture the behaviour of more complex processes. Therefore, more sophisticated approaches are needed.

In order to describe flow and transport in fractured systems, the system can be treated by describing the system as a fractured, but otherwise homogeneous porous medium. In this approach the fractures and the porous medium are treated as two connected systems (e.g. *Berkowitz et al.*, 1988 and *Wollrath*, 1990).

The double (or even multiple) continua approaches assume that the fracture network can be replaced by an equivalent porous medium on a large scale. This artificial continuum is interwoven with the continuum of the porous medium. Thus, each place in space is a part of two continua. These continua are coupled to each other by flux interchanges between the media (*Barenblatt et al.* ,1960; *Warren and Root,* 1963).

The approaches presented above are the most prominent ones to describe fractured porous media. Each of them has specific advantages and disadvantages. The description of single fractures requires exact knowledge of the spatial distribution of the fractures and additionally of their properties. Even in very well prospected areas it is virtually impossible to determine those parameters for all fractures of all sizes. Thus, only the largest features will be described on the basis of direct measurements, while the typically large amount of smaller scale fractures will either be treated by statistically generated fractures, or as a part of the porous medium.

The multiple continua models suffer from the introduced coupling parameters between the continua. As the number of those parameters increases with the number of continua, the modeller gets many screws to turn in order to tune his model. Though simulations will eventually nicely reproduce the measurements, this does not mean that they catch the behaviour of the natural system for a different set of boundary conditions.

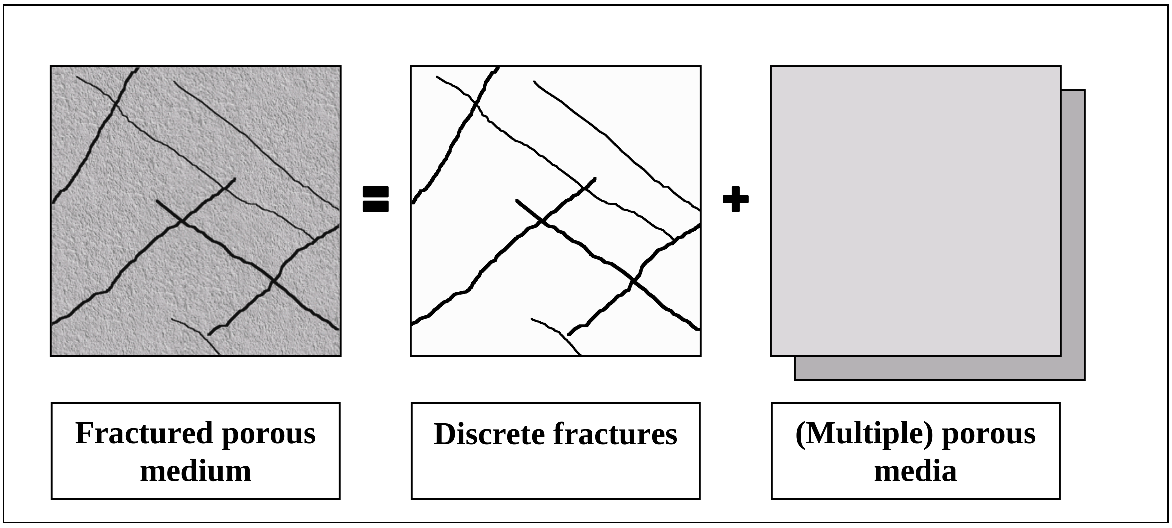


Fig. 1.3: System of multiple overlapping continua. The fractured porous medium is described by a backbone of single fractures and (multiple) homogeneous porous media.

In this work a discrete fracture model with attached matrix was chosen, because it is capable of reproducing known features nicely. Furthermore a discrete fracture model – single matrix continuum can later be extended to a coupled discrete fracture - multiple continua model, i.e. a model that describes known fractures as separate features in space and couples those to multiple continua that describe the properties of the unknown small scale fractures and the porous medium (Figure 1.3).

1.2 Density driven flow

The movement of fluids in porous media, under mixed conditions of applied pressures and buoyancy forces, has been evaluated in many experimental and numerical set-ups. In particular, a considerable number of studies have dealt with density-dependent flow, emphasizing saltwater-freshwater interactions under fully saturated conditions. Classical experimental set-ups include those of *Henry* (1960), as an example of saltwater intrusion into a coastal aquifer, and of *Benard* (1900, in *Holzbecher*, 1998), which treats free convection in a Hele-Shaw cell due to heat. The example by *Benard* was later re-analysed by *Elder* (1967) for applications to porous media. *Diersch* (1984) analysed numerically the behaviour of saltwater beneath a pumping well. More recently, *Schincariol and Schwartz* (1990) expanded analyses to the behaviour of mixed convection in heterogeneous systems in a laboratory set-up. Over the years these and other experiments have formed the basis for the evaluation of numerical models that are incorporated, e.g., in the so-called *HYDROCOIN* (1990) test suite. An overview of the common benchmarks for coupled flow and transport simulations is provided by *Kolditz et al.* (1998a). It should be stressed, however, that all of the studies mentioned above are limited to analysis of density-driven flow in fully saturated systems.

Boufadel et al. (1999) examined numerically the behaviour of water flow below dry lakes with a two-dimensional model. They extended the well-known test case of *Elder* (1967) for application to partially saturated systems, but did not validate their numerical results with laboratory experiments. *Simmons et al.* (1999) compared numerical results for density-driven flow under an evaporating salt lake with experimental results from a Hele-Shaw cell. Though the numerical code would in principle have been able to simulate flow in partially saturated regions, only flow in the fully saturated system was investigated. Recently, *Li et al.* (2000) applied an adaptive multi-grid technique for the numerical solution of density-driven flow in saturated and partially saturated systems. They provide two brief examples that focus on evaluation of their numerical methods, but the systems they considered were not based on or compared to experimental data.

1.3 Multiphase flow

The flow of multiple phases in the subsurface is a question that has been under investigation for a rather long time. The research in this field was initiated by the commercial interest of the oil industry, resulting in a multitude of early pioneering works in this area. In the most recent years the search for secure atomic waste depositories and for remediation strategies for contaminated aquifers raised more attention.

In the context of this work the expression “phase” is used to characterize fluids or solids that do not mix into each other. De facto they can mix in small amounts, but tend to form larger aggregates if the concentration rises above the solubility. On the boundary between two phases the physical properties of the media change with no transition. Thus, a phase is a part of an inhomogeneous system that is bounded by a surface and that is homogeneous within its boundaries.

The “saturation” S_α of a phase α denotes the volumetric amount of the pore space (or of the free aperture of a fracture) that is occupied by the regarded phase in relation to the total pore space:

$$S_\alpha = \frac{\mathit{Volume\ of\ (fluid)\ phase\ \alpha\ in\ REV}}{\mathit{Volume\ of\ \ pore\ space\ in\ REV}} \tag{1.2}$$

Thus, the saturation of a phase is always between zero and one. Furthermore, all phase saturations sum up to unity:

$$\sum_{\alpha=0}^{\mathrm{phases}-1} S_\alpha = 1 \tag{1.3}$$

In an early work *Leverett* (1938) developed the conceptual model of relative permeabilities, based on a set of experiments, in order to quantify how much multiple fluid phases hinder each other in their movement. Later *Leverett* (1940) additionally described the capillary pressures as functions of the fluid saturations.

In the probably first serious attempt to quantify multiphase flow, *Buckley and Leverett* (1941) developed a conceptual model for displacement processes of water and oil under external pressures. They used relative permeabilities to couple the phase saturations with the pressure field. Their physical model for multiphase flow was limited to advective processes, capillary pressures were not regarded.

Later *Snell* (1961) experimentally investigated the behaviour of three-phase flow in sand and derived relative permeability functions. In more recent works *Parker et al.* (1987a) re-evaluated several parametric functions for relative permeabilities and capillary pressures in two- and three-phase systems. In an additional paper *Parker and Lenhard* (1987b) described a set of hysteretic functions.

Flow behaviour of water in the partially saturated zone is often described using the equation derived by *Richards* (1931), which can be regarded as a simplified formulation of the multiphase flow problem (e.g., *Bear*, 1990) adapted to a partially saturated, single-phase system. The most common parametric functions for this approach are based on the work of *van Genuchten* (1980). The parametric functions by *van Genuchten* are used in full multiphase formulations, too.

The flow of multiple phases in fractures and in heterogeneous media was for a long time simplified by assuming equivalent porous media for those systems. As discrete fracture models in single-phase flow gained more attention, the research in multiphase flow was enhanced, too. *Kueper and McWhorter* (1991) investigated the infiltration of DNAPL into fractured clay and rock. *Persoff and Pruess* (1995) worked with artificial fractures and found that linear approaches are not valid for multiphase flow in fractures. In an earlier work by *Pruess and Tsang* (1990), numerical simulations were used to determine the parametric functions for numerous different realizations of rough walled fractures. They found that the permeability - saturation relationship can be described by a power law model. The work by *Murphy and Thomson* (1993) points into the same direction, they simulated two-phase flow in a single rough fracture.

The heterogeneity of the media can lead into discrepancies between the different conceptual models. *Duijn et al.* (1995) described in detail the physics of capillary pressure in heterogeneous porous media and a possible method to circumvent problems on the boundary between different materials. *Helmig and Huber* (1998) compared different numerical methods for two-phase flow in heterogeneous media.

1.4 Objective of this work

Though already a large number of numerical simulation packages are available for different physical processes in the subsurface, most of these simulators are specialized in some aspects of the flow processes described in the prior chapters. During the development of the numerical simulator package RockFlow 3 (*Kolditz et al.*, 1999), it became obvious that it is desirable to have a flow simulator that is *scalable* based on the complexity of the relevant physical processes for a given task. Thus, the goal was to develop a flow module that is suitable for a multitude of different flow processes, i.e.

- single-phase saturated flow,
- single-phase unsaturated flow,
- multiphase flow and
- density driven single- and multiphase flow

of compressible and incompressible fluids in a fractured porous medium.

The developed flow module is *model adaptive*, as it adapts the physical model to the processes that are relevant in the system. This adaptation is separated into two parts. In an a priori adaptation step the modeller decides which general class of problems he wants to regard. Later the model during runtime automatically adapts this preselection to the actual situation in the system.

At the beginning of the implementation work, a grid adaptive transport model formed the basis for the later development (*Barlag*, 1997 and *Schulze-Ruhfus*, 1996). This model had to be enhanced to the context of multiple components tracer transport in multiple fluid phases. Furthermore, a superset simulator for the flow processes presented above was to be developed.

As it is difficult, if not impossible, to verify a complex numerical model, it was compared to a large set of test cases. These are either based on known analytical solutions or, for more complex processes and geometries, on the results of other researchers or at least on plausibility considerations. Clearly, there is a lack of experimental measurements on density-driven flow in partially saturated and coupled saturated/partially-saturated systems, which are suitable for verifying numerical models. Thus an effort was made to address this problem, by combining experimental and numerical investigations of density-driven flow patterns in coupled saturated/partially-saturated systems.

1.5 Implementation of the numerical model

A large part of this work is based on computer implementation. Thus, a few words are necessary to describe the development. As already pointed out in the prior chapter, a grid adaptive numerical model for tracer transport simulations was the condensation core for the further developments. The development of the new RockFlow 3 was based on this software basis and subsequently absorbed the capabilities of the prior RockFlow 2 (*Wollrath*, 1990, *Kröhn*, 1991, *Helmig* 1993) and aTM (*Barlag*, 1997 and *Schulze-Ruhfus*, 1996). Apart from the author of this work, *M. Schulze-Ruhfus* (1996), *U. Jüttner* (1999), O. Kolditz, R. Kaiser and A. Habbar (*Kolditz et al.*, 1998b and *Kolditz et al.*, 1999) worked on the concept and inner kernels of RockFlow 3. Very helpful was the work of *T. Rother* (2001), who was involved in grid generation, and R. Ratke, who coded the very efficient storage and preconditioning scheme for large, sparse systems of linear equations.

Why C?

Due to the requirement of dynamic memory allocation for the grid adaptive procedures, RockFlow 3 was no longer implemented in FORTRAN 77, but in C. It is important to note that at the time of this decision FORTRAN 90, C++, Java etc. were either not yet available, established or approved by a standard. In the long run, this decision has proven to be a good choice. The software concept incorporated in the simulator combines a good set of object-oriented approaches with sufficient freedom for further developments and numerical efficiency. Thus, C has proven to be flexible enough for the numerous demands on the programming language, as it easily transforms into a high level object-oriented language for structures that are not computationally intensive and can come down to high speed for numerical tasks. Or, as D. M. Ritchie, one of the inventors of C, stated it (*Ritchie*, 1993): "*C is quirky, flawed, and an enormous success. While accidents of history surely helped, it evidently satisfied a need for a system implementation language efficient enough to displace assembly language, yet sufficiently abstract and fluent to describe algorithms and interactions in a wide variety of environments*."

2 Mathematical model

The physical processes tackled in this work are described by multiple nonlinear coupled partial differential equations (PDEs). These PDEs describe the conservation and movement of fluids and tracers within them. In the current implementation PDEs for the

- pressure field,
- saturation distributions and
- tracer distributions.

are regarded. The types of the regarded PDEs depend on the choice of the material parameters and on the coupling functions. For many applications the pressure field is mainly parabolic, while the other PDEs are hyperbolic. Due to the extreme nonlinearity of the coupling functions between pressure and saturation fields, the saturation fields can develop shock fronts and are therefore difficult to be treated numerically.

The regarded PDEs are restricted in the following by some constraints. The regarded systems are assumed to be non-deformable and isothermal. Furthermore, phase transitions are not further investigated, though an interface is given to add them via the regarded source and sink terms in the model.

2.1 Governing equations

In general the balance equation of an extensive quantity E within the domain U can be expressed verbally (*Bear,* 1990) by:

$$\left\{\begin{matrix}\text{rate of accumulation}\\ \text{of } E \text{ within } U\end{matrix}\right\} = \left\{\begin{matrix}\text{net influx of } E \text{ into}\\ U \text{ through surface } S\end{matrix}\right\} + \left\{\begin{matrix}\text{net rate of production}\\ \text{of } E \text{ within } U\end{matrix}\right\} \tag{2.1}$$

After applying Gauss’ theorem to the first term of the right hand side and shrinking the volume to an arbitrary point this is equivalent to a PDE for the conservation of an extensive quantity e:

$$\frac{\partial \mathrm{nSe}}{\partial \mathrm{t}} + \mathrm{div}\left(\mathrm{nSe}\mathbf{V} + \mathrm{nS}\mathbf{j}\right) - \mathrm{nS}\tau = 0 \tag{2.2}$$

In this formulation n denotes the porosity, S the saturation, e the extensive quantity per volume unit of the pore space, **V** the fluid’s velocity, **j** diffusive or dispersive fluxes and τ the production rate of the regarded quantity. Here and in the following bold letters denote vectors and matrices while standard letters denote scalars.

2.1.1 Fluid mass conservation

Starting from the general mass conservation equation (2.2), the conservation of mass for a fluid phase α is regarded. For this formulation it is assumed that in the REV the volumetric amount occupied by a fluid phase α is reduced due to porosity n and saturation S_α. The product of porosity, saturation and fluid velocity was replaced by the Darcy velocity $\mathbf{q}_\alpha$. The density ρ was separated from the production term, so that the production Q denotes the production in volume units per volume and time unit.

$$\frac{\partial(nS_\alpha\rho_\alpha)}{\partial t}+\mathrm{div}(\rho_\alpha\mathbf{q}_\alpha)-\rho_\alpha Q_\alpha=0 \tag{2.3}$$

A modified Darcy's law is used to describe the special case of multiphase flow (*Bear* 1972, *Busch et al.* 1993, *Sahimi* 1995). The velocity $\mathbf{q}_\alpha$ of phase α is assumed to be a nonlinear function of the head gradient (expressed by the gradient of the phase pressure p_α and the gravitational force $\rho_\alpha\mathbf{g}$), the permeability tensor **k**, the fluid viscosity μ and the relative permeability k_{r_α}:

$$\mathbf{q}_\alpha=nS_\alpha\mathbf{V}_\alpha=-\frac{k_{r_\alpha}\mathbf{k}}{\mu_\alpha}(\mathrm{grad}\,p_\alpha-\rho_\alpha\mathbf{g}) \tag{2.4}$$

The changes of system variables can induce changes of the relative permeability k_{r_α} (due to changes of all phase saturations or nonlinear behaviour because of non-negligible inertial forces), the fluid viscosity μ_α and fluid density ρ_α (changes of fluid composition, temperature or pressure):

$$k_{r_\alpha}=f_1(S_0,\ldots,S_{\mathrm{phases}-1},v_\alpha)\ ,\quad \mu_\alpha=f_2(c_{\alpha,\beta},T,p_\alpha)\ ,\quad \rho_\alpha=f_3(c_{\alpha,\beta},T,p_\alpha) \tag{2.5}$$

Generally it is difficult to take care of all effects on the relative permeability in a single approach. Therefore it is supposed to treat the effects separately and to multiply all single relative permeabilities to each other, so that a total relative permeability is gained.

$$k_{r_\alpha}=\prod_{\mathrm{effects}}k_{r_\alpha}^{\mathrm{effect}} \tag{2.6}$$

Later this product row will be multiplied with the basic permeability tensor **k** to achieve the final permeability for a fluid phase.

The phase pressures p_α can be described as relative pressures versus a reference pressure. This can either be one of the phase pressures or a weighted pressure in between. Here, one of the phase pressures is chosen as reference pressure, denoted by p_{ref} in the following. The difference between a regarded phase pressure p_α and the reference phase pressure p_{ref} can be evaluated from the corresponding capillary pressures $p_{c_{0,\alpha}}$ and $p_{c_{0,\mathrm{ref}}}$, which depend on the fluid saturations of all phases. The capillary pressures, e.g. $p_{c_{0,\alpha}}$, are evaluated as the pressure difference between the least wetting phase (index 0) and the regarded phase (index α or index ref). Thus, the capillary pressures p_c are positive.

$$p_{c_{0,\dots}} = f\left(S_0,..,S_{\text{phases}-1}\right) \tag{2.7}$$

$$\left.\begin{aligned} p_\alpha &= p_0 - p_{c_{0,\alpha}} \\ p_{\text{ref}} &= p_0 - p_{c_{0,\text{ref}}} \end{aligned}\right\} p_\alpha = p_{\text{ref}} - \left(p_{c_{0,\alpha}} - p_{c_{0,\text{ref}}}\right)$$

After inserting Equation 2.4 into Equation 2.3 and expanding the time derivative of the conservation equation, the final conservation equation, which has to be fulfilled for each fluid phase, is obtained:

$$nS_\alpha \frac{\partial \rho_\alpha}{\partial t} + n\rho_\alpha \frac{\partial S_\alpha}{\partial t} + S_\alpha \rho_\alpha \frac{\partial n}{\partial t} - \operatorname{div}\left(\rho_\alpha \frac{k_{r_\alpha}\mathbf{k}}{\mu_\alpha}\left(\operatorname{grad}\left(p_\alpha\right) - \rho_\alpha \mathbf{g}\right)\right) - \rho_\alpha Q_\alpha = 0 \tag{2.8}$$

2.1.2 Multiphase flow field equations

In the simulation of multiphase flow systems several formulations are feasible. The most common ones are:

- Pressure formulations
- Pressure-saturation formulations
- Saturation formulations

The **pressure formulation** requires a capillary pressure function that is strictly monotic. Only for this case it is possible to replace the saturation in the equations with the reverse capillary pressure function. This is a vast restriction, as the gradient of the capillary pressures function can change rapidly (on material borders) or be very small for some materials (*Peaceman,* 1977). This makes this formulation unusable for the class of problems regarded here.

For the **pressure-saturation formulation** the capillary pressures are used to determine the phase pressures in relation to the pressure of a reference phase. Furthermore the saturation of the phase can be replaced by taking into account the additional condition that all saturations sum up to unity. This type of formulation is quite robust and widely used. It is feasible too for systems with steep gradients of the capillary pressure function. In the following a formulation will be presented which is related to the family of so-called “IMPES” (IMplicit Pressure, Explicit Saturation) formulations. In this name the terms “implicit” and “explicit” do not necessarily denote the time level for which the equations will be solved, but the necessity to solve a system of linear equations or not. Though some authors suggest that this type of formulation is not suitable for compressible fluids (*e.g. Helmig,* 1998b) the formulation presented in the following is suitable for compressible fluids, too.

The **saturation formulation** is based on the idea that a common transport equation for both phases can be set up if both fluids are incompressible (*Peaceman,* 1977). This transport equation contains the "fractional flow function" which is used in the Buckley-Leverett example (Chapter 4.3). Due to the limitation to incompressible fluids, no further investigation was carried out.

2.1.2.1 Pressure Field

With the assumption of non-deformable pore space and by division with the fluid's density the mass conservation equation (2.8) is transformed to a volumetric type:

$$nS_\alpha \frac{1}{\rho_\alpha}\frac{\partial \rho_\alpha}{\partial t} + n\frac{\partial S_\alpha}{\partial t} - \frac{1}{\rho_\alpha}\operatorname{div}\left(\rho_\alpha \frac{k_{r_\alpha}\mathbf{k}}{\mu_\alpha}\left(\operatorname{grad}(p_\alpha) - \rho_\alpha \mathbf{g}\right)\right) - Q_\alpha = 0 \tag{2.9}$$

The linear combination of the conservation equations for all phases results in an equation describing the common pressure field for all phase:

$$\sum_{\alpha=0}^{\text{phases}-1}\left[nS_\alpha \frac{1}{\rho_\alpha}\frac{\partial \rho_\alpha}{\partial t} + n\frac{\partial S_\alpha}{\partial t} - \frac{1}{\rho_\alpha}\operatorname{div}\left(\rho_\alpha \frac{k_{r_\alpha}\mathbf{k}}{\mu_\alpha}\left(\operatorname{grad} p_\alpha - \rho_\alpha \mathbf{g}\right)\right) - Q_\alpha\right] = 0 \tag{2.10}$$

The additional condition that all phase saturations sum up to unity can be used to eliminate the time derivatives of the saturations:

$$\sum_{\alpha=0}^{\text{phases-1}} S_\alpha = 1 \Rightarrow \sum_{\alpha=0}^{\text{phases-1}} \frac{\partial S_\alpha}{\partial t} = \frac{\partial}{\partial t}\sum_{\alpha=0}^{\text{phases-1}} S_\alpha = \frac{\partial}{\partial t}1 = 0 \tag{2.11}$$

This eliminates the saturation derivatives from the balance equation and leads to the final form of the pressure equation, which will be used in the following:

$$\sum_{\alpha=0}^{\text{phases}-1}\left[nS_\alpha \frac{1}{\rho_\alpha}\frac{\partial \rho_\alpha}{\partial t} - \frac{1}{\rho_\alpha}\operatorname{div}\left(\rho_\alpha \frac{k_{r_\alpha}\mathbf{k}}{\mu_\alpha}\left(\operatorname{grad} p_\alpha - \rho_\alpha \mathbf{g}\right)\right) - Q_\alpha\right] = 0 \tag{2.12}$$

This is the governing equation of the combined pressure field of all phases. In contrast to the formulation proposed by *Bear* (1990) this expression does not contain the time derivatives of the saturations. Though this is of lower interest for systems where the densities of the phases are virtually equal (e.g. oil-water systems), it can give an advantage in convergence speed for highly advective systems with highly varying densities, i.e. flow of gases and liquids under high external pressure gradients. For systems that are governed by capillary forces it can be useful to reformulate the pressure field equation in terms of the most wetting fluid.

2.1.2.2 Richards' approximation for partially saturated systems

For soil systems that are connected to the surface it is a common assumption (*Richards,* 1931) that pressure differences in the gaseous phase can be neglected. In this formulation it is assumed that the air pressure is constant within the domain. This leads to a direct dependency between water pressure and water saturation. For water pressure above the reference pressure the pore space is assumed to be fully saturated, whereas in the partially saturated system the water saturation is a function of the water pressure (e.g. the capillary pressure). Starting with Equation 2.9, we obtain for a single phase 0:

$$\frac{n\mathrm{S}_0}{\rho_0}\frac{\partial \rho_0}{\partial t} + n\frac{\partial \mathrm{S}_0}{\partial t} - \frac{1}{\rho_0}\mathrm{div}\left(\rho_0 \frac{k_{r_0}\mathbf{k}}{\mu_0}\left(\mathrm{grad}(p_0) - \rho_0\mathbf{g}\right)\right) - \rho_0 Q_0 = 0 \tag{2.13}$$

As the compressibility of the second phase (i.e. air) and of the matrix are missing in this equation, it is common to introduce the soil storativity S_0 as a lumped parameter for all compressibility effects (i.e. the compressibility of water, residual gas and the solid matrix) and to replace the term that contains temporal derivative of the density by the new term $S_0\mathrm{S}_0\,\partial p_0/\partial t$:

$$S_0\mathrm{S}_0\frac{\partial p_0}{\partial t} + n\frac{\partial \mathrm{S}_0}{\partial t} - \frac{1}{\rho_0}\mathrm{div}\left(\rho_0 \frac{k_{r_0}\mathbf{k}}{\mu_0}\left(\mathrm{grad}(p_0) - \rho_0\mathbf{g}\right)\right) - \rho_0 Q_0 = 0 \tag{2.14}$$

This expression can either be solved with the saturation or with the pressure as the primary variable. It is equivalent to the one proposed by *Celia et al.* (1990). If the saturation is used as primary variable, the equation is only valid in partially saturated areas. With the pressure as primary variable it is valid for all saturations.

Additionally it is possible to replace the temporal derivative of the saturation by the product of the derivative of the capillary pressure function multiplied with the temporal derivative of the pressure. In the derivative of the capillary pressure function the phase pressure p_0 is equivalent to the negative capillary pressure. This is the classic Richards' approximation. It has the disadvantage that it is only mass conservative if finite, and non-zero derivatives of the saturation–capillary pressure function can be determined for all suction pressures in the system.

$$\left(S_0\mathrm{S}_0 + n\frac{\partial \mathrm{S}_0}{\partial p_0}\right)\frac{\partial p_0}{\partial t} - \frac{1}{\rho_0}\mathrm{div}\left(\rho_0 \frac{k_{r_0}\mathbf{k}}{\mu_0}\left(\mathrm{grad}(p_0) - \rho_0\mathbf{g}\right)\right) - Q_0 = 0 \tag{2.15}$$

After evaluating either Equation 2.14 or Equation 2.15, the phase pressures are used to calculate new saturations. This is performed by evaluating the inverse capillary pressure function. This procedure is enclosed into an iterative loop and will be evaluated until sufficient convergence is reached.

Furthermore it is possible to use "mixed" equation types, i.e. to switch the primary variable of the equation based on the current results. *Forsyth et al.* (1995) proposed this so-called Primary Variable Switching Technique (PVST).

2.1.3 Generalized pressure field equation

The pressure field equation for multiphase flow (2.12) and the Richards' equation (2.15) are combined to a common equation, thus forming an equation that describes the superset of all regarded physical processes. In this equation the excessive compressibility effects must be switched on or off in respect to the desired physical behaviour:

$$\sum_{\alpha=0}^{\text{phases-1}}\left[nS_\alpha \frac{1}{\rho_\alpha}\frac{\partial \rho_\alpha}{\partial t} + \left(S_0 S_\alpha + n\frac{\partial S_\alpha}{\partial p_\alpha}\right)\frac{\partial p_\alpha}{\partial t} - \frac{1}{\rho_\alpha}\operatorname{div}\left(\rho_\alpha \frac{k_{r_\alpha}\mathbf{k}}{\mu_\alpha}\left(\operatorname{grad} p_\alpha - \rho_\alpha \mathbf{g}\right)\right) - Q_\alpha \right] = 0 \tag{2.16}$$

In the next equation the time derivatives of the densities are replaced by using the fluid compressibility. This simplification introduces the assumption that changes of the fluid volume are only induced by compressibility. If further reasons for volumetric changes shall be regarded (thermal contraction, volume changes due to solutes), they have to be incorporated here.

$$\sum_{\alpha=0}^{\text{phases-1}}\left[\left(\frac{nS_\alpha}{\rho_\alpha}\frac{\partial \rho_\alpha}{\partial p_\alpha} + S_0 S_\alpha + n\frac{\partial S_\alpha}{\partial p_\alpha}\right)\frac{\partial p_\alpha}{\partial t} - \frac{1}{\rho_\alpha}\operatorname{div}\left(\rho_\alpha \frac{k_{r_\alpha}\mathbf{k}}{\mu_\alpha}\left(\operatorname{grad} p_\alpha - \rho_\alpha \mathbf{g}\right)\right) - Q_\alpha \right] = 0 \tag{2.17}$$

This equation is the "superset" pressure equation, which can be transformed to correspond to either single-phase flow of compressible or incompressible fluids, unsaturated flow (Richards' approach, Equation 2.15) or multiphase flow (Equation 2.12) by choosing the number of phases and appropriate physical parameters. By this means the chosen physical model is adapted to the needs of the modeller. This approach, called *"model adaptivity"* in the following, was developed in order to keep the amount of redundant computer code low, as it can be reused for a large class of problems. Some typical examples for the model adaptive approach are presented in Chapter 3.5.1.

As the superset pressure equation still contains multiple pressures p_α, it is necessary to replace them against relative pressures versus a reference pressure p_{ref} in order to obtain a single primary variable (Equation 2.7). This reference pressure can either be one of the phase pressures or a weighted pressure in between. Here, as reference pressure one of the phase pressures is used:

$$p_\alpha = p_{ref} - \left(p_{c_{0,\alpha}} - p_{c_{0,ref}}\right) \tag{2.18}$$

Thus, now the primary variable is p_{ref}:

$$\begin{aligned}&\sum_{\alpha=0}^{\text{phases-1}}\left[\left(\frac{nS_\alpha}{\rho_\alpha}\frac{\partial\rho_\alpha}{\partial p_\alpha}+S_0S_\alpha+n\frac{\partial S_\alpha}{\partial p_\alpha}\right)\frac{\partial p_{\text{ref}}}{\partial t}\right.\\&\left.-\frac{1}{\rho_\alpha}\operatorname{div}\left(\rho_\alpha\frac{k_{r_\alpha}\mathbf{k}}{\mu_\alpha}\left(\operatorname{grad}p_{\text{ref}}-\rho_\alpha\mathbf{g}\right)\right)\right]=\\&\sum_{\alpha=0}^{\text{phases-1}}\left[\left(\frac{nS_\alpha}{\rho_\alpha}\frac{\partial\rho_\alpha}{\partial p_\alpha}+S_0S_\alpha+n\frac{\partial S_\alpha}{\partial p_\alpha}\right)\frac{\partial\left(p_{c_{0,\alpha}}-p_{c_{0,\text{ref}}}\right)}{\partial t}\right.\\&\left.-\frac{1}{\rho_\alpha}\operatorname{div}\left(\rho_\alpha\frac{k_{r_\alpha}\mathbf{k}}{\mu_\alpha}\operatorname{grad}\left(p_{c_{0,\alpha}}-p_{c_{0,\text{ref}}}\right)\right)+Q_\alpha\right]\end{aligned} \tag{2.19}$$

If heterogeneities are in the system, it is recommended to use the most-wetting phase pressure as reference, because otherwise unphysical pressure jumps on the boundary can occur, if capillary pressure functions with non-zero entry pressures are used.

2.1.4 Saturation field equation for multiphase flow

Starting with the conservation Equation 2.8, similar steps as for the pressure field equation are performed. Again the mass conservation equation is converted to a volumetric one by dividing with the fluid's density. Furthermore, the assumption is introduced that the fluid volume is only influenced by changes of the pressure.

$$nS_\alpha\frac{1}{\rho_\alpha}\frac{\partial\rho_\alpha}{\partial p_\alpha}\frac{\partial p_\alpha}{\partial t}+n\frac{\partial S_\alpha}{\partial t}-\frac{1}{\rho_\alpha}\operatorname{div}\left(\rho_\alpha\frac{k_{r_\alpha}\mathbf{k}}{\mu_\alpha}\left(\operatorname{grad}p_\alpha-\rho_\alpha\mathbf{g}\right)\right)-Q_\alpha=0 \tag{2.20}$$

In this equation only the first two terms contain the primary variable S, thus large parts of the equation can be treated explicitly, because the pressure is already known from the prior calculation of the pressure field.

If the saturation field is not evaluated for the same phase α as the pressure field, the pressure of the regarded phase is evaluated from the capillary pressures and the reference pressure. This is not very effective if capillary forces govern the system. In this case, it is suggested to incorporate the derivative of the capillary pressure function into the equation. If the pressure of phase 0 is used as reference pressure, this leads to the following formulation:

$$nS_\alpha\frac{1}{\rho_\alpha}\frac{\partial\rho_\alpha}{\partial p_\alpha}\frac{\partial p_\alpha}{\partial t}+n\frac{\partial S_\alpha}{\partial t}-\frac{1}{\rho_\alpha}\operatorname{div}\left(\rho_\alpha\frac{k_{r_\alpha}\mathbf{k}}{\mu_\alpha}\left(\operatorname{grad}p_0-\frac{\partial p_{c_{0,\alpha}}}{\partial S_\alpha}\operatorname{grad}S_\alpha-\rho_\alpha\mathbf{g}\right)\right)-Q_\alpha=0 \tag{2.21}$$

This increases the "implicitness" of the scheme and therefore enhances the convergence. It requires that the derivative of the capillary pressure function is definite and sufficiently smooth.

As all phase saturations sum up to unity, for one of the phases the saturation equation must not be solved. Instead Equation 2.11 can be used to determine the saturation. Another approach (*Forsyth et al.*, 1995) is to evaluate the saturation for all phases. As this produces redundant and possibly contradictory information about the saturation fields, afterwards a weighting procedure is used to determine the final saturations. This approach seems to be promising and should be investigated for future developments.

The boundaries between different materials impose special problems. As the phase pressures (and therefore the capillary pressure, if the saturations are above the residual saturations) are constant across the boundary, the saturation will have a jump on the boundary if the capillary pressure functions differ (s. Chapter 3.5.1.5). As the error diminishes by sufficiently refining the grid, this will not be done here. Anyhow it is supposed for future development to assign multiple saturations to a node on the boundary between different materials.

2.1.5 Tracer mass conservation

2.1.5.1 Governing equation for tracer transport

This work mainly focuses on flow processes in the subsurface space. The tracer transport equations are only used to model the impact of a tracer on the fluid properties. Therefore only inert, non-reactive tracers like sodium chloride are regarded. Furthermore phase transitions are not taken into account. The modelling of reactive tracer transport processes is described e.g. by *Habbar* (2001). For the handling of phase transitions further literature (e.g. *Kolditz*, 1996, *Helmig*, 1998 or *Bear*, 1990) should be reviewed. It is easily possible to extend the chosen formulation; phase transitions and reactive exchange terms can be taken into account via the sink/source terms in the equations.

Starting with the conservation equation for extensive quantities (2.2) the conservation of a tracer in a one of the phases can be described. The regarded quantity e is in this case:

$$e = c_{\alpha,\beta} \tag{2.22}$$

In this equation $c_{\alpha,\beta}$ denotes the concentration of tracer β in phase α. The concentration is defined as the tracer mass per volume unit of the fluid. The advective flux term is described by assuming that the Darcy velocity is related to the average absolute velocity $\mathbf{V}_\alpha$ of phase α by saturation and porosity:

$$\mathbf{V}_\alpha = \frac{1}{nS_\alpha}\mathbf{q}_\alpha \tag{2.23}$$

Furthermore diffusive and dispersive fluxes **j** are described by a modified Fick’s law, introducing the diffusion tensor **D**:

$$\mathbf{j} = -\mathbf{D}_{\alpha,\beta} \cdot \operatorname{grad} c_{\alpha,\beta} \tag{2.24}$$

Thus an advection dispersion equation (ADE) is used to model the transport process. The ADE is largely under discussion, as it produces fairly bad results for fracture systems that are treated by single continuum models. Due to the modelling concept with the discrete description of fractures that is used here, this restriction is not necessarily valid, so that in the following the ADE will be used to describe the motion and conservation of a conservative tracer β in phase α:

$$\frac{\partial\left(nS_\alpha c_{\alpha,\beta}\right)}{\partial t}+\operatorname{div}\left(c_{\alpha,\beta}q_\alpha-nS_\alpha D_{\alpha,\beta}\operatorname{grad}c_{\alpha,\beta}\right)-c_{\alpha,\beta}Q_\alpha-Q_{c_{\alpha,\beta}}=0 \tag{2.25}$$

Equation 2.25 can be expanded to:

$$\begin{aligned}&nS_\alpha\frac{\partial c_{\alpha,\beta}}{\partial t}+c_{\alpha,\beta}\frac{\partial\left(nS_\alpha\right)}{\partial t}+c_{\alpha,\beta}\operatorname{div}q_\alpha+\\&q_\alpha\operatorname{grad}c_{\alpha,\beta}-\operatorname{div}\left(nS_\alpha D_{\alpha,\beta}\operatorname{grad}c_{\alpha,\beta}\right)-c_{\alpha,\beta}Q_\alpha-Q_{\alpha,\beta}=0\end{aligned} \tag{2.26}$$

In this expression the second term denotes the changes of the fluid volume in the REV due to compressibility of the fluid, changes of the pore space or the fluid's saturation under multiphase flow conditions. After inserting the fluid continuity equation 2.3 this equation simplifies to:

$$nS_\alpha\frac{\partial c_{\alpha,\beta}}{\partial t}+q_\alpha\operatorname{div}c_{\alpha,\beta}-\operatorname{div}\left(nS_\alpha D_{\alpha,\beta}\operatorname{grad}c_{\alpha,\beta}\right)-Q_{\alpha,\beta}=0 \tag{2.27}$$

This equation has to be fulfilled for each component β in each fluid or solid phase α. The numerical treatment of this equation will not be presented completely in this work, as it was derived in the prior work of *Kröhn* (1991) and implemented in its current form by *Barlag* (1997). In this work the implementation by Barlag was revised and extended to the scope of multiple component tracer transport in multiphase flow. Furthermore, a Method of Characteristics (MOC) scheme was added to improve the performance in fracture networks. Only the differences to the original implementation will be pointed out where it is appropriate.

The diffusive flux scaling factor **D** is a combined diffusion-dispersion tensor according to *Scheidegger* (1961). In the simplest form it is the sum of molecular diffusion D_m (reduced by the tortuosity factor T) and hydro-mechanical dispersion. The dispersion tensor **D** is described as the product of the components of the velocity vector **V** and dispersion length α. In most cases the dispersion length α is split into α_l and α_t, denoting the differences between longitudinal and transversal dispersion. Frequently the longitudinal dispersion coefficient is assumed to be about 5 to 10 times bigger than the transversal dispersion coefficient.

$$\mathbf{D}_{\alpha,\beta} = \begin{bmatrix} TD_{m_{\alpha,\beta}} + \alpha_l|\mathbf{V}_\alpha| & 0 & 0 \\ 0 & TD_{m_{\alpha,\beta}} + \alpha_t|\mathbf{V}_\alpha| & 0 \\ 0 & 0 & TD_{m_{\alpha,\beta}} + \alpha_t|\mathbf{V}_\alpha| \end{bmatrix} \quad (2.28)$$

The dispersion tensor is dependent on the flow directions and velocities on the one hand and on the heterogeneous and anisotropic structure of the porous media on the other. In multiphase flow the dispersion will furthermore be influenced by the fluid saturations. The diffusion is dependent on both the type of solute and fluid. Furthermore, it is a function of the temperature. The dispersion is supposed to describe the influence of small scale heterogeneous and anisotropic soil structures on the tracer distribution, while the (known) large scale structures should be resolved directly. More complex variants of the dispersion tensor may be used for special cases.

For example one could take into account strong density differences in the fluids. In this case the vertical dispersion would be reduced by the stabilizing effect of the stratification, while the horizontal dispersion is not influenced or even increased. *Rinnert* (1983) performed a set of experiments with a horizontal flow field. He found a significant decrease of the transversal dispersion of about 40% for a salt content of 30 g/l. Later *Moser* (1995) showed in another set of experiments that the longitudinal dispersion length is reduced for his vertically arranged soil column, if fluids with varying density are used. Summarizing both works, it must be pointed out that the stratification modifies neither transversal nor longitudinal dispersion, but the vertical dispersion. This decrease is a function of Peclet and Rayleigh numbers (after *Rinnert,* 1983):

$$\alpha_z / \alpha = f\left(\frac{|\mathbf{V}| \cdot d}{D_m}, \frac{\Delta\rho \cdot d^3 \cdot g}{\eta \cdot \alpha \cdot |\mathbf{V}|} \right) \leq 1 \quad (2.29)$$

In this equation d denotes the characteristic diameter of the regarded heterogeneities. In the work by *Rinnert* this was set to the grain diameter, as he used a homogeneous filling.

Furthermore the tortuosity and the dispersion lengths can be functions of the saturation. Some authors (in *Bear* 1990) suppose to relate the tortuosity to the saturation by:

$$T = \frac{(n \cdot S)^{10/3}}{n^2} = n^{4/3} \cdot S^{10/3} \quad (2.30)$$

2.1.5.2 Operator splitting technique

The ADE can change its classification from a parabolic type for diffusive problems to a hyperbolic type for advective problems. The Peclet number Pe characterizes the ratio of the advective tracer transport against the diffusive transport:

$$\mathrm{Pe} = \frac{|\mathbf{V}| \cdot \mathrm{L}}{|\mathbf{D}|} \tag{2.31}$$

Within this equation **V** denotes the fluid velocity, L a typical length (i.e. the grid spacing in a numerical simulation) and **D** the diffusion/dispersion coefficient. For Pe << 1 the ADE is mainly diffusive, for Pe >> 1 mainly advective. As the performance and suitability of different numerical schemes largely depend on the Peclet number, it is desirable to use a scheme that fits with the expected range of Peclet numbers.

Especially in fracture networks the Peclet numbers inside the fractures can be very high. If the fractures network is not treated by a “lumped” model with a single ADE for the whole domain, but by a discrete approach, the numerical scheme must deal with this problem.

As a solution for these objectives a split of the ADE into a more diffusive part that is treated by an Eulerian approach (named after *Leonhard Euler*, 1707 - 1783) and an advective one that is treated by a Lagrangian approach (named after *Joseph-Louis Lagrange*, 1736 - 1813) is suitable. In Eulerian schemes the equations are formulated from the point of view of a fixed observer, while the fluid moves in front of him. In a Lagrangian formulation the observer moves within the fluid. Obviously this is a superior way to examine fast moving fluids (Figure 2.1).

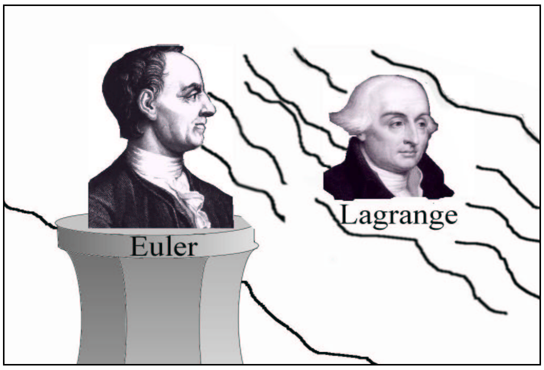


Fig. 2.1: Lagrangian and Eulerian points of view while investigating moving fluids.

The operator splitting offers the possibility to treat parts of the PDE by different numerical schemes. The time derivative of the concentration in Equation 2.27 is discretized (Chapter 3.3). The indices of c where dropped here for simplicity reasons. The derivative can be separated into two parts:

$$\frac{\partial c}{\partial t} \approx \frac{c^{t+\Delta t} - c^{t}}{\Delta t} = \frac{c^{t+\Delta t} - c^{t^{\sim}}}{\Delta t} + \frac{c^{t^{\sim}} - c^{t}}{\Delta t} \tag{2.32}$$

This leads to the split form of the conservation equation (after *Hinkelmann*, 1997):

$$
\begin{aligned}
\frac{c^{t+\Delta t} - c^{\tilde{t}}}{\Delta t} &= -\frac{1}{nS_\alpha} \mathbf{q}_\alpha \cdot \operatorname{grad} c \\
\frac{c^{\tilde{t}} - c^{t}}{\Delta t} &= \frac{1}{nS_\alpha} \operatorname{div}\left(nS_\alpha \mathbf{D} \operatorname{grad} c\right) + \frac{1}{nS_\alpha} Q_\beta
\end{aligned}
\tag{2.33}
$$

These equations can be solved separately by an adequate scheme, e.g. by a Lagrangian characteristics scheme for the advective part and an Eulerian finite-element approach for the diffusive part.

2.2 Equations of state

A set of equations of state is necessary to couple the conservation equations to each other and thus close the set of equations. The equations of state can be sorted into several groups. One group covers the interaction between fluid and porous medium. Often these functions differ in space, because the porous medium changes. A second group of functions describes the impact of the primary variables on the physical properties of the fluids (i.e. functions describing density, viscosity, diffusion etc.). A third group would be the combination of the two, describing the influence of changes of the primary variables on the equations of state of the porous medium. The first group of functions is mainly related to geophysical sciences, while the second one belongs to the research area of chemical engineering. The combination of different influences on the parameters opens a broad field for functional dependencies, which depend on the regarded species.

It is impossible to define “general” functions that are numerically feasible and cover all possible physical processes. Therefore, the software concept incorporated in RockFlow uses an open, object oriented interface, which makes it easy to incorporate any kind of function. Due to this ease in enhancing the physical dependencies, only a limited subset of the possible functions was implemented and will be presented here.

2.2.1 Permeabilities

The general idea of Darcy’s law (i.e. a linear relation between pressure gradient and velocity) is only valid for a very limited range of conditions. The system

- must be fully saturated,
- apertures must be big enough (otherwise Knudsen flow for gases and surface effects for liquids dominate the system) and
- velocities must be in a certain range.

Various approaches are available to describe the permeability as a function of some properties of the porous medium. The most famous one is the *Kozeny-Carman* equation

(*Bear*, 1979). It is based on the porosity n, the specific inner surface M_s of the porous medium and a scaling factor C_0, for which Carman suggested 1/5:

$$k = C_0 \frac{n^3}{(1-n)^2 M_S^2} \tag{2.34}$$

If fractures are represented by a „parallel plate“ model with low local variability of the aperture b, the local permeability k can be approximated from so-called *“cubic law“* (*Witherspoon et al.*, 1980). The cubic law originally described an approach to evaluate the transmissivity of a fracture. Thus, it is divided by the aperture b to obtain the permeability:

$$k = \frac{b^2}{12} \tag{2.35}$$

This approximation is only valid if the fracture roughness is low compared to the open aperture of the fracture.

To extend Darcy’s law for multiphase systems multiple approaches are available in the literature (*Helmig*, 1998b, gives a good overview of common approaches) to describe the impact of all phases on the total flow field. These approaches can either be rather simple ones like the Brooks-Corey approach (*Brooks and Corey*, 1966), for which the relative permeability for each phase is directly related to the corresponding phase saturation, or rather complicated for approaches with interaction of all saturations on the relative permeability of the other phases.

2.2.2 Relative permeability-velocity function

The linear dependency between pressure head and fluid velocity is not valid over the complete range of Reynold’s numbers Re:

$$\mathrm{Re}_\alpha = \frac{|\mathbf{v}_\alpha| b \rho_\alpha}{\mu_\alpha} \tag{2.36}$$

In this equation b denotes a characteristic length, for example a pore diameter or the fracture aperture. For very low Reynold’s numbers (Re<<1), and especially for very thin fractures or fissures, the molecular bounds between water and rock matrix can significantly hinder the water from moving. For slightly larger Reynold’s numbers the linear approach of Darcy’s law is valid. For Reynold’s numbers between 10 and approximately 100 the flow field is still laminar, but inertial forces in the Navier-Stokes equation should no longer be neglected.

The classical approach to describe this behaviour was developed by *Forchheimer* (1914). It relates the pressure drop in the porous medium to a first and a second order function of the velocity:

$$\frac{\mu}{k}\mathbf{q} + \beta\rho\mathbf{q}^2 = -(\mathrm{grad}\ p - \rho\mathbf{g}) \tag{2.37}$$

In this expression the first term rises in importance for low velocities. It characterizes the linear dependency between pressure drop and velocity, which is typical for laminar flow. The second term is of importance for high velocities and resembles the pressure loss due to inertial forces. The variable β is used as a scaling factor for this effect.

The nonlinear flow behaviour can be accounted for by an element in the product row of relative permeability effects (Equations 2.6). In this case the "turbulent" relative permeability $k_{r,turb.}<1$ is used to scale the permeability. It should be pointed out that turbulence does only occur under very rare conditions. The index "turb" denotes nonlinear flow because of inertial forces and, additionally, the rare case of real turbulence:

$$k_{r,turb.} = f(\mathbf{q}) \tag{2.38}$$

The main problem is that the regarded systems are locally heterogeneous, thus the flow field is turbulent in parts while it is still laminar in others. A smooth transition from the linear to the nonlinear regime is needed to describe the averaged behaviour. Some restrictions can be formulated for the relative permeability function for nonlinear flow:

- $k_{r,turb.} = 1$ for small velocities (laminar flow)
- $k_{r,turb.} = \alpha / |\mathbf{q}|$ for high velocities (turbulent flow)
- $k_{r,turb.}$ must be strictly monotic dropping with rising velocities

The first constraint leads to the classic Darcy approach with a linear dependency between pressure drop and velocity. The second constraint describes the quadratic pressure drop for turbulent flow. A smooth transition between the two approaches would be useful, because the REV will not completely switch its behaviour at a certain velocity. For multiphase flow it might be useful to enhance this approach by relating the function to the fluid saturation.

2.2.3 Relative permeability-saturation functions

If more than one fluid phase exists in the pore space of a porous medium those fluid phases will interact with each other. Even if the drag on the fluid surfaces is neglected, this will result in a reduced effective permeability for the regarded phase. This disturbance is characterized by the relative permeability k_{r_α}. Several authors worked on the determination of the relative permeability at the micro scale. Models to analyse this behaviour are e.g. network models, which describe the underground as a network of nodes and interconnecting capillary tubes (*Parker et al.* 1989). Those investigations confirmed the well known macro scale relations between relative permeability and saturation of a phase in the medium. These functions can be hysteretic, i.e. the behaviour for draining and imbibition is differing. Furthermore they differ between the fluid with a higher affinity for the solid matrix (“wetting fluid”) and other ones with a lower affinity (“non-wetting fluids”).

A summary of different models for the relative permeability-saturation function is given e.g. by *Miller et al.* (1998) or *Helmig* (1998b). It was assumed that the relative permeability is a scalar, though it is supposed to be a tensor. This simplification is made because of the lack of available functions to describe this tensor, and, even more so, the lack of experimental data. Furthermore, the relative permeability functions for fractures can be influenced by the fluid velocity (*Fourar et al.*, 1993). This is the case, if the local equilibrium is not governed by capillary pressures, but by external pressures and inertial forces.

Here a set of scalar functions is presented, these are currently implemented in the simulator. Due to the object oriented concept and the open interfaces of RockFlow, the simulator can easily be enhanced to other functions is this necessary.

In the following the effective saturation S_{eff_α} will be used to simplify the equations:

$$S_{eff_\alpha} = \frac{S_\alpha - S_{res_\alpha}}{S_{full_\alpha} - S_{res_\alpha}} \quad ; 0 \le S_{eff_\alpha} \le 1 \tag{2.39}$$

Here, S_{res_α} denotes the residual saturation and S_{full_α} the saturation, for which the maximal relative permeability is reached.

User defined curve

This is a very generic approach. The user can specify curves simply by entering a set of points along the desired curve. By this means any type of curve can easily be specified, i.e. experimentally gathered data can directly be used (after some smoothing) without of the necessity to fit it to a special function.

Perfectly mobile

All phases show perfect mobility and do not interact with each other. This means that k_{r_α} is equal to unity for all phases:

$$k_{r_\alpha} = 1 \tag{2.40}$$

Linear function

In this approach the permeabilities rise linearly between the residual saturation and the maximal saturation. If the Richards’ approach is used to compute the seepage in large scale systems, the linear approach is suitable to take into account the reduced flow through partially flooded elements. In two-phase systems the full saturation is determined by the residual saturation of the other phase:

$$k_{r_\alpha} = S_{\mathrm{eff}\,\alpha} \tag{2.41}$$

Exponential function

An exponential dependency with a variable exponent is frequently used and a “classic” approach. An exponent between two and five is common for the wetting phase, while the non-wetting phase usually is described by a lower exponent.

$$k_{r_\alpha} = \left(S_{\mathrm{eff}\,\alpha}\right)^a \tag{2.42}$$

Books-Corey functions

Brooks and Corey (1964) distinguished the non-wetting phase permeability $k_{r_{nw}}$ from the wetting phase permeability k_{r_w}:

$$\begin{aligned} k_{r_{nw}} &= \left(1 - S_{\mathrm{eff}\,nw}\right)^2 \left(1 - \left(S_{\mathrm{eff}\,nw}\right)^{\frac{2+\lambda}{\lambda}}\right) \\ k_{r_w} &= \left(S_{\mathrm{eff}\,w}\right)^{\frac{2+3\lambda}{\lambda}} \end{aligned} \tag{2.43}$$

Van Genuchten function

This approach was derived from the one proposed by *Van Genuchten* (1980). The main field for this approach are infiltration problems of water into the soil. In the original work only the relative permeability for the wetting phase was addressed. Thus, for the non-wetting phase another approach must be chosen. The wetting phase relative permeability is calculated from:

$$k_{r_\alpha} = S_{\mathrm{eff}_\alpha}{}^2 \left(1 - \left(1 - S_{\mathrm{eff}_\alpha}{}^{\frac{1}{m}}\right)^m\right) \tag{2.44}$$

Haverkamp function

The approach is a slightly modified version of the formula proposed by *Haverkamp* (1977). As the Van Genuchten approach, it is mainly used for infiltration problems of water into the soil and no information about the non-wetting phase is given:

$$k_{r_\alpha} = \frac{c}{c + \left(a \dfrac{S_\alpha - S_{\mathrm{res}_\alpha} - S_{\mathrm{full}_\alpha}}{S_\alpha}\right)^b} \tag{2.45}$$

In the original publication the following form factors were used:

$$a = 1.611 \cdot 10^6; \qquad b = 1.197; \quad c = 1.175 \cdot 10^6$$

2.2.4 Capillary pressure functions

Another important characteristic of multiphase processes is the existence of interfaces between the phases at the micro scale. Surface tensions on the interfaces between the phases induce the movement of the fluids until a local equilibrium is reached. This local equilibrium depends among other influences on the fluids themselves, the solutes inside them, the surrounding porous material and the temperature. The different affinity of wetting and non-wetting fluids for the matrix of the porous medium can be observed on the micro scale as a difference in the contact angles between fluids and solid.

On a medium scale the soil can be interpreted as a network of capillary tubes of different diameters. If a certain amount of a wetting fluid is infiltrated into this network, the fluid will settle finally in equilibrium. The smallest pores are filled first (due to the high suction pressure), while larger pores are filled later. Obviously the system will not show the same behaviour during imbibition and draining, i.e. the behaviour is hysteretic (as it is in nature). This type of model was successfully used for two-phase flow (*Lenormand et al.*, 1988) and three-phase flow (*Soll and Celia*, 1993).

For larger scales heuristic approaches are widely used. They can be interpreted as functions that relate saturations and capillary pressures. These functions can either be gained from experimental data or by volume averaging the results of the network models mentioned previously.

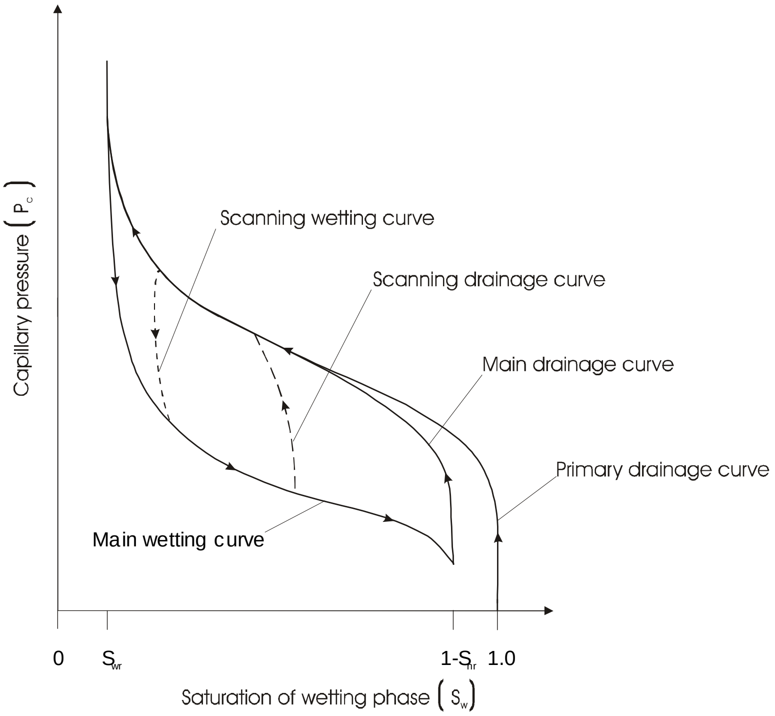


Fig. 2.2: General description of a capillary pressure-saturation function (*Redeker*, 1999).

Figure 2.2 shows a typical hysteric capillary pressure–saturation function. The hysteresis has several reasons:

- Differences in the contact angle between advancing and withdrawing liquid.
- The “ink-bottle” effect of pores of different shape.
- Reaction of the porous material on the fluid (e.g. swelling of clay particles).

Several models for the capillary pressure - saturation relationship are available. A summary of these models is given in *Miller et al.* (1998). The functions used in this work will be outlined in the following. The inverse function has been added, as it is useful for the Richards’ approach. If the inverse function is not known, it is possible to use a Newton or a regula-falsi scheme to determine the inverse value, though this is less effective

User defined curve

The most flexible approach is the possibility to define a curve based on several single points. The inverse function can easily be determined numerically:

$$p_{c_\alpha} = f\left(S_\alpha\right) \tag{2.46}$$

Linear function

The capillary function is assumed to be a linear function of the saturation. This approach is useful for the Richards’ approach, as it can be used to specify artificial capillary pressures if the saturation changes are too sharp and numerical stability problems are encountered. Furthermore for systems with large vertical extent of the unsaturated zone this approach is useful to specify curves, which fulfil the criteria for the Richards’ approach:

$$p_{c_\alpha} = a \cdot \left(1 - S_{\mathrm{eff}\,\alpha}\right) \Leftrightarrow S_{\mathrm{eff}\,\alpha} = 1 - \frac{p_{c_\alpha}}{a} \tag{2.47}$$

Exponential function

The capillary pressure rises with an exponent 1/b in the area between minimal and maximal saturation. It is scaled by the multiplier a:

$$p_{c_\alpha} = a \cdot \left(1 - S_{\mathrm{eff}\,\alpha}\right)^b \Leftrightarrow S_{\mathrm{eff}\,\alpha} = 1 - \left(\frac{p_{c_\alpha}}{a}\right)^{\frac{1}{b}} \tag{2.48}$$

Van Genuchten function

The approach by *Van Genuchten* (1980) has developed to a standard for infiltration problems. It has the form:

$$p_{c_\alpha} = \frac{\rho g}{a}\left(S_{\mathrm{eff}\,\alpha}{}^{-\frac{1}{m}} - 1\right)^{\frac{1}{n}} \Leftrightarrow S_{\mathrm{eff}\,\alpha} = \left(\left(\frac{a \cdot p_{c_\alpha}}{\rho g}\right)^{n} + 1\right)^{-m} \tag{2.49}$$

In this approach a, m, n are form factors of the capillary pressure function. A large amount of literature is available with form factors for multiple soils.

Haverkamp function

The approach by *Haverkamp* (1977) is mainly suited for infiltration problems, too. It was originally formulated for the water retention curve, re-evaluation for capillary pressures gives:

$$p_{c_\alpha} = 0.01\rho g\left(\frac{a}{S_{\mathrm{eff}\,\alpha}} - a\right)^{\frac{1}{b}} \le c \Leftrightarrow \frac{a}{\left(\frac{p_{c_\alpha}}{0.01\rho g}\right)^{b} + a} = S_{\mathrm{eff}\,\alpha} \tag{2.50}$$

In the original publication the following form factors were used:

$a = 1.611 \cdot 10^6$; $b = 3.96$; $c = 1 \cdot 10^5$

Brooks-Corey function

The Brooks-Corey model (*Brooks and Corey,* 1966) uses a more physical approach to describe the properties of the porous medium. Their approach uses the bubbling (or threshold) pressure p_b (the pressure at which the non-wetting phase starts to enter the system), the residual saturation S_{res} and the pore size distribution factor λ to characterize the capillary pressures.

$$p_{c_\alpha} = p_b \cdot \left(\frac{S_\alpha - S_{\mathrm{res}\alpha}}{1 - S_{\mathrm{res}\alpha}}\right)^{-\frac{1}{\lambda}} \Leftrightarrow S_\alpha = S_{\mathrm{res}\alpha} + \left(1 - S_{\mathrm{res}\alpha}\right)\left(\frac{p_b}{p_{c_\alpha}}\right)^{\lambda} \tag{2.51}$$

In contrast to the van-Genuchten approach, the capillary pressure is not zero for a saturation equal to unity. This is the so-called threshold pressure, which must be surpassed by the non-wetting phase before it can enter the porous medium. If it is assumed that the soil is a network of tubes, this is equal to the assumption that the distribution of tube diameters has a sharp upper boundary and the largest tube has a capillary pressure equal to the entry pressure. In contrast, the van-Genuchten approach assumes that the pore size distribution has no sharp upper bound. For technical systems and very well sorted natural porous media, the Brooks-

Corey approach is superior, while the van-Genuchten approach is better suited for natural porous media with a wide range of pore diameters.

2.2.5 Fluid and tracer properties

The properties of the regarded fluids and tracers are depending on multiple parameters. Pressure, temperature, mixing ratio and the choice of materials themselves for two or more components opens an overabundance of possible interactions. The evaluation of those parameters belongs to the field of chemical engineering and only some aspects will be presented in the following. It must be pointed out that the object oriented software concept of RockFlow (*Kolditz et al.*, 1998b) easily enables the incorporation of any other equation of state if this is necessary.

2.2.5.1 Fluid Density

The density of a fluid is influenced by multiple factors, i.e. pressure, temperature and solved solutes. For gases the density can be described by the Boyle-Marriot formula. For ideal gases and isothermal conditions it can be simplified to:

$$\rho = \frac{\partial \rho}{\partial p} p \tag{2.52}$$

The pressure only to a small degree influences the density of liquids, as liquids are nearly incompressible. For liquids the dependency between density, temperature and pressure is not necessarily linear. A good example is water, which has a density maximum at four degrees centigrade. For temperatures that are sufficiently high, a linear approximation is accurate enough to estimate the temperature dependency in a small range from the reference temperature:

$$\rho = \rho_{\mathrm{ref.}} + \frac{\partial \rho}{\partial p} p + \frac{\partial \rho}{\partial T} T \tag{2.53}$$

The effect of solutes on the liquid density can be linearized for low solute concentrations, as they mix with the liquid with no volumetric changes. Thus the prior equation is extended to:

$$\rho = \rho_{\mathrm{ref.}} + \frac{\partial \rho}{\partial p} p + \frac{\partial \rho}{\partial T} T + c_{\beta} \tag{2.54}$$

For specific liquids, i.e. water, more sophisticated approaches are available to determine the density for a wide range of pressures, temperatures and salinities. For example *Stuyfzand* (1989, in *Holzbecher* 1998) proposes for saline water:

$$\rho_{\mathrm{water}} = 1000 \cdot \left(1 + 0.805 \cdot c_m - 6.5 \cdot 10^{-6} \left(T - 4 + 220 \cdot c_m\right)^2\right) \tag{2.55}$$

As already pointed out, multiple other functions are available to describe the density of a liquid. For seawater intrusion problems the formula by the *UNESCO* (1981) should be investigated further. Due to the open interface concept of the developed program, any function can easily be added to the system.

2.2.5.2 Fluid Viscosity

The viscosity of liquids changes largely with temperature, but is quite invariant against pressure changes. It is interesting to note, that this is not only valid for liquids, but also for gases. For water *Pawlowski* (1991) proposes the following function for the temperature dependency:

$$\mu_{\mathrm{water}} = 10^{-3} \cdot \left(1 + 0.01551 \cdot \left(T - 293.15\right)\right)^{-1.572} \tag{2.56}$$

Another approach is given by *Reid et al.* (1988). It is a more general approach and is approximately valid for all liquids, if the temperatures are far enough from the freezing and boiling temperature. In the following equation A and B are fitting parameters:

$$\mu = e^{A + \frac{B}{T}} \tag{2.57}$$

Furthermore the viscosity of liquids can be influenced by solutes. Small salt contents have a rather small effect on the viscosity, i.e. seawater with 35 kg salt per cubic metre increases the viscosity by less than 10%. For brines with high salt contents of up to 300 kg/m^3, changes in viscosity must be taken into account as the viscosity is increased by more than 100% (*Spitz*, 1985).

2.2.5.3 Tracer diffusion

The molecular diffusion coefficient D_m determines the mass of tracer that diffuses through an area of one square meter, if the concentration gradient (with the concentration being defined as mass per volume) is equal to unity (*DVWK*, 1985).

In a porous medium the diffusion is hindered by the structure of the medium. It is possible to take this into account by multiplying the molecular diffusion D_m (which is a property of the fluid and the solute) with the tortuosity T of the porous medium. Furthermore very small pore diameters can influence the diffusion, as the collisions of molecules between each other are superposed by the collisions with the wall (Knudsen diffusion).

For many ions (like sodium chloride) the diffusion coefficient in water can be expected to be a range between $1 \cdot 10^{-9}$ m^2/s and $2 \cdot 10^{-9}$ m^2/s at 20°C. The diffusion is highly dependent upon temperature. For water self diffusion rises approximately from $1 \cdot 10^{-9}$ m^2/s to $5 \cdot 10^{-9}$ m^2/s between freezing and boiling temperature (*Reid et al.*, 1988). The pressure or the concentration only to a small degree influence the diffusion coefficient of a liquid, so that those influences can be neglected.

3 Numerical approximation

The basic idea is to replace the rather complex to find solution for the mathematically formulated physical problem by a simpler one, i.e. solving a system of linear equations. In this work the physical problem is described by the conservation equations which were evaluated in Chapter 2.1.

For the temporal discretization of those equations among others the finite element method (FEM) as well as the finite differences method (FDM) are suitable. Due to the simpler general structure, the FDM has been chosen for the temporal discretization (Chapter 3.3). For the spatial discretization a large amount of numerical methods is available. The most prominent ones are

- the finite differences method (FDM),
- the finite volume method (FVM),
- the finite element method (FEM),

and a lot of hybrid methods, which are in between. For a general overview it is recommended to review a textbook (e.g. *Hirsch*, 1988) on computational fluid dynamics (CFD). All of the methods presented above have strong and weak points. The most important points will be outlined in the following.

The FDM has been largely successful in the numerical simulation due to a number of good reasons. First of all it is straightforward to understand and to implement in computer codes. The spatial derivatives of variables are approximated by polynomials, which are evaluated on rectangular grids. It is possible to use coordinate transformed formulations, so that the actually modelled domain is no longer rectangular. In that case (useful for example for rivers) the grid would be rectangular after applying a backward coordinate transformation on the grid. The FDM results in systems of linear equations, which are filled to a small degree (“sparse”) and show a banded structure. The main advantage of the FDM is the ease in implementation on the one hand and the speed in execution on the other. It must be pointed out that this is only true if the computational effort *per number of nodes* is regarded. This type of evaluation of the efficiency of numerical methods is misleading, as the *quality of the results per number of nodes* may differ between the methods. Thus the *achieved quality per CPU time* should be compared. Actually, for cases with complex geometries (e.g. fracture systems or heterogeneous soils with thin layers) the FDM may be inefficient, because the necessity to place a high number of nodes in areas with changing material properties forces the placement of other nodes due to the restriction to rectangular grids. This can be partially circumvented by using “zoom” methods or floating refined grids. Furthermore the FDM in standard form has the disadvantage of not being mass conservative. Thus the modeller is forced to use higher order interpolation schemes to keep mass losses bounded and phase speed errors low.

The FEM and the FVM are both based on elements of not necessarily rectangular shape and are therefore well fitted to approximate arbitrarily shaped geometries. Furthermore they are globally mass conservative and produce, like the FDM, sparse systems of linear equations. Traditionally the FEM has been largely used in modelling solid mechanics, while the FVM is increasingly popular in fluid mechanics. The FVM is quite often supposed to be superior because of the so-called “local conservation”. Actually the local conservation only describes the ease in understanding (and implementing) the method, not a numerical quality. *Gresho and Sani* (1998) demonstrate that the FVM in fact is comparable to a lower order FEM, and shows lower accuracy in modelling the phase speed. The following table is supposed to recapitulate the statements made above:

Table 3.1: Comparison of important features of the regarded numerical schemes.

	FDM	FEM	FVM
Efficiency of spatial discretization	-	+	+
Efficiency per node	+	-	-
Global mass conservation	o	+	+
Implementation	+	-	o

For hydro-geological systems very complex geological structures must be reproduced. This is important for layered, non-fractured aquifer system, but even more important for fracture systems. Thus the FDM has not been chosen for the spatial discretization. Both the FEM and the FVM or their hybrid forms are suitable for the spatial discretization of geological systems.

In general the FEM produces good results for mechanical problems (i.e. the pressure field equation), while the FVM shows nice behaviour for transport processes. As it is planned to incorporate mechanical problems into the numerical code and a variety of methods are available to enhance the performance for transport simulations, the FEM has been chosen for the further steps. For future development it should be considered to use the FVM to describe transport processes.

3.1 Discretization with the finite element method

3.1.1 Spatial discretization

The continuous solution for the whole domain is replaced by a number of small, interconnected sub-domains called finite elements. The temporal discretization is treated by Finite Differences (Chapter 3.3) and the nonlinearities are resolved by an external iteration loop (Chapter 3.4).

The process of partitioning the domain into single elements is called mesh generation. It depends on the types of available finite elements and on the geometry of the underlying geophysical system. In this work one-dimensional two node line elements, two-dimensional four node elements and three-dimensional eight node elements (Figure 3.1) are used. These elements can freely be interconnected in 3D-space, in order to represent preferential flow paths, fractures, and the solid matrix. If this coupling is used, it is important that the volumes of the coupled elements are suitable to describe the relevant processes, i.e. if 2D elements (representing a very thin fracture) are connected to large 3D elements (representing the rock matrix), the mass storage of the 3D elements would hinder the movement in the fractures in an unphysical manner. In this case it is necessary to use a graded layer of matrix elements or to implement special transition conditions between fracture and matrix.

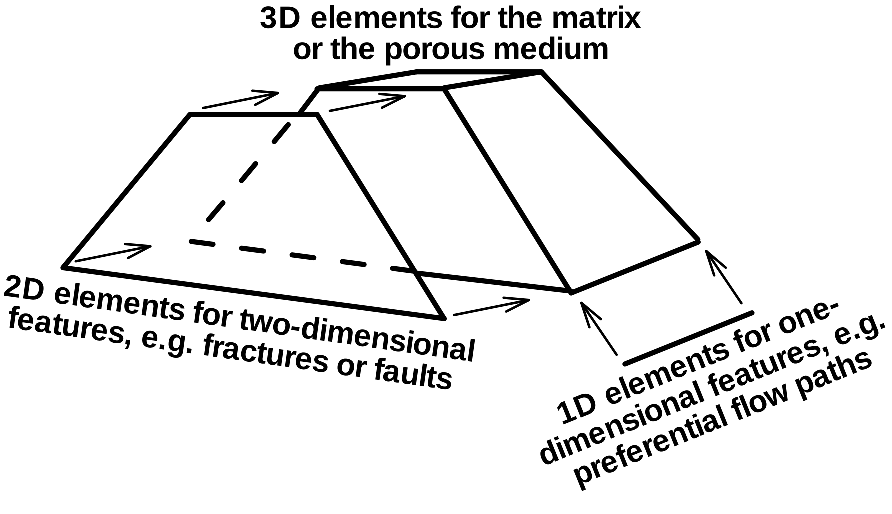


Fig. 3.1: Coupling of different element types in three-dimensional space. Line elements for preferential flow paths, planar elements for fractures and hexahedral elements for the matrix can freely be coupled to represent complex geological systems.

The geometry of the mesh should be adapted to the simulated physical processes. For example, it is possible to use streamline oriented meshes, if transverse numerical dispersion is to be reduced (Figure 3.2).

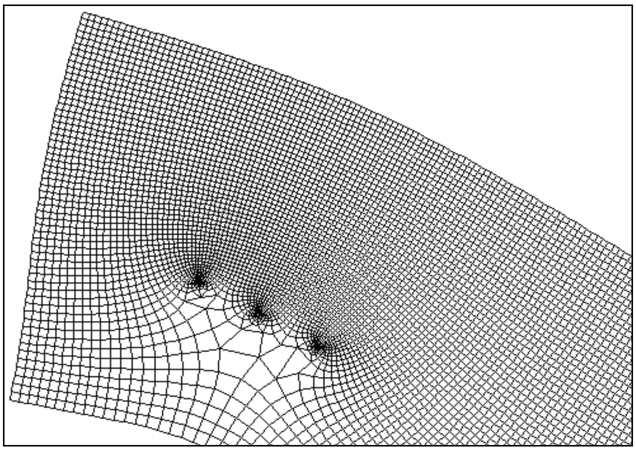


Fig. 3.2: Streamline adapted finite element mesh for a gallery of three wells (*Thorenz*, 1995).

Furthermore, the mesh should be sufficiently fine in regions where large changes of the regarded quantities occur. As the regarded problems are transient, a grid-adaptive scheme is used to re-adapt the mesh during runtime of the simulator (*Barlag*, 1997 and *Kaiser*, 2001). In this work *linear* interpolation functions are used to describe the local changes of spatially varying quantities. Thus, the second order derivatives of those quantities (i.e., the curvatures in space) are a good estimate for the neglected higher order parts of a Taylor series analysis of the spatial distribution. This means that the second order derivatives can be used to trigger a local refinement of the grid.

If a continuous natural system is replaced by a discretized system, a procedure is required to minimize the introduced errors. In the following, the *method of weighted residuals* (MWR) will be used for this task.

A set of single steps must be performed for the FEM:

- Evaluation of the conservation equations (Chapter 2.1).
- Spatial interpolation of the domain (Ritz’ method, Chapter 3.1.2).
- Orthogonalization of the equations (MWR, Chapter 3.1.3).
- Evaluation of element or patch based system matrices (Chapter 3.1.3).
- Assembling and solving of a linear system of equations (Chapters 3.1.7 to 3.1.10).

These steps will be treated in short form in the following chapters. For a more detailed description on applying the FEM the books by *Gresho and Sani* (1998) and *Zienkiewicz and Godbole* (1975) can be recommended.

3.1.2 Spatial interpolation (Ritz’ approach)

Some factors of the conservation equations can be regarded as constant in space, while other have non-zero spatial derivatives. These are described in the FEM with the help of a set of functions, called “basis functions”. These are chosen in a way that they fulfil the following restrictions:

- Each function is based on a node of the mesh.
- The function is equal to unity on its node.
- The function is equal to zero on all other nodes.
- The sum of all functions is equal to unity in the domain.

These features can be fulfilled by multiple functions, more simple linear ones or higher order polynomials. The necessary order of the polynomials is determined by the order of the spatial derivatives in the conservation equations. If second order derivatives are used in the conservation equations, at least quadratic basis functions must be used. In this work the Gauss-Green integral theorem is used to reduce the order of the derivatives, so that linear basis functions can be used.

The variable u is approximated by û, which is the sum of the products of all node values u_i and their basis functions ω_i over all nodes:

$$u \approx \hat{u} = \sum_{i=1}^{\text{nodes}} \omega_i u_i \tag{3.1}$$

In Figure 3.3 a set of linear basis functions is presented. The upper picture shows the linear functions belonging to each node i, the lower picture presents the resulting interpolated function. This function is the sum of all nodal basis functions after they are multiplied with the appropriate node value.

In the following the sum of products between basis functions and primary variables will be replaced by a vector notation:

$$\sum_{i=1}^{n} \omega^i u^i = \overline{\boldsymbol{\omega}}\, \mathbf{u} \tag{3.2}$$

In this expression $\overline{\boldsymbol{\omega}}$ denotes the horizontally oriented vector of basis functions for all nodes and $\mathbf{u}$ the vector of node values of the regarded variable.

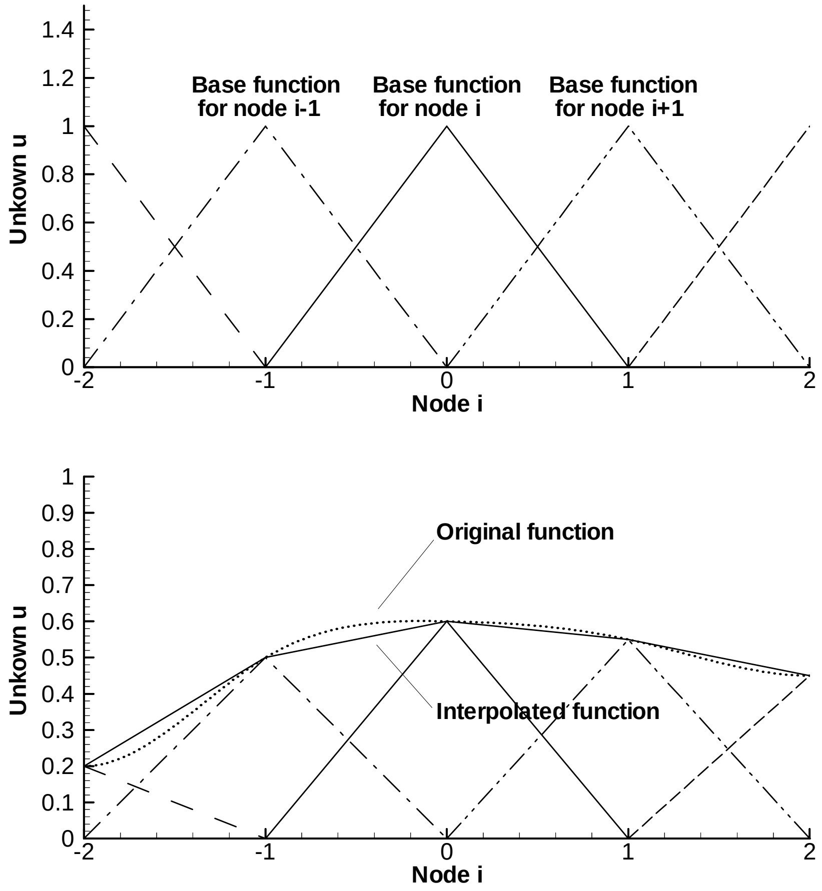


Fig. 3.3: Ritz' method for interpolation. The upper picture shows the basis functions and the lower picture the resulting piecewise linear interpolation of a given function.

Generally this approach has to be chosen for all variables that vary in space. Due to simplicity of the numerical implementation, it is sometimes convenient to assume that the spatial variance is sufficiently approximated by using element-by-element constant variables. This is the case for material parameters like porosity and permeability of the porous medium, as this enables sharp changes of the material properties on element boundaries. For most of the calculations presented in this work the porosity, permeability, storativity, compressibility and capillary pressure function are assumed to be element-by-element constant. In the following the pressures, relative permeabilities, viscosities, densities, saturations and concentrations are evaluated by using Equation 3.1. If the density of a phase is in the denominator of an expression, the density is assumed to be constant in the element for the denominator.

All dependent values that vary in space (u_a and u_b in Equation 3.3) can either be replaced by a constant value for each element or by a Ritz' approximation. The pressures, saturations and concentrations as the primary unknown variables must be replaced by the approximation gained from the Ritz' approach (Equation 3.1). To avoid the multiplication of the separate linear approximations for all dependent variables, some authors suggest using the *Group FEM* or *Product Approximation* (Equation 3.3). The *Group FEM* approach is equivalent to a local linearization, thus the accuracy is lower than for the standard *Galerkin FEM.*

$$u_a u_b \approx \sum_{i=1}^{\text{nodes}} \omega^i u_a^i \cdot \sum_{i=1}^{\text{nodes}} \omega^i u_b^i \approx \sum_{i=1}^{\text{nodes}} \omega_i u_a^i u_b^i \tag{3.3}$$

This approach was suggested by *Swartz and Wendroff* (1969) and later by *Christie et al.* (1981) and other authors. It will not be used in this work, because it reduces the accuracy of the numerical solution. Anyhow it would be easy to change the numerical scheme appropriately, in order to trade numerical accuracy for computational speed.

The *Group FEM* can have a performance advantage if numerous spatial variables are multiplied in a single term. For the evaluation of the mobility $k_{r_\alpha} \mathbf{k} / \mu_\alpha$ on a single hexahedral element with three integration points for each direction (compare Equation 3.19), the *Group FEM* requires eight evaluations of the mobilities on the nodes and afterwards 27 interpolations of the mobilities to the Gaussian points. The *Galerkin FEM* requires to interpolate all relevant primary variables to the 27 Gaussian points and to evaluate relative permeabilities and viscosities on each of them. Obviously the effort for the *Galerkin FEM* is much bigger, anyhow it was chosen due to the better accuracy.

3.1.3 Orthogonalization of the equations

The resulting superset pressure equation (2.17) and the saturation equation (2.20) are treated by the *Method of Weighted Residuals* by multiplying them with the vector of weighting functions and integrating the result over the domain V. Only a linear independent subset of the total space of weighting functions is regarded. The subset has as many member functions φ_i as the system has degrees of freedom. In this work each node has one degree of freedom for each equation, so the number of weighting functions is chosen equal to the number of nodes. Thus, we have to solve multiple equations, one for each node i:

Pressure field, for all $i \in \{0, ..., \text{nodes} - 1\}$ (3.4)

$$\int_V \varphi_i \left(\sum_{\alpha=0}^{\text{phases-1}} \left[\left(\frac{nS_\alpha}{\rho_\alpha} \frac{\partial \rho_\alpha}{\partial p_\alpha} + S_0 S_\alpha + n \frac{\partial S_\alpha}{\partial p_\alpha} \right) \frac{\partial p_\alpha}{\partial t} \right.\right.$$
$$\left.\left. - \frac{1}{\rho_\alpha} \operatorname{div} \left(\rho_\alpha \frac{k_{r_\alpha} \mathbf{k}}{\mu_\alpha} (\operatorname{grad} p_\alpha - \rho_\alpha \mathbf{g}) \right) - Q_\alpha \right] \right) dV = 0$$

Saturation field, for all $i \in \{0, ..., \text{nodes} - 1\}$ (3.5)

$$\int_V \varphi_i \left(nS_\alpha \frac{1}{\rho_\alpha} \frac{\partial \rho_\alpha}{\partial p_\alpha} \frac{\partial p_\alpha}{\partial t} + n \frac{\partial S_\alpha}{\partial t} \right.$$
$$\left. - \frac{1}{\rho_\alpha} \operatorname{div} \left(\rho_\alpha \frac{k_{r_\alpha} \mathbf{k}}{\mu_\alpha} (\operatorname{grad} p_\alpha - \rho_\alpha \mathbf{g}) \right) - Q_\alpha \right) dV = 0$$

In the Bubnov-Galerkin FEM the weighting functions φ_i are equal to the basis functions ω_i of the Ritz' approach (Equation 3.1). This is not strictly necessary. In Chapter 3.1.6 some effects of different weighting functions are presented.

Rearranging integrals and sums of the pressure field equation leads to:

Pressure field, for all $i \in \{0,\ldots,\text{nodes}-1\}$ (3.6)

$$\begin{aligned}
\sum_{\alpha=0}^{\text{phases-1}} \Bigg[& \int_V \varphi_i \left(\frac{nS_\alpha}{\rho_\alpha}\frac{\partial\rho_\alpha}{\partial p_\alpha} + S_0 S_\alpha + n\frac{\partial S_\alpha}{\partial p_\alpha} \right) \frac{\partial p_\alpha}{\partial t}\, dV \\
& - \int_V \varphi_i \frac{1}{\rho_\alpha} \operatorname{div}\left(\rho_\alpha \frac{k_{r_\alpha}\mathbf{k}}{\mu_\alpha} \left(\operatorname{grad} p_\alpha - \rho_\alpha \mathbf{g}\right) \right) dV \\
& - \int_V \varphi_i Q_\alpha \, dV \Bigg] = 0
\end{aligned}$$

The second order derivatives cannot be treated adequately by linear finite elements. The Gauss-Green integral theorem transforms them into lower order derivatives:

Pressure field, for all $i \in \{0,\ldots,\text{nodes}-1\}$ (3.7)

$$\begin{aligned}
\sum_{\alpha=0}^{\text{phases-1}} \Bigg[& \int_V \varphi_i \left(\frac{nS_\alpha}{\rho_\alpha}\frac{\partial\rho_\alpha}{\partial p_\alpha} + S_0 S_\alpha + n\frac{\partial S_\alpha}{\partial p_\alpha} \right) \frac{\partial p_\alpha}{\partial t}\, dV \\
& - \int_S \varphi_i \frac{k_{r_\alpha}\mathbf{k}}{\mu_\alpha} \left(\operatorname{grad} p_\alpha - \rho_\alpha \mathbf{g}\right) dS \\
& + \int_V \operatorname{grad}\left(\varphi_i \frac{1}{\rho_\alpha} \right) \rho_\alpha \frac{k_{r_\alpha}\mathbf{k}}{\mu_\alpha} \left(\operatorname{grad} p_\alpha - \rho_\alpha \mathbf{g}\right) dV \\
& - \int_V \varphi_i Q_\alpha \, dV \Bigg] = 0
\end{aligned}$$

Saturation field, for all $i \in \{0,\ldots,\text{nodes}-1\}$ (3.8)

$$\begin{aligned}
& \int_V \varphi_i \left(nS_\alpha \frac{1}{\rho_\alpha}\frac{\partial\rho_\alpha}{\partial p_\alpha}\frac{\partial p_\alpha}{\partial t} + n\frac{\partial S_\alpha}{\partial t} \right) dV \\
& - \int_S \varphi_i \frac{k_{r_\alpha}\mathbf{k}}{\mu_\alpha} \left(\operatorname{grad} p_\alpha - \rho_\alpha \mathbf{g}\right) dS \\
& + \int_V \operatorname{grad}\left(\varphi_i \frac{1}{\rho_\alpha} \right) \rho_\alpha \frac{k_{r_\alpha}\mathbf{k}}{\mu_\alpha} \left(\operatorname{grad} p_\alpha - \rho_\alpha \mathbf{g}\right) dV \\
& - \int_V \varphi_i Q_\alpha \, dV = 0
\end{aligned}$$

The integrals over the surface S in this formulation denote fluxes into and out of the regarded domain. They will be treated separately by applying appropriate sink and source boundary conditions to the right hand side of the equation system and therefore will not be regarded in this context anymore.

3.1.4 Application of Ritz' approach on the orthogonalized equations

Replacing the spatial variables of the regarded variables according to Ritz' approach (Equation 3.2) and some rearrangement leads to:

Pressure field, for all $\mathrm{i} \in \{0,\ldots,\mathrm{nodes}-1\}$ (3.9)

$$\sum_{\alpha=0}^{\text{phases-1}} \left[\int_V \varphi_i \left(\frac{nS_\alpha}{\rho_\alpha}\frac{\partial \rho_\alpha}{\partial p_\alpha} + S_0 S_\alpha + n\frac{\partial S_\alpha}{\partial p_\alpha} \right) \overline{\boldsymbol{\varpi}}\, dV \frac{\partial \mathbf{p}_\alpha}{\partial t} \right.$$

$$+ \int_V \operatorname{grad}\left(\varphi_i \frac{1}{\rho_\alpha} \right) \rho_\alpha \frac{k_{r_\alpha}\mathbf{k}}{\mu_\alpha} \left(\operatorname{grad} \overline{\boldsymbol{\varpi}} \right) dV\, \mathbf{p}_\alpha$$

$$\left. = \int_V \operatorname{grad}\left(\varphi_i \frac{1}{\rho_\alpha} \right) \rho_\alpha \frac{k_{r_\alpha}\mathbf{k}}{\mu_\alpha} \rho_\alpha \mathbf{g}\, dV + \int_V \varphi_i Q_\alpha\, dV \right]$$

Saturation field, for all $\mathrm{i} \in \{0,\ldots,\mathrm{nodes}-1\}$ (3.10)

$$\int_V \varphi_i n \overline{\boldsymbol{\varpi}}\, dV \frac{\partial \mathbf{S}_\alpha}{\partial t} =$$

$$- \int_V \varphi_i \frac{nS_\alpha}{\rho_\alpha}\frac{\partial \rho_\alpha}{\partial p_\alpha} \overline{\boldsymbol{\varpi}}\, dV \frac{\partial \mathbf{p}_\alpha}{\partial t}$$

$$- \int_V \operatorname{grad}\left(\varphi_i \frac{1}{\rho_\alpha} \right) \rho_\alpha \frac{k_{r_\alpha}\mathbf{k}}{\mu_\alpha} \left(\operatorname{grad} \overline{\boldsymbol{\varpi}} \right) dV\, \mathbf{p}_\alpha$$

$$+ \int_V \operatorname{grad}\left(\varphi_i \frac{1}{\rho_\alpha} \right) \rho_\alpha \frac{k_{r_\alpha}\mathbf{k}}{\mu_\alpha} \rho_\alpha \mathbf{g}\, dV$$

$$+ \int_V \varphi_i Q_\alpha\, dV$$

By this procedure the regarded primary variables are extracted from the integrals. In the following the point of view will be switched from a global to an element-by-element viewpoint. Another possibility would be a patch-wise evaluation. The patch based scheme has a larger memory demand, as the system matrices of the patches overlap. Therefore an element-by-element scheme is used here. First the local element integrals are evaluated and later the local element-by-element view is replaced again by the global view by summing up all element integrals:

$$\int_V (\ldots) dV = \sum_{E=0}^{\text{elements}-1} \int_{V_E} (\ldots) dV_E \qquad (3.11)$$

To evaluate the local element integrals, a vertically aligned vector $\boldsymbol{\varphi}^E$ is introduced for the weighting functions of an element. It contains the parts of the weighting functions φ_i belonging to the regarded element E. Both the integral over the sum of all patch based weighting functions and the sum over all element integrals are equal to the volume of the regarded domain:

$$\int_{V} \sum_{i=0}^{\text{nodes}-1} \varphi_i \mathrm{d}V = \sum_{E=0}^{\text{elements}-1} \int_{V_E} \boldsymbol{\varphi}^E \mathrm{d}V_E = V \tag{3.12}$$

The same procedure is used for the horizontally aligned vector of basis functions $\overline{\boldsymbol{\varpi}}$. In the following the subset $\overline{\boldsymbol{\varpi}}^E$, which is belonging to the regarded element E, is used. Both vectors have a number of entries that is equivalent to the number of nodes for element E.

3.1.5 Resulting element matrices

To simplify the expressions, a set of matrices and vectors is introduced. In these matrices all spatial variables are evaluated according to their local position. The Ritz' approach is seemingly not used here for dependent variables. It is used in the actual integration procedure but is left out here, because otherwise the readability of the equations would suffer:

Pressure dependent storage : (3.13)

$$\mathbf{A}_\alpha^E = \int_{V_E} \boldsymbol{\varphi}^E \left(\underbrace{\frac{nS_\alpha}{\rho_\alpha} \frac{\partial \rho_\alpha}{\partial p_\alpha}}_{\text{I}} + \underbrace{S_0 S_\alpha}_{\text{II}} + n \underbrace{\frac{\partial S_\alpha}{\partial p_\alpha}}_{\text{III}} \right) \overline{\boldsymbol{\varpi}}^E \mathrm{d}V_E$$

Mass storage for multiphase flow : (3.14)

$$\mathbf{B}_\alpha^E = \int_{V_E} \boldsymbol{\varphi}^E n \overline{\boldsymbol{\varpi}}^E \mathrm{d}V_E$$

Flux : (3.15)

$$\mathbf{C}_\alpha^E = \int_{V_E} \mathrm{grad}\, \boldsymbol{\varphi}^E \frac{k_{r_\alpha} \mathbf{k}}{\mu_\alpha} \mathrm{grad}\, \overline{\boldsymbol{\varpi}}^E \, \mathrm{d}V_E - \int_{V_E} \boldsymbol{\varphi}^E \mathrm{grad}\, \rho_\alpha \frac{k_{r_\alpha} \mathbf{k}}{\rho_\alpha \mu_\alpha} \mathrm{grad}\, \overline{\boldsymbol{\varpi}}^E \, \mathrm{d}V_E$$

Gravity : (3.16)

$$\mathbf{D}_\alpha^E = \int_{V_E} \mathrm{grad}\, \boldsymbol{\varphi}^E \frac{k_{r_\alpha} \mathbf{k}}{\mu_\alpha} \rho_\alpha \mathbf{g} \, \mathrm{d}V_E - \int_{V_E} \boldsymbol{\varphi}^E \, \mathrm{grad}\, \rho_\alpha \frac{k_{r_\alpha} \mathbf{k}}{\rho_\alpha \mu_\alpha} \rho_\alpha \mathbf{g} \, \mathrm{d}V_E$$

Sinks and sources : (3.17)

$$\mathbf{E}_\alpha^E = \int_{V_E} \boldsymbol{\varphi}^E Q_\alpha \mathrm{d}V_E$$

It must be pointed out that $\boldsymbol{\varphi}^E$ and $\overline{\boldsymbol{\varpi}}^E$ are function vectors. The derived matrices are assembled to the global system of equations according to the procedure outlined in Chapter 3.1.7.

For further simplification of the equations, the sum of all element matrices of a certain type is abbreviated:

$$\mathbf{A}_\alpha = \sum_{E=0}^{\text{elements}-1} \mathbf{A}_\alpha^E \tag{3.18}$$

The same procedure is used for the element matrices **B**, **C** and **D**. The matrices of type **E** are not treated as matrices in the code, but instead equivalent single node loads are evaluated.

3.1.5.1 Element integration

The element matrices can be integrated analytically for one-dimensional elements (Appendix A), for elements of higher dimension a Gaussian quadrature is used. The Gaussian quadrature is performed on a normalized element by replacing the integral with the sum of single point values and their appropriate weight (*Lapidus and Pinder*, 1999).

$$\begin{aligned}\int_V f(x,y,z)\,dV &= \int_{-1}^{1}\int_{-1}^{1}\int_{-1}^{1} f(r,s,t)\ \det[\mathbf{J}]\,dr\,ds\,dt \\ &\approx \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n}\left(w_i w_j w_k f(r_i,s_j,t_k)\det[\mathbf{J}]\right)\end{aligned} \tag{3.19}$$

In this expression det[**J**] denotes the determinant of the Jacobian matrix. The weighting factors w are dependent on the number of integration points n and the current integration point in local coordinates r, s and t (Table 3.2). The needed number of integration points largely depends on the shape of the elements and on the nonlinearity of the integrands. For simple linear problems and rectangular element shape two Gauss points in each local element direction are sufficient; for nonlinear problems or arbitrary shaped elements at least three Gauss points are required to reach sufficient accuracy in the integration.

Tab. 3.2: Integration points and weighting factors for Gauss integration

Number of Gauss points n	Integration point in local coordinates $\pm r, s, t$	Weighting factor $w_{i,j,k}$
1	0.000000	2.000000
2	0.577350	1.000000
3	0.774600	0.555556
	0.000000	0.888889
4	0.861136	0.347855
	0.339981	0.652145

3.1.6 Special weighting functions

The weighting functions $\boldsymbol{\varphi}$ are not necessarily equal to the basis functions $\boldsymbol{\omega}$. They are equal for the special case of the *Bubnov-Galerkin FEM*. Other variants are possible: piecewise constant functions (CVFEM), higher order polynomials, point collocation etc. In the following two special methods, upwinding and mass lumping, will be focused upon. It must be pointed out that those methods are in general of lower accuracy than the *Bubnov-Galerkin FEM*. Anyhow they are very useful, as they produce more stable results. For hyperbolic problems these methods may be necessary to achieve any result at all, though the modeller should be careful not to believe too much in “smooth” solutions. This smoothness is paid for by unphysical numerical diffusion that is added to the system. Numerical oscillations (“wiggles”) can be valuable information for the modeller, as they tell him that his numerical solution is doubtable. Or, as J. Ferziger was quoted by *Gresho and Sani* (1999): “*The greatest disaster one can encounter in computation is not instability or lack of convergence, but results that are good enough to be believable but bad enough to cause trouble.*”

3.1.6.1 Upwinding

Upwinding is based on the idea to modify the weighting functions in a way that ensures a higher weighting of the information coming from *upstream* (*Brooks & Hughes,* 1982). Adding either higher or lower order polynomials to the weighting functions can do this. *Helmig* (1993) proposed to used higher order polynomials for multiphase flow problems. In this work the standard s*treamline upwind Petrov-Galerkin* (SUPG) method was used for both multiphase flow and transport problems and was enhanced by an additional mobility upwinding for the multiphase flow processes.

Generally upwinding of the weighting functions is useful for advection dominated problems as expressed by the Peclet number Pe (Equation 2.31). For multiphase flow problems the Peclet number can be interpreted as the relation between phase pressure gradients and capillary pressure gradients. If the capillary pressures are low and linear basis functions are used, it is necessary to use an upwinding method in multiphase flow problems, because otherwise the solutions will loose their physical meaning.

For the SUPG scheme a constant value is added to the weighting functions (Figure 3.4) on the streamline upward side of the weighting function. On the opposite side the same value is subtracted from the weighting function. The same is true for elements of higher dimension.

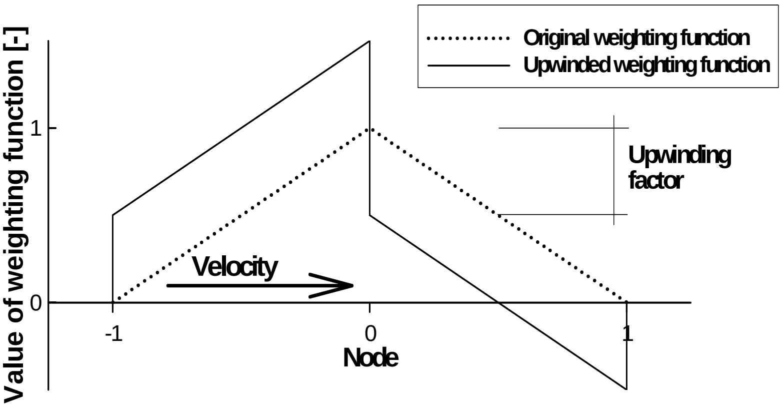


Fig. 3.4: Weighting functions for a one-dimensional element. The weighting functions are shifted upwards on the upstream side and downwards on the downstream side.

Due to the definition of the FEM, this modified weighting function has to be used for all terms of the regarded conservation equation. Anyhow it can be advantageous to upwind only *parts* of the regarded PDE ("Partial upwinding"). For the tracer transport equation it is possible to limit the upwinding to the advective term of the PDE ("Advection upwinding").

The upwinding factor α can furthermore be related (Equation 3.20) to the Peclet number:

$$\alpha = \begin{cases} 0 & \mathrm{Pe} < 1 \\ \alpha_0 \left(\dfrac{\mathrm{Pe} - 1}{\mathrm{Pe}} \right) & \mathrm{Pe} \geq 1 \end{cases} \tag{3.20}$$

This has the advantage that only in the advection dominated parts of the domain upwinding will be used, while in the diffusion dominated rest of the domain the original weighting functions are used and therefore higher accuracy is reached. A similar effect can be gained by using advection upwinding, because the impact of upwinding vanishes if the advective term looses its dominance in the PDE.

For multiphase flow problems it is necessary to use an upwinded scheme to achieve the physically correct behaviour. In Figure 3.5 the results of the not upwinded and the SUPG schemes are compared to the analytical solution of the Buckley-Leverett test case (Chapter 4.3). This test case is difficult to handle, as no capillary pressures are regarded. Thus, the behaviour is purely advective. Furthermore the front is self-sharpening. It is obvious that the scheme without upwinding largely misses to catch the system behaviour, though it still fulfils the mass conservation and is, in contrast to the statement by *Helmig and Huber* (1998), stable, though not oscillation free. The SUPG scheme shows a better performance and is much nearer to the analytical solution, though oscillations are still obvious.

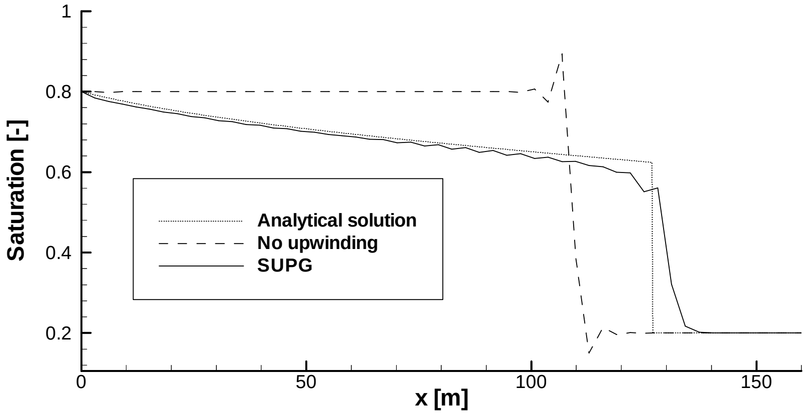


Fig. 3.5: Impact of SUPG on the solution of the Buckley-Leverett test case. The solution with SUPG is significantly closer to the analytical solution than the standard solution.

Another very simple approach is streamline upwards shifting of the Gaussian integration points (Table 3.2) in the element (Figure 3.6). This procedure will be called *Gaussian point upwinding* (GPU) in the following. The upstream weighting of the fluid mobility during the integration of the flux and gravitation matrices (Equations 3.15 and 3.16) largely enhances the results. Though this is not strictly derivable from a mathematical point of view, this approach has proven to be very useful in multiphase flow simulations. For this approach the normalized velocity vector in local element coordinates is multiplied with the user specified upwind factor and then subtracted from the local integration point coordinates.

As the coordinates have to be in the range between minus one and plus one, two alternative limiting strategies are added. The first one scales the vector appropriately, so that none of the shifted points will violate the constraint (Figure 3.6). Thus, a part of the spatial information transverse to the flow direction is kept.

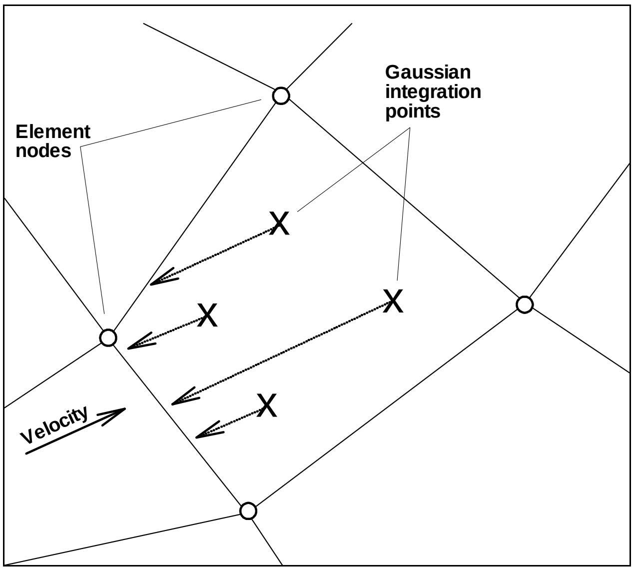


Fig. 3.6: Upwinding of Gaussian integration points. The integration points are shifted streamline upwards until an element bound is reached.

The second, more simplistic approach, limits the result of the shifting operation for each of the coordinates separately to the desired range. In practice this means that for large upwinding factors a single node value will determine the mobility assigned to the whole element. This is equivalent to the *fully upwind Galerkin* (FUG) method as proposed by *Helmig et al.* (1998). Some spatial information is lost in this procedure, compared to the one proposed before. The FUG method has shown good performance for heterogeneous systems, where the entry behaviour of the non-wetting phase is difficult to reproduce numerically.

The impact of the GPU for the fluid phase mobility is presented in Figure 3.7. It shows again the results for the Buckley-Leverett test case. In comparison to the SUPG scheme the oscillations on the plateau are largely reduced and the shock front is matched significantly better.

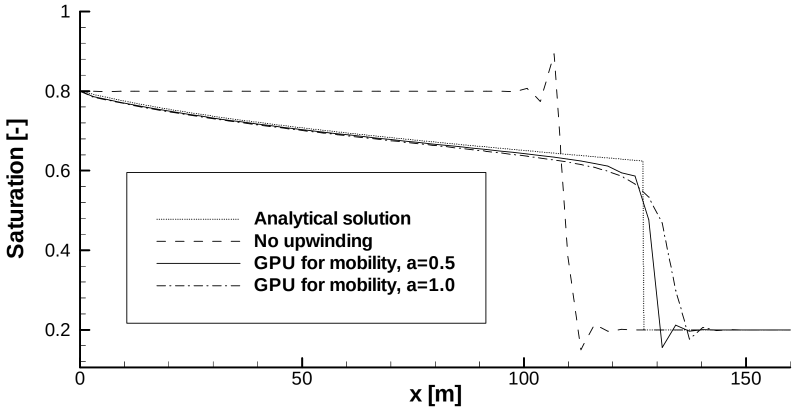


Fig. 3.7: Impact of mobility upwinding on the solution of the Buckley-Leverett test case. The solution with GPU is close to the analytical solution and is oscillation free on the plateau.

Additionally the SUPG and the mobility upwinding scheme can be combined. The results are not presented here, as they resemble the expected behaviour: If both methods are used to their full extent, the front is heavily smoothed and shows no more oscillations. If both methods are applied with a halved upwind factor, the results are between the results of the single methods.

3.1.6.2 Mass lumping

This procedure is very common to suppress wiggles which result from the inability of the FEM to reproduce high spatial frequencies of the field variables. Already the approximation of the initial conditions with the mass matrix will produce wiggles in the solution, if the initial conditions are not smooth enough.

The basic idea behind mass lumping is to modify the mass matrix **M** (Matrix **A** in Equation 3.13 and matrix **B** in Equation 3.14) in a way that only the diagonal elements have non-zero entries. For the numerical simulation of partially saturated systems this was first suggested by *Neumann* (1973). This is equivalent to the *decoupling* of each node from his neighbours, as far as the mass matrix is concerned. In the FEM meaning this can be done either by summing up the mass matrix rows or by using modified weighting functions:

$$\mathbf{M} = \int_V \boldsymbol{\varphi}(\ldots)\overline{\boldsymbol{\omega}}\mathrm{dV} \rightarrow \mathbf{M}_{\mathrm{ML}} = \boldsymbol{\delta}\int_V (\ldots)\overline{\boldsymbol{\omega}}\mathrm{dV} \tag{3.21}$$

In this equation $\boldsymbol{\delta}$ denotes a vector, containing the parts of a Kronecker delta at each element node that are belonging to the regarded element. This is equivalent to moving all weight of the weighting functions to the single node i. This type of formulation is related to Box-FEM and Control-Volume-FEM schemes. The mass lumping procedure can be interpreted as using a different numerical method for the storage term of the PDE. Here it is optionally applied to the storage matrices **A** (Equation 3.13) and **B** (Equation 3.14). The procedure is equivalent to replacing the storage term by a typical volume that is assigned to the regarded node. This volume is gained in the mass lumping procedure by integrating over the patch volume; it can also be gained by using a Voronoi partitioning or other partitioning methods in order to assign a specific volume to the node.

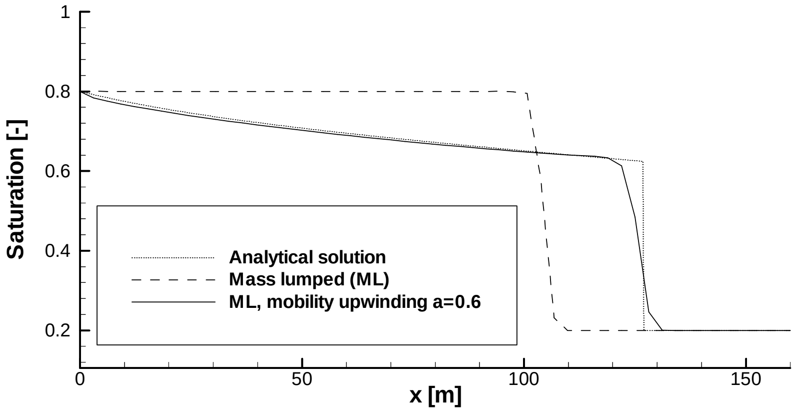


Fig. 3.8: Impact of mass lumping and mobility upwinding on the solution of the Buckley-Leverett test case. Mass lumping is a very effective tool to suppress wiggles in the solution. Together with the mobility upwinding a solution close to the analytical one can be reached.

Figure 3.8 shows the impact of mass lumping on the results of the Buckley-Leverett test case. For the not upwinded solution the wrong physical behaviour is still obvious, though the solution is now oscillation free. For the mobility upwinding the solution was enhanced, though now a small mass loss becomes obvious. The combination of SUPG and mass lumping is not possible, as the lumping of the mass matrices eliminates the influence of the SUPG on those matrices.

It must be pointed out that lumping the mass matrices reduces the accuracy of the results. This is especially true for non-uniform grids. Finalizing, mass lumping can be a very useful enhancement for FEM formulations, though it should be used with care and not in general. For critical situations where large curvatures of the field cannot be resolved by the FEM (e.g. the upstream side of sources of tracers, self-sharpening fronts or fracture intersections) mass lumping should be switched on automatically as needed.

3.1.7 Assembling of systems of linear equations

The matrices and vectors introduced with the Equations 3.13 to 3.17 form the basis of a construction set for multiple physical problems. Together with the temporal semi-discretization (introduced in Chapter 3.3), it is possible to assemble the global system of linear equations for each of the primary variables. After an introductory part concerning the storage of the element matrices in the global system matrix, the typical assemble procedures for the superset pressure equation and the saturation field equation will be outlined. The adaptation to several classes of simpler problems is outlined in Chapter 3.5.1.

3.1.7.1 Principles of matrix storage handling

Before assembling the system matrices to the global system of equations, it is useful to set up a translation table between the node numbers and their appropriate line/column in the system of linear equations. This process, called renumbering, leads to ordered systems of equations that show a banded structure. In Figure 3.9 the map of non-zero entries in the system of linear equations for the four fracture system described in Chapter 4.7 is presented.

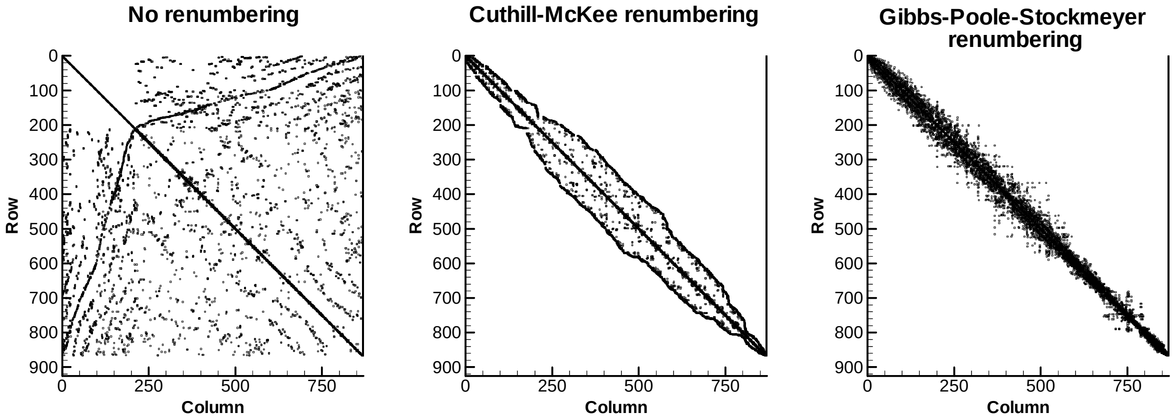


Fig. 3.9: Map of non-zero entries in the system of linear equations. Both Cuthill-McKee and Gibbs-Poole-Stockmeyer renumbering significantly reduce the bandwidth of the matrix.

The left image shows the structure of the system of linear equations if no renumbering is applied. Obviously both the *Cuthill-McKee* (1969) and the *Gibbs-Poole-Stockmeyer* (1976) renumbering scheme largely reduce the bandwidth. The *Gibbs-Poole-Stockmeyer* scheme (implemented by R. Ratke) produces the most compact system, avoiding the strong outer bands of the *Cuthill-McKee* scheme. The renumbering not only compacts the system, but also has a positive impact on the convergence of iterative solvers.

The matrix of the system of linear equations is not stored completely, as most of the entries are equal to zero. Instead it is useful to use a sparse storage like *compressed row* or *compressed column* storage. The main idea of those storage schemes is not to store the rows or columns themselves, but to have a set of two arrays for each row or column that is limited to the rows or columns that actually have entries. The first array contains a list of indices, denoting the column or row to which the entry of the second array belongs.

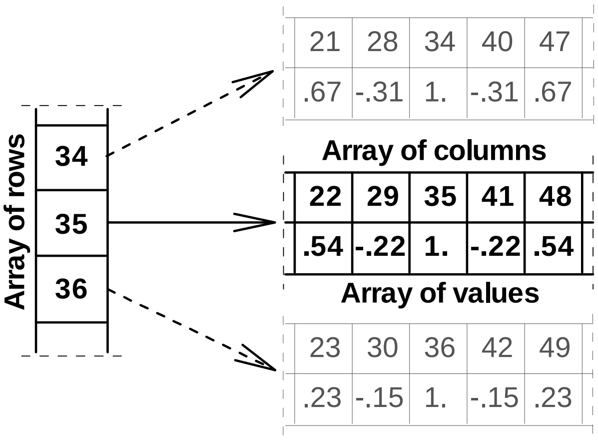


Fig. 3.10: Principle of compressed row storage. Only a limited subset of the matrix is stored.

In Figure 3.10 the basic idea of compressed row storage is illustrated. An array of pointers is used to refer to the appropriate arrays of column indices and values. In the example presented in Figure 3.10 the access to matrix element **A**(35,29) would first result in a pointer to the double array belonging to row 35. Searching in this array for column 29 returns the corresponding value of –0.22. It is useful to extract the diagonal entries from this scheme, as it is faster to store them into a separate vector. In RockFlow a compressed row storage scheme (implemented by M. Schulze-Ruhfus) and a combined compressed row/column storage scheme (implemented by R. Ratke) are used.

3.1.7.2 Discretized field equations

To describe the field equations for multiple phases α, the matrices for all phases must be considered. Furthermore, it is necessary to choose a reference pressure in order to reduce the number of variables (Equation 2.7) in the pressure field equation.

After applying the temporal semi-discretization (Chapter 3.3) on the pressure field equation (2.19), this leads to the matrix formulation for the pressure field in terms of a freely selectable reference phase pressure p_{ref}:

$$
\begin{aligned}
&\textit{Pressure field :} \\
&\sum_{\alpha=0}^{\text{phases}-1}\left(\frac{1}{\Delta t}\mathbf{A}_{\alpha_{I}}^{t+\Theta\Delta}+\Theta\mathbf{C}_{\alpha}^{t+\Theta\Delta}\right)\mathbf{p}_{ref}^{t+\Delta t}= \\
&\sum_{\alpha=0}^{\text{phases}-1}\left(\frac{1}{\Delta t}\mathbf{A}_{\alpha_{I,II,III}}^{t+\Theta\Delta}-(1-\Theta)\mathbf{C}_{\alpha}^{t+\Theta\Delta}\right)\mathbf{p}_{ref}^{t}+ \\
&\sum_{\alpha=0}^{\text{phases}-1}\left(\frac{1}{\Delta t}\mathbf{A}_{\alpha_{I,II,III}}^{t+\Theta\Delta}\left(\left(\mathbf{p}_{c_{0,\alpha}}^{t+\Delta t}-\mathbf{p}_{c_{0,ref}}^{t+\Delta t}\right)-\left(\mathbf{p}_{c_{0,\alpha}}^{t}-\mathbf{p}_{c_{0,ref}}^{t}\right)\right)\right)+ \\
&\sum_{\alpha=0}^{\text{phases}-1}\left(\mathbf{C}_{\alpha}^{t+\Theta\Delta}\left(\mathbf{p}_{c_{0,\alpha}}^{t+\Theta\Delta t}-\mathbf{p}_{c_{0,ref}}^{t+\Theta\Delta t}\right)\right)+ \\
&\sum_{\alpha=0}^{\text{phases}-1}\left(\mathbf{D}_{\alpha}^{t+\Theta\Delta}+\mathbf{E}_{\alpha}^{t+\Theta\Delta}\right)
\end{aligned}
\tag{3.22}
$$

After solving the pressure field equation, additionally phases-1 saturation field equations (2.20) must be solved. Again first the temporal semi-discretization (Chapter 3.3) must be applied on the equations. Then the discretized equation can be assembled using the same matrices as for the previously described pressure field equation, apart from the fact that the mass storage matrix **B** must be considered additionally. This approach is only valid if the same time collocation point θ (Chapter 3.3) is used in both equations, otherwise it would be necessary to recalculate the matrices. In this work it was assumed that θ is chosen equal for the pressure and the saturation fields.

$$
\begin{aligned}
&\textit{Saturation fields :} \\
&\frac{1}{\Delta t}\mathbf{B}\,\mathbf{S}_{\alpha}^{t+\Delta t}=\frac{1}{\Delta t}\mathbf{B}\,\mathbf{S}_{\alpha}^{t} \\
&-\frac{1}{\Delta t}\mathbf{A}_{\alpha_{I}}^{t+\Delta t}\left(\left(\mathbf{p}_{ref}^{t+\Delta t}-\mathbf{p}_{ref}^{t}\right)-\left(\mathbf{p}_{c_{0,\alpha}}^{t+\Delta t}-\mathbf{p}_{c_{0,ref}}^{t+\Delta t}\right)+\left(\mathbf{p}_{c_{0,\alpha}}^{t}-\mathbf{p}_{c_{0,ref}}^{t}\right)\right)+ \\
&-\mathbf{C}_{\alpha}^{t+\Delta t}\left(\mathbf{p}_{ref}^{t+\Theta\Delta t}-\mathbf{p}_{c_{0,\alpha}}^{t+\Theta\Delta t}+\mathbf{p}_{c_{0,ref}}^{t+\Theta\Delta t}\right)+ \\
&\mathbf{D}_{\alpha}^{t+\Delta t}+\mathbf{E}_{\alpha}^{t+\Delta t}
\end{aligned}
\tag{3.23}
$$

If the regarded phase “α" is not equal to the reference phase “ref” that was previously considered in the pressure field calculation, the phase pressures must be calculated with the help of Equation 2.7. If the phases are equal, the capillary pressures in the prior equation are equal to zero. Further aspects of treating the saturation field equations are covered in Chapter 3.5.1.5.

If the mass matrix is built using the mass lumping procedure described in Chapter 3.1.6.2, the system of linear equations will only have entries on the diagonal. Thus, no equation solver is needed; the values can directly be evaluated. If the simulator is working with adaptive grids, the treatment of irregular nodes (Chapter 3.1.9.1) introduces new values on both sides of the diagonal, so that a system of equations must be solved again.

3.1.7.3 Treatment of systems with large vertical extent

If systems with very large vertical extent and low horizontal gradients are regarded, the limited number accuracy of computer programs can lead into numerical problems. In the Equation 3.22 the right hand side vector will carry very large numbers arising from the gravity terms in the equations. Additionally, the solution vector will contain very large numbers, which mainly reflect the hydrostatic pressure in the system. Due to the limited accuracy, the small pressure differences that actually drive the fluid movement in the system can easily vanish in the numerical noise that is introduced by the limited accuracy of the floating point number representation. This is especially true, if an iterative equation solver is used, because the rounding errors increase due to the multiple iteration cycles.

To circumvent the problems outlined above, it is possible to use high accuracy numerical libraries. These are available in arbitrary precision and could easily solve the problems. The drawback of those libraries is the extremely low performance, as the basic operations are no longer directly processed in the floating point unit(s) of the processor, but are split into multiple operations.

Another possibility to steer clear of those problems is to change the way of assembling the system of equations. The primary variable $\mathbf{p}_{\mathrm{ref}}$ is replaced by the difference $\mathbf{dp}_{\mathrm{ref}}$ to the known prior time level. The pressure on the old time level already contains the hydrostatic pressure, so that it is no longer contained in the new solution vector. Introducing a new primary variable **dp** performs this:

$$\left.\begin{aligned}\mathbf{dp}_{\mathrm{ref}} &= \mathbf{p}_{\mathrm{ref}}^{t+\Delta t} - \mathbf{p}_{\mathrm{ref}}^{t}\\ \mathbf{p}_{\mathrm{ref}} &= \Theta\mathbf{p}_{\mathrm{ref}}^{t+\Delta t} + (1-\Theta)\mathbf{p}_{\mathrm{ref}}^{t}\end{aligned}\right\} \Rightarrow \left\{\begin{aligned}\mathbf{p}_{\mathrm{ref}}^{t+\Delta t} - \mathbf{p}_{\mathrm{ref}}^{t} &= \mathbf{dp}_{\mathrm{ref}}\\ \mathbf{p}_{\mathrm{ref}} &= \Theta\mathbf{dp}_{\mathrm{ref}} + \mathbf{p}_{\mathrm{ref}}^{t}\end{aligned}\right. \tag{3.24}$$

If the discretized form (Equation 3.22) of the superset pressure field Equation 2.17 is treated with this replacement, the following expression is obtained:

$$\begin{aligned}&\textit{Pressure field :}\\ &\sum_{\alpha=0}^{\mathrm{Phases}}\left(\frac{1}{\Delta t}\mathbf{A}_{\alpha_{\mathrm{I,II,III}}}^{t+\Theta\Delta t} + \Theta\mathbf{C}_{\alpha}^{t+\Theta\Delta t}\right)\mathbf{dp}_{\mathrm{ref}} =\\ &\sum_{\alpha=0}^{\mathrm{Phases}}\left(\frac{1}{\Delta t}\mathbf{A}_{\alpha_{\mathrm{I,II,III}}}^{t+\Theta\Delta t}\left(\left(\mathbf{p}_{c_{0,\alpha}}^{t+\Delta t} - \mathbf{p}_{c_{0,\mathrm{ref}}}^{t+\Delta t}\right) - \left(\mathbf{p}_{c_{0,\alpha}}^{t} - \mathbf{p}_{c_{0,\mathrm{ref}}}^{t}\right)\right)\right) +\\ &\sum_{\alpha=0}^{\mathrm{Phases}}\left(\mathbf{C}_{\alpha}^{t+\Theta\Delta t}\left(-\mathbf{p}_{\mathrm{ref}}^{t} + \mathbf{p}_{c_{0,\alpha}}^{t+\Theta\Delta t} - \mathbf{p}_{c_{0,\mathrm{ref}}}^{t+\Theta\Delta t}\right) + \mathbf{D}_{\alpha}^{t+\Theta\Delta t} + \mathbf{E}_{\alpha}^{t+\Theta\Delta t}\right)\end{aligned} \tag{3.25}$$

It should be pointed out that still the same set of system matrices is used for this formulation, no changes on the matrices are required. The new formulation has the advantage that the right hand side terms $\mathbf{Cp}_{\mathrm{ref}}$ (the gradient of the pressure field in the old time step) and **D** (the gravitational forces) nearly cancel against each other. If $\mathbf{Cp}_{\mathrm{ref}}$ and **D** are added upon each other *before* adding them upon the right hand side vector, the information that is contained in the other right hand side expressions will not be largely falsified by the “small differences of large numbers” errors any more. Due to the impact on the program structure, the procedure proposed above was not yet adopted to all boundary condition types described in the next chapters.

The matrix expression that is presented in Equation 3.25 can be downsized in order to achieve formulations similar to the ones presented in the prior chapters. As the primary variables are exchanged in the procedure outlined above, the treatment of boundary conditions must be exchanged, too. This is done by subtracting the pressure of the old time level from the desired boundary condition value before actually applying it.

If the saturation fields are evaluated with a formulation that uses the derivatives of the capillary pressure function, it is necessary to take care that the derivative is actually defined and has a value. For the Richards’ approach this can be achieved by using a piecewise linear dependency between capillary pressure and saturation, which covers the whole range of possible suction pressures.

3.1.8 Boundary conditions

If nothing else is explicitly specified, the FEM automatically treats all boundaries as impermeable for fluxes. Most boundary conditions can be treated in a very simple and natural way. It must be pointed out that “boundary” conditions in a FEM formulation are not limited to the actual boundary of the domain. It is a convenient fact that the boundary treatment can be applied to any node of the FEM mesh. The following boundary conditions will be regarded here:

- Fixed value boundary conditions (“Dirichlet”, or first type),
- fixed flux / fixed gradient boundary conditions (“Neumann”, or second type),
- mixed type boundary conditions (“Cauchy”, or third type),
- boundary conditions for special questions.

To assign boundary conditions to nodes it is necessary to specify the nodes that are to be treated. The most obvious way to do this would be the direct specification of the node numbers. Actually this is quite inconvenient, as changes of the mesh (e.g. because the modeller decides that the mesh is not good enough) require re-evaluating which nodes have to be treated. Furthermore the dynamic grid adaptation generates new nodes during simulation. Thus originally closed chains of boundary condition nodes would be broken up by the new nodes, which subsequently would not be treated. Due to this reason a set of geometrically based methods has been derived to assign the boundary conditions to nodes that fulfil geometric criteria. Among them are methods to:

- Set a fixed value to all nodes in a distance ε from a point.
- Set a fixed value to all nodes in a distance ε from an open polygon. For each point of the polygon a specific value can be set. Those values are interpolated to the nodes.
- Set a hydrostatic fixed value to all nodes in a distance ε from an open polygon.
- Set a fixed value to all nodes in a distance ε from a plain, if they are additionally inside a closed four cornered polygon on this plain. For each point of the polygon a specific value can be set, which will be assigned to the node by a bilinear interpolation.
- Set a hydrostatic fixed value to all nodes in a distance ε from a plain that are inside a closed four cornered polygon.
- Distribute a given flux along a line source. The flux is in this case given by the regarded quantity per length and time and will be placed onto all nodes in distance ε from an open polygon. The fluxes will be distributed to the nodes according to the spacing of their projection points on the polygon. Thus an equal flux is gained even if the node spacing is not equidistant.
- Distribute a given flux (u/s) onto all nodes in distance ε from a given point. Furthermore a distance weighting can be specified, so that the applied fluxes can be graded according to their distance. This is very helpful for multiphase flow situations, where single point sources otherwise can result into local oscillations.

All the methods stated above are available for all field equations in model. They are based on vector geometry. It would blast the frame of this work to elaborate on them. Further information should be taken from any textbook on vector geometry.

In the following the linear system of equations will be referred to by using this form:

$$\mathbf{A}\mathbf{x} = \mathbf{b} \tag{3.26}$$

In Equation 3.26 **A** denotes the system matrix, **x** the vector of unknown variables and **b** the right hand side vector.

3.1.8.1 Flux or gradient type boundary conditions

The Neumann (second type) boundary condition is referred to in the literature to classify either fluxes or gradients of the extensive variable over a given boundary. In this work the Neumann boundary condition was interpreted as a flux boundary condition. Other authors preferred to set the projection of the gradient of the regarded variable u onto the boundary normal vector **n** to a certain value c:

$$\mathbf{n}\,\mathrm{grad}\,u = c \tag{3.27}$$

This is quite easy to implement and is meaningful for a FDM formulation, but is less useful for the FEM. In the FEM the special case c = 0 is fulfilled automatically for each boundary of the system if no boundary conditions are explicitly specified. Due to the possibility to use arbitrary shaped grids in the FEM the treatment of gradient type boundary conditions is quite complicated and less practical. Thus the much easier and obvious method to specify fluxes is used.

The flux values are gained by integrating the base functions of the regarded node over the spatial function that defines the flux. For point sources this comes down to the straightforward approach to simply add the desired flux value to the corresponding node. It is important to node that single point sources lead to singularities in the regarded differential equation, thus they should be spread across several nodes instead.

For the fluxes a vector **f** is defined, which is set to the value of the desired flux on the places belonging to nodes with flux boundary conditions (according to the renumbered ordering described in Chapter 3.1.7.1) and to zero everywhere else.

A new right hand side vector $\bar{\mathbf{b}}$ is derived by evaluating:

$$\bar{\mathbf{b}} = \mathbf{b} + \mathbf{f} \tag{3.28}$$

Thus a modified system of linear equations is to be solved:

$$\mathbf{Ax} = \bar{\mathbf{b}} \tag{3.29}$$

3.1.8.2 Fixed value boundary conditions

The first type boundary conditions or so-called Dirichlet type boundary conditions for nodes can simply be regarded as known values in the linear system of equations. Thus they can be eliminated from the system of equations. For the elimination a vector **c** is defined. This vector is set to the value of the desired boundary condition on the places belonging to nodes with boundary conditions (again following the renumbering described in Chapter 3.1.7.1) and to zero everywhere else. It must be noted that the sequence of treating fluxes and fixed boundary conditions is not random. The fixed boundary conditions must be treated last, because otherwise the fluxes might overwrite the previous entries.

A new right hand side vector $\bar{\bar{\mathbf{b}}}$ is derived:

$$\bar{\bar{\mathbf{b}}} = \bar{\mathbf{b}} - \mathbf{Ac} \tag{3.30}$$

Afterwards the row and column entries are set to zero in the matrix for each boundary node. Afterwards the boundary condition is inserted into the system of equations. This should not be done by setting the diagonal to unity and the right hand side value to the desired boundary condition value, as this destroys the numerical properties of the system matrix. Instead the diagonal value should be kept and the right hand side is set to the product of diagonal value and desired boundary condition value. By this procedure, the impact on the properties of the matrix is minimized:

$$\mathbf{A}_{i,n} = \mathbf{A}_{n,i} = 0 \quad \textit{for all i except n} \quad , \quad \overline{\overline{\mathbf{b}}}_{\mathbf{n}} = \mathbf{c}_n \cdot \mathbf{A}_{n,n} \tag{3.31}$$

Finally the following system of equations is to be solved:

$$\mathbf{Ax} = \overline{\overline{\mathbf{b}}} \tag{3.32}$$

3.1.8.3 Mixed type boundary conditions

Third type or Cauchy boundary conditions are defined by evaluating a flux **f** that is depending on a reference value $\mathbf{u}_{\mathrm{ref}}$ of the regarded variable, the actual value **u** of the variable and a scaling factor **c**:

$$\mathbf{f} = \frac{\mathbf{u}_{\mathrm{ref}} - \mathbf{u}}{\mathbf{c}} \tag{3.33}$$

This type of boundary condition is useful to simulate the infiltration of water in rivers and lakes, if a mud layer covers the riverbed or lakebed. In this work the Cauchy boundary condition was not regarded, as the same effect can be achieved by adding a thin layer of finite elements with reduced permeability and a fixed value boundary condition. They show the same behaviour as the Cauchy boundary condition.

3.1.8.4 Special variants of boundary conditions

The simulation of some natural systems requires enhancing the standard boundary conditions. A quite common problem is the definition of “open” boundary conditions. They are supposed to be a gateway for the regarded extensive quantity and should have a minimal impact on the solution inside the domain. This problem has to be solved for the transport of tracers and for the fluid mass in multiphase flow problems. The advective transport of tracers out of the system is automatically treated by the FEM, whereas these boundaries are closed for diffusive fluxes. As the regarded systems are mostly advective, this is not further investigated.

3.1.8.5 Open boundary for multiphase flow problems

The test case by *Buckley & Leverett* (1941, described in Chapter 4.3) is a typical application where special treatment of the saturation equation is required. In Figure 3.11 the saturation field for different times is presented. If the open boundary on the right hand side is not treated adequately, the saturation field starts to oscillate when the front reaches the boundary (solid line in Figure 3.11).

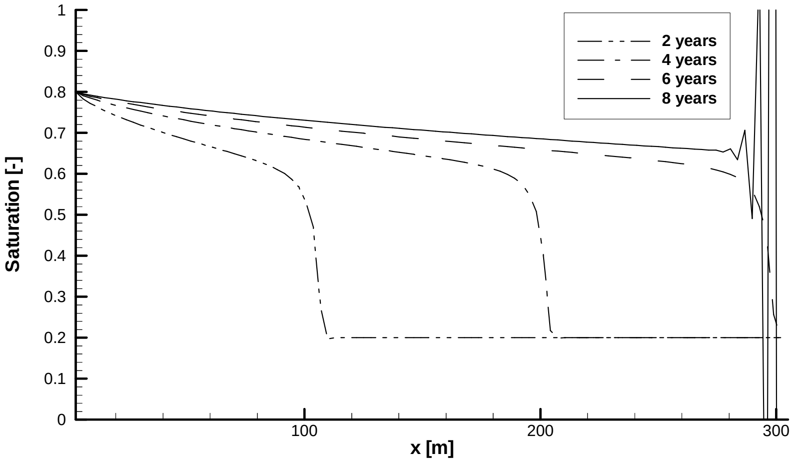


Fig. 3.11: Instable behaviour of the saturation field if no open boundary correction is applied.

Omitting the boundary integrals of the saturation conservation equation causes this behaviour. In the following a procedure to treat them is presented.

If the fluxes are evaluated according to the procedure derived in Chapter 3.1.11 this information can be used to determine the boundary fluxes. For boundary nodes the entries of the flux vector are added to the right hand side of the assembled system of linear equations of the saturation conservation equation. Thus, they are added as fluxes into the saturation equation. This enhances the result for the Buckley-Leverett problem largely (Figure 3.12) in comparison to the uncorrected results (Figure 3.11).

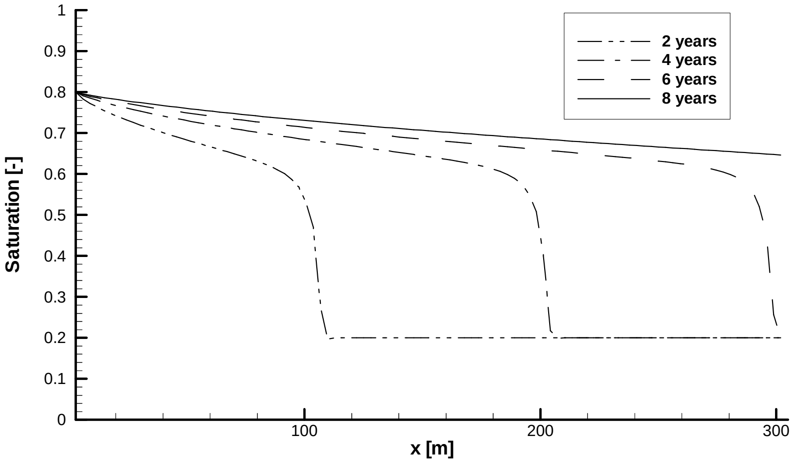


Fig. 3.12: Stable behaviour of the saturation field if open boundary correction is applied.

This type of open boundary condition is also used for the four-fracture system described in Chapter 4.7. Again the performance on the boundaries of the system is reasonable, though some small oscillations are observed near the open boundary. These can be related to the strong curvature of the flow field near the boundary conditions and are dependent on the resolution of the grid, i.e. for smaller grid sizes the oscillations are smaller, too.

Additionally the method presented above was coupled to a conditional Dirichlet boundary condition for the saturation. If the pressure field changes dramatically, so that the previous outflow boundary will become an inflow boundary, the saturation will be treated in a way corresponding to the tracer boundary condition described in the next chapter.

3.1.8.6 Conditional boundary conditions for tracer transport problems

If a Dirichlet type boundary condition for the tracer transport is set on an outflow boundary, the mesh has to be refined to a very high degree. This is necessary, as under those conditions the physical process of backward diffusion has to be resolved numerically. Accordingly the grid must be refined until the grid Peclet number (Equation 2.31) is smaller than unity. This results in very small grid sizes near the boundary conditions and the time steps have to be reduced, too, according to the Neumann criterion (Equation 3.51) and the Courant criterion (Equation 3.50).

Especially for the class of density driven flow problems it is not known a priori which nodes on the boundary can safely be set to a fixed value. For a steady state system the modeller can perform multiple runs of the simulator to evaluate the nodes to be set to a fixed value. Though this is possible, it is time consuming. If the process is transient, it is even impossible.

Summarizing, it seems useful to ignore the boundary condition on nodes that are on an outflow boundary. The fluxes on the boundary nodes are gained from the vector of residuals **r** of the assembled system of linear equations (3.41). For boundary nodes with negative fluxes the Dirichlet boundary condition is switched off (first condition in Equation 3.34). This procedure has successfully been applied (*Kröhn*, 1991) to the saltwater intrusion problem by *Henry* (1960). According to the considerations about the influence of the Peclet number (Equation 2.31) presented above, the method should be enhanced by adding an additional constraint (second condition in Equation 3.34), i.e. the boundary condition should not be switched off if the local grid Peclet number is below unity. Node flux r_i and the element Peclet numbers of the connected elements have to fulfil the combined criterion for the fixed boundary condition to be switched off (Equation 3.34).

$$(r_i < 0) \wedge \left(\min_{E \in \left(\substack{\text{connected} \\ \text{elements}} \right)} \left| Pe_E \right| \right) < 1 \tag{3.34}$$

3.1.8.7 Combined infiltration and free seepage boundary condition

In natural systems the flow regime is quite often too complex to be described by simple flux and fixed pressure boundary conditions. In partially saturated systems it is necessary to treat the groundwater recharge by infiltration, free seepage conditions and a combination of the two (compare Figure 1.1). Those systems can be tackled by the Richards' approximation in the model. To take density driven flow effects into account, the coupling to a transport model is necessary. The type of boundary condition for the transport model can change; hence it is necessary to treat the tracer transport boundary conditions with special methods as already mentioned above.

To treat the seepage condition and the recharge by infiltration, the decision method presented in Figure 3.13 is derived. It must be pointed out that in the following a reference pressure p = 0 Pa is used in the Richards' approximation. Thus, pressures above zero correspond to full saturation, while pressures below zero indicate partial saturation.

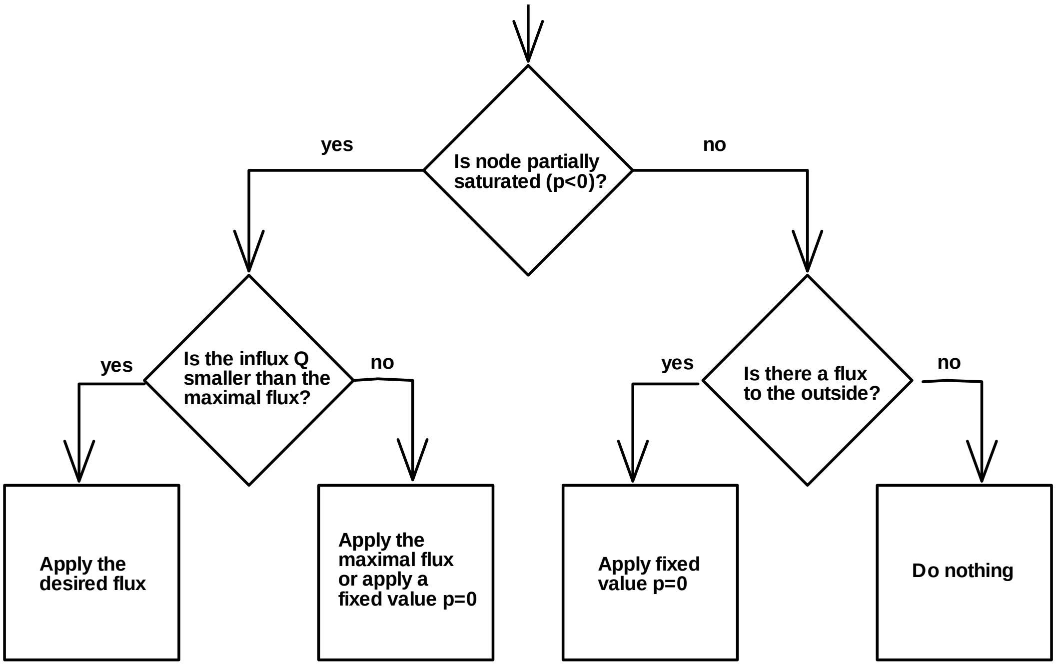


Fig. 3.13: Decision tree to handle combined groundwater recharge and free seepage boundary conditions in the Richards' model. A pressure p = 0 Pa denotes the transition between partial and full saturation in the Richards' model.

The flux into the system on open surfaces must be limited in order to resemble the natural imbibition process. The maximal flux is determined by integrating the volume of the pore space around the regarded node which is not yet saturated and dividing the result by the time step dt:

$$Q_i \overset{!}{\le} \frac{n}{dt} \int \left(\omega_i (1 - S_i) \right) dV \tag{3.35}$$

In this integral ω_i is the set of base functions and S_i the water saturation on node i. If the saturation is evaluated on the old time level, the downward directed fluxes that lead to further desaturation are not included in this expression. Evaluating the applied flux iteratively can circumvent this:

$$Q_i^{iter+1} = Q_i^{iter} + \frac{n}{dt} \int \left(\omega_i \left(1 - S_i^{t+\Delta t} \right) \right) dV < Q_i \tag{3.36}$$

To fulfil the condition in Equation 3.35, either the flux Q_i may be limited to the maximum flux or the type of boundary condition can be switched. Switching the boundary conditions is performed, if the criterion in Equation 3.35 is not fulfilled. The pressure on the regarded node is set to zero, thus the saturation will be equal to unity.

3.1.9 Considerations about grid adaptation

The simulator package RockFlow is capable of performing automatic grid adaptation (“h-adaptation”). This procedure has proven to be very efficient for coupled groundwater flow and transport problems (*Kaiser*, 2001). Applying those methods to the problems considered in this work requires some additional considerations and certain drawbacks are encountered.

First of all the method loses part of its efficiency, if multiple primary variables (i.e. pressure, saturations, concentrations, heat etc.) are regarded, as the requirements of all those influences on the grid refinement are treated in an additive way. Thus, the grid is refined in several areas for all processes, while eventually only one of the processes requires the adaptation. This situation can be enhanced by using the model adaptive procedures presented in Chapter 3.5, as they reduce the computational effort for the processes with low temporal variability.

Another possible drawback is the influence of changing the grid on the equilibrium that is maintained in the nonlinear coupled equations. Due to the changes of the grid, the prior equilibrium is no longer valid and some iterations must be performed to reach a new equilibrium that corresponds with the new grid. This situation is worst, if the error margins for refining and coarsening the grid are very close to each other. Therefore it is useful to do some excessive refinement in order to avoid too many changes of the grid, though this results in an increased number of nodes.

3.1.9.1 Treatment of non-conforming elements

In the current implementation of the RockFlow simulator package, the grid adaptive scheme will produce so-called “irregular nodes”. These nodes are seemingly missing a connection to one side and require special treatment. In Figure 3.14 the interchange of information between the nodes on a finite element mesh is illustrated. Obviously the vertical interchange is blocked for the irregular nodes, thus a significant error is introduced if no special treatment is used.

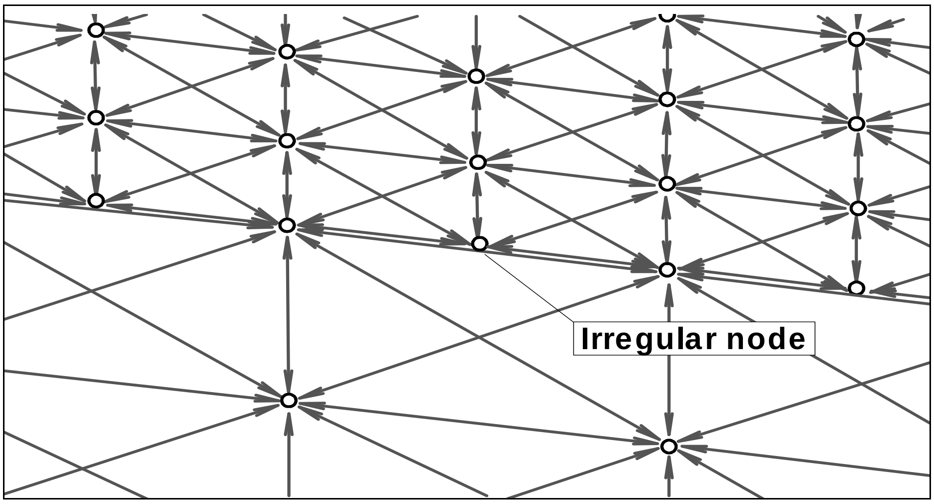


Fig. 3.14: Interrupted information exchange near irregular nodes on a finite element mesh. The double arrows show the information interchange within the grid.

One possibility to circumvent this problem is to eliminate those nodes by using transformation matrices upon the element matrices of all elements connected to an irregular node (*Barlag,* 1997). This procedure eliminates the entries for the irregular node, so that the corresponding row and column in the system of equations are left empty. After solving the system of equations, the value for the irregular node is interpolated from its regular neighbours.

An alternative approach is used in this work. The irregular nodes are directly eliminated from the system of linear equations, i.e. row and column entries of an irregular node are distributed onto the surrounding regular neighbours. This is done by multiplying the row and column entries of each irregular node with 0.5 for irregular nodes on edges (two regular neighbours) or 0.25 for irregular nodes on plains (four regular neighbours). The result is added to the rows and columns of the corresponding regular neighbour nodes (two for irregular nodes on edges, four for irregular nodes on plains).

The pseudo-code for the elimination of an irregular node next to an edge is presented in Equation 3.37. The elimination of an irregular node next to a plain is similar. The number of regarded neighbouring nodes and their weights must be adjusted accordingly. The linear system of equations is represented as **Ax**=**b**. The index irr denotes the entry of the irregular node, while the indices reg1 and reg2 correspond to its regular neighbours.

```
$\text{for}(i = 0; i < n; i++)\{$                                        (3.37)
    $\mathbf{A}_{i,reg1} = \mathbf{A}_{i,reg1} + 0.5 \cdot \mathbf{A}_{i,irr}$
    $\mathbf{A}_{i,reg2} = \mathbf{A}_{i,reg2} + 0.5 \cdot \mathbf{A}_{i,irr}$
    $\mathbf{A}_{i,irr} = 0$ $\}$
$\text{for}(i = 0; i < n; i++)\{$
    $\mathbf{A}_{reg1,i} = \mathbf{A}_{reg1,i} + 0.5 \cdot \mathbf{A}_{irr,i}$
    $\mathbf{A}_{reg2,i} = \mathbf{A}_{reg2,i} + 0.5 \cdot \mathbf{A}_{irr,i}$ $\}$
$\mathbf{b}_{reg1} = \mathbf{b}_{reg1} + 0.5 \cdot \mathbf{b}_{irr}$
$\mathbf{b}_{reg2} = \mathbf{b}_{reg2} + 0.5 \cdot \mathbf{b}_{irr}$
```

Afterwards the row of the irregular node is set to zero and the column is replaced by the interpolation condition. In Equation 3.38 the pseudo-code for the irregular node on an edge is given. It should be pointed out that the interpolation condition is introduced in a way that keeps the diagonal entry for the irregular node unchanged. Otherwise, the estimation of errors in the linear solver would be difficult, because the size of the entries for irregular nodes differs by several orders of magnitude from the other entries.

```
$\text{for}(i = 0; i < irr; i++)\ \mathbf{A}_{irr,i} = 0$                      (3.38)
$\text{for}(i = irr + 1; i < n; i++)\ \mathbf{A}_{irr,i} = 0$
$\mathbf{b}_{irr} = 0$
$\mathbf{A}_{irr,reg1} = \mathbf{A}_{irr,reg2} = -0.5 \cdot \mathbf{A}_{irr,irr}$
```

3.1.9.2 Influence on nonlinear coupled systems

As shown in the prior chapter, the variable values on irregular nodes are forced to a linear interpolation of the regular neighbour’s results. This procedure is well suited for linear systems, but produces some contradictions in nonlinear coupled system. This means, if *one* of the nonlinear coupled system variables fulfils the interpolation condition, the other probably will not. This is true for most of the nonlinear couplings described in Chapter 2.2.

This problem becomes especially obvious in density driven flow. Here, the pressure and the concentration as primary variables in the flow and transport equations are forced to a linear interpolation on the irregular nodes. As the density is nearly a linear function of the concentration, a quadratic increase in hydrostatic pressure is expected for a linear concentration increase over depths. This is a contradiction to the interpolation condition for the pressure on irregular nodes. As the pressures on the irregular nodes are forced to a linear interpolation and the neighbouring free nodes reproduce the correct quadratic increase, erroneous velocities are encountered in the elements near irregular nodes. In Figure 3.15 (left picture) a magnification of the simulation results for one of the set-ups described in Chapter 4.6 is presented. The system is heavily stratified (dashed lines are iso-lines of salinity). Obviously the velocity vectors near the irregular nodes are erroneous in this case.

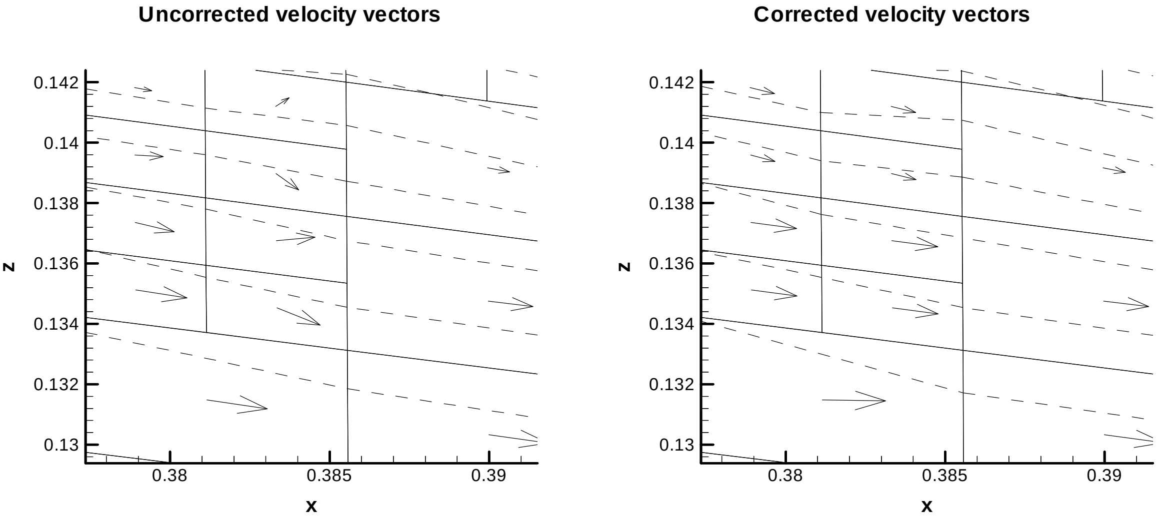


Fig. 3.15: Impact of irregular nodes on the velocity calculation for density driven flow problems. Left picture shows standard solution, right picture corrected solution.

For the special case of density-driven flow a work-around was used. For elements that are connected to irregular nodes, the velocity calculations are performed on the coarser element belonging to the elements with irregular nodes. This enhances the velocity calculation largely (Right picture in Figure 3.15). This approach was used for the post-processing and for the transport calculations.

A more general approach to avoid these problems would be to use another grid-adaptation method, i.e. to use triangles or tetrahedrons to replace the coarse element next to the irregular node. This would convert the irregular node to a regular one and would be suitable for all types of nonlinear couplings. As the grid adaptation is not the primary object of interest in this work but is used as a tool, this will not be deepened here.

3.1.10 Solving of systems of linear equations

After the system of linear equations is assembled, it is the next task to actually solve the system. A large amount of literature is available on solving large systems of linear equations. Only some basics will be outlined here. In the following the regarded systems are referred to by the matrix equation:

$$\mathbf{A}\mathbf{x} = \mathbf{b} \tag{3.39}$$

In this equation **A** denotes the system matrix, **x** the vector of unknowns and **b** the right hand side vector. The systems under consideration fall into the class “large, sparse, diagonal dominant, asymmetric” matrices. “Large” means that the number of unknowns equals the number of nodes in the FEM grid for the formulation regarded here, i.e. they can be in the order of 10^5 at the time being if standard computers are used. For this type of matrices direct solvers like the Gaussian elimination show a weak performance and desire enormous amounts of memory.

Much better performance is achieved with iterative solvers. The iterative solvers can be divided into two groups: 1) stationary solvers like the Jacobi, the Gauss-Seidel or the successive over-relaxation (SOR) solvers. These solvers are largely outperformed for the regarded problem class by the 2) non-stationary solvers. Typical solvers of this class are the different derivatives of the conjugate gradient (CG) method and of the minimal residual (MinRes) method (*Barrett et al.,* 1994). In RockFlow a large variety of solvers is incorporated (*Habbar,* 1995).

Recently multi-grid solvers (*Hackbusch,* 1991) gained much attention in numerical analysis. These solvers are a combination of two (or more) of the methods mentioned above. The main idea is the combination of a “smoothing” solver like SOR with a “sharp” solver like a CG scheme on grids of different spacing. These solvers show excellent performance *even if no initial estimate of the solution is known*. As the regarded problems of this work are transient, the results of the last time step should be very near to the solution of the next time step and thus, a multi-grid solver is not expected to perform so much better that the implementation effort seems justified.

In this work a stabilized Bi-Conjugate Gradient solver (BiCGStab) was used (based on a template by *Barrett et al.*, 1994), as it is suitable for asymmetric matrices and shows a reasonable performance. To enhance the convergence rate of iterative solvers it can be useful to incorporate a preconditioning scheme into the solution process. Simply spoken a pre-conditioner is a matrix (**M** in Figure 3.16) that is multiplied with the system of linear equations in order to enhance the numerical quality of the system of equations. A very simple one is the Jacobi (or point) pre-conditioner, which consists of the reciprocal values of the matrix diagonal. More complex are block methods or multi-grid based schemes. Generally invoking pre-conditioners is a cost trade-off. The computational effort to determine the preconditioning matrix can be significant; on the other hand the possible advantages for the iterative solution are rather large.

In this work an incomplete left-diagonal-upper (iLDU) factorisation was used (implementation by R. Ratke). The preconditioning matrix is left-multiplied to the system of equations. The cost for using the pre-conditioner largely pays off in terms of computational efforts, as it increases the convergence of the BiCGStab solver remarkably. The pseudo-code of the solver algorithm is presented in Figure 3.16.

```
start:
k = 0
$\mathbf{r_0} = \mathbf{b} - \mathbf{Ax}$ // Determine initial residuum
if $(\|\mathbf{r_0}\| < \varepsilon)$ goto end
$\mathbf{p} = \mathbf{r} = \tilde{\mathbf{r}} = \mathbf{M}^{-1}\mathbf{r_0}$ // Apply preconditioning
repeat {
    $k = k + 1$ // Iteration counter
    $\rho_1 = (\tilde{\mathbf{r}}, \mathbf{r}_{k-1})$ // Determine scalar product
    if $(\rho_1 < \varepsilon)$ goto start // Perform an emergency restart
    if $(k \neq 1)$ {
        $\beta = (\rho_1/\rho_2) \cdot (\alpha/\omega)$
        $\mathbf{p} = \mathbf{r} + \beta(\mathbf{p} - \omega\mathbf{v})$
    }
    $\mathbf{v} = \mathbf{M}^{-1}\mathbf{Ap}$ // Apply preconditioning
    $\alpha = \rho_1 / (\tilde{\mathbf{r}}, \mathbf{v})$
    $\mathbf{s} = \mathbf{r} - \alpha\mathbf{v}$
    if $(\|\mathbf{s}\| < \varepsilon)$ { // Convergence is reached, can't go on
        $\mathbf{x} = \mathbf{x} + \alpha\mathbf{p}$ // Final update of solution
        goto start // Perform a restart
    }
    $\mathbf{t} = \mathbf{M}^{-1}\mathbf{As}$ // Apply preconditioning
    $\omega = (\mathbf{t}, \mathbf{s})/(\mathbf{t}, \mathbf{t})$
    $\mathbf{x} = \mathbf{x} + \alpha\mathbf{p} + \omega\mathbf{s}$
    $\mathbf{r} = \mathbf{s} - \omega\mathbf{t}$
    if $(\|\mathbf{r}\| < \varepsilon)$ goto start // Perform a restart because convergence is reached
    $\rho_2 = \rho_1$
}
end:
```

Fig. 3.16: Pseudo-code for solving large sparse systems of linear equations with the BiCGStab method (modified version of a template by *Barrett et al*, 1994).

Within the algorithm an error criterion ε must be set. It is useful to scale an initial error ε_0 criterion with a set of properties derived from the system of linear equations. It is difficult to choose a universal criterion that is suitable for all possible physical processes and situations, good results where achieved with the following criterion:

$$\varepsilon = \varepsilon_0 \max\left(\|\mathbf{Ax_0}\|, \|\mathbf{b}\|, \|\mathbf{Ax_0} - \mathbf{b}\|\right) \tag{3.40}$$

If no scaling for the error criterion and no pre-conditioner is used, the components of the residual vector **r** can be interpreted as erroneous fluxes into or out of the system. These fluxes can be compared to the "physical" fluxes inside the system in order to judge over the introduced error in the mass balance.

If a simple Jacobi pre-conditioner is used (i.e. scaling the system diagonal to unity), the components of the residual vector can be interpreted in a different manner. If the matrix is sufficiently diagonal dominant (it is in most cases), the residual vector components give a rough estimate of the absolute error of the regarded variable.

The standard BiCGStab solver was improved by the possibility to restart the solver after the error criterion is reached. This can be advantageous, as rounding errors tend to accumulate in the solution of CG solvers. If the solver is restarted, the initial guess is already close to the solution, so that only a few further iterations are required to reach convergence again. Thus, a more accurate solution is obtained.

3.1.11 Calculation of fluxes

After assembling the local matrices of the pressure equation to a linear system of equations (Chapter 3.1.7) and before applying the boundary conditions, the pressure field of the last time step is used to determine the residual of the linear system of equations:

$$\mathbf{r} = \mathbf{b} - \mathbf{A}\mathbf{x} \tag{3.41}$$

The residual on each node is equivalent to the total volumetric flux of all phases leaving or entering the system. Together with the fractional flow function (Chapter 4.3) the flux r_{α_i} of each phase α on each node i can be separated:

$$r_{\alpha i} = f_{\alpha i} r_i = \frac{\frac{k_\alpha}{\mu_\alpha} r_i}{\sum_{i=0}^{\text{phases}} \frac{k_i}{\mu_i}} \tag{3.42}$$

After the fluxes have been separated, they can be added as additional sinks to the conservation equation of the fluid saturation on open boundary nodes. It must be pointed out that this procedure is only strictly valid for advection dominated systems that are homogeneous on the boundary conditions.

If capillary pressures dominate the system or the material parameters change on the boundary conditions, an evaluation of the fluxes on each element will produce better results. Thus the *local flux* for each element and each phase is evaluated.

$$\mathbf{r}_\alpha^E = \mathbf{A}_\alpha^E \mathbf{p}_\alpha^E \tag{3.43}$$

In this expression $\mathbf{A}_\alpha^E$ denotes the assemble of all matrices of a single element belonging to phase α, $\mathbf{p}_\alpha^E$ is the vector of pressures of phase α on the nodes belonging to this element and $\mathbf{r}_\alpha^E$ is the resulting vector of fluxes on the nodes of the element.

3.2 Spatial discretization for tracer transport processes in fracture networks

The simulation of tracer transport processes in fracture networks can either be described by volume averaged approaches (e.g. equivalent porous media) or by the direct modelling of discrete fractures and matrix blocks. Multiple numerical schemes are suitable for the modelling of transport processes, e.g. Eulerian ones, like the finite-elements-method (FEM), or the finite-volume-method (FVM) or Lagrangian ones, like the method of characteristics (MOC) or particle tracking (PT). The FEM implementation of tracer transport processes is not the main scope of this work; it is extensively covered in the prior works by *Kröhn* (1991), *Wollrath* (1990) and *Barlag* (1997).

As the number of elements needed to represent the fracture system rises rapidly with the number of intersected fractures, a vast computational effort is necessary to perform a transport simulation, even if only the fractures and not the matrix are considered. Furthermore the transport regime in fractures is often mainly advective, leading to numerical problems and time step restrictions for Eulerian schemes (Chapter 2.1.5.2). Thus, a Lagrangian scheme for the tracer transport is supposed to be effective (*Thorenz*, 2000).

Earlier work (e.g. *Geier et al.* 1995) presented the application of PT to fracture networks. In contrast to PT the MOC requires the uniqueness of characteristics, which is not the case in fracture networks as mixing occurs at the intersections. This work presents an extension to the MOC, which enables the tracing of characteristics across intersections and therefore enhances the applicability of the MOC to fracture network simulations.

3.2.1 Method of characteristics (MOC)

The advective step of the split conservation equation (2.33)

$$\frac{c^{t+\Delta t} - c^{\tilde{t}}}{\Delta t} \approx \left(\frac{\partial c}{\partial t}\right)^{\sim} = -\frac{1}{nS_\alpha}\mathbf{q}_\alpha \cdot \operatorname{grad}(c) = -\mathbf{V} \cdot \operatorname{grad}(c) \tag{3.44}$$

is equivalent to a zero total derivation

$$\frac{dc}{dt} = 0 \tag{3.45}$$

i.e. the concentration remains conserved, while following a characteristic at the fluid's speed (e.g. *Abbott*, 1966). Therefore, the integration of velocity vectors over time gives a scheme to determine the new position of a fluid element at the new time level. At this position (the base point of the characteristic) the concentration is interpolated from the concentration of the surrounding nodes and is transferred back to the starting node. This transfer is possible in both directions (forward or backward in time). In this implementation an upstream tracing is chosen (i.e. characteristics are followed upstream or backward in time from a given node).

A particle tracker would compute the new position of one of the particles in a similar way, if only advection is regarded. The resulting concentration would be gained by dividing the number of particles in a cell through the cell's volume. A large number of particles are necessary to achieve smooth approximations, while in a MOC scheme the number of characteristics to be followed is equal to the number of nodes.

3.2.1.1 Tracing of characteristics

Starting from a mesh node, the velocity vectors are followed from edge to edge in the mesh along the regarded characteristic. The required travel times are summed up until they reach the length of the time step. Finally the tracing will be stopped and the node values are interpolated from the surrounding nodes to the current position. This information will be transferred back to the starting node (Fig. 3.17).

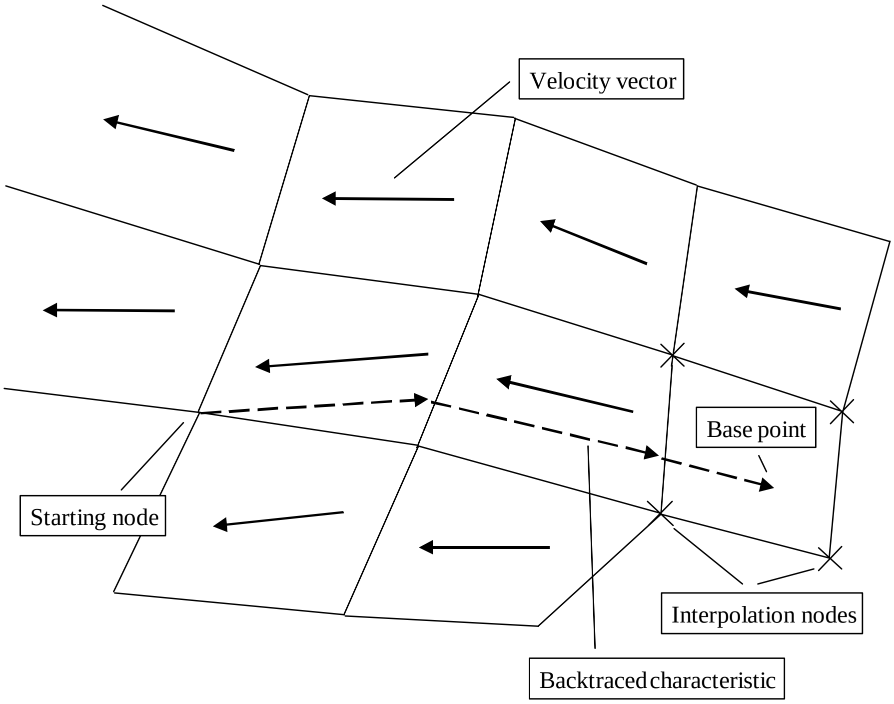


Fig. 3.17: Tracing of characteristics through the mesh. After tracing along a characteristic the information about concentrations are interpolated to the base point and then transferred to the starting node.

Treatment of non-unique characteristics

The standard implementation of the MOC requires the uniqueness of followed characteristics. In fracture networks this uniqueness cannot be guaranteed. The following method extends the capabilities to the tracing of characteristics over intersections.

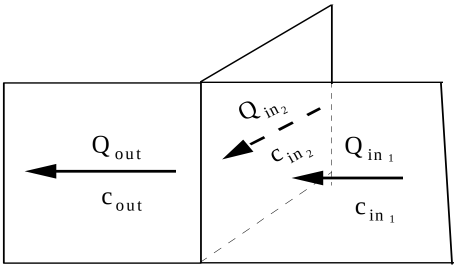


Fig. 3.18: Mixing at a single intersection. The concentration in the downstream branch(es) is determined from concentrations and fluxes of the upstream branches.

Assuming total mixing at intersections, the resulting concentration behind a single intersection (Fig. 3.18) can be computed as follows:

$$c_{out} = \frac{\sum_{j1=1}^{\text{influxes}} \left(Q_{in_{j1}} c_{in_{j1}} \right)}{\sum_{j2=1}^{\text{influxes}} Q_{in_{j2}}} = \sum_{j1=1}^{\text{influxes}} \left(\frac{Q_{in_{j1}}}{\left(\sum_{j2=1}^{\text{influxes}} Q_{in_{j2}} \right)} c_{in_{j1}} \right) = \sum_{j=1}^{\text{influxes}} \left(w_j c_{in_j} \right) \tag{3.46}$$

where c_{out} is the resulting concentration, Q_{in} are volumetric fluxes to the intersection, c_{in} is the corresponding concentration of each flux. It is possible to introduce a local weighting factor w. This factor describes the influence of the upstream branches on the downstream branches and therefore on the starting node. A number of new characteristics (corresponding to the number of influxes) will be followed. Each of them carries its individual weight w. This approach is independent from the number of upstream or downstream branches.

If a number of intersections i is crossed during the tracing, this concept has to be extended. On each intersection encountered, the weight will be distributed again by multiplying the previous weight of the characteristic with the new local weights. The sum of all weights in the system remains always equal to one. The new node concentration c_{node} is gained by following a characteristic backwards in time across several intersections on its way. Thus the resulting concentration c_{node} is obtained as the sum of several results multiplied with the corresponding weight:

$$c_{node} = \sum_{j=1}^{\text{base points}} \left(w_j c_{in_j} \right) = \sum_{j=1}^{\text{base points}} \left(\left(\prod_{i=1}^{\text{crossed intersections}} w_i \right) c_{in_j} \right), \quad \sum_{j=1}^{\text{base points}} w_j \overset{!}{=} 1 \tag{3.47}$$

Finally the resulting concentration is computed from the sum of the concentrations c_j for j base points (Equation 3.47), each of them multiplied with the corresponding weight w_j. Therefore the main task is to determine the resulting weight w_j of a characteristic's base point. This process can be described as the gathering of information (Figure 3.19).

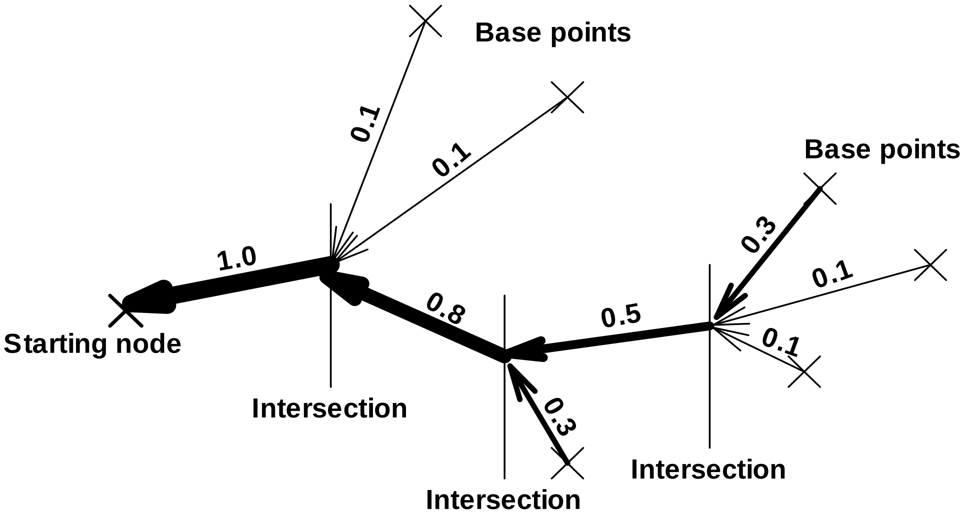


Fig. 3.19: Gathering of information along the paths of multiple characteristics in a fracture network. The numbers on the arrows correspond to their weight in the information stream.

3.2.2 Implementation

The tracing along characteristics is implemented in an object oriented way. So called “agents“ are introduced, which are represented by a set of information and methods (Figure 3.19).

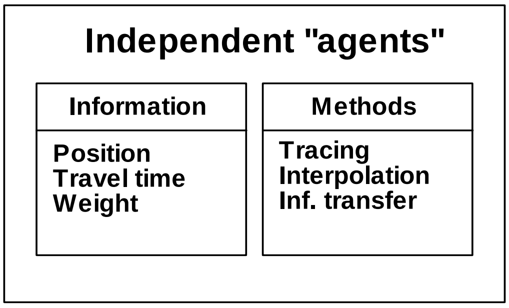


Fig. 3.20: Informational content of the agents. Each agent is treated is independent data object.

At each starting node a single agent is generated with a weight of one. Each agent travels along a characteristic until an intersection with more than one influx is reached. According to the number of influxes, a set of new agents is generated; the old agent dies and passes weight and travel time to the new ones. The weight of each new agent is the result of the old agents weight, multiplied with the new local weight. This procedure is equivalent to the determination of the local weighting w in Equation 3.46.

Each agent travels until his time is used up (i.e. the end of the time step is reached) and then reports its own weight and the interpolated concentration of the tracer back to the starting node, where both values are multiplied and summed up. By this simple scheme the more complex expression of Equation 3.47 has been largely simplified.

This scheme has been implemented recursively and is embedded into a loop over all nodes in the system. After all agents reported back their weights and concentrations, the total sum of weights per node should be equal to one. A small error can be encountered as characteristics with very little weight can optionally be neglected if the weight falls below an adjustable tolerance. To further increase the efficiency of the scheme, it would be useful to change the implementation to a stack-based scheme. Each generated agent would be added to the stack. It would be deleted when giving birth to children or when its travel time is used up. This stack would not be used in standard FIFO (“First in first out”) or LIFO (“Last in first out”) style, but it should be sorted based on the weight of the agents in stack. Thus it could be called HWFO (“Highest weight first out”). This implementation should largely speed up the calculation, if the lower weighted agents are neglected because the user specified minimum return of information is reached already with a lower number of regarded agents.

After the advective step the FEM core is started for the diffusive step. At this point decay and sources or sinks are integrated, too. Examples for the application of the MOC scheme are presented in Chapter 4.5.

3.3 Temporal discretization

As the regarded processes are transient, a discretization in time is necessary. Though it is possible to use the FEM for the discretization, Finite-Differences (FD) are in general preferred for this scope, as they are much more straightforward in formulation and implementation.

All transient variables u are approximated by the expression

$$u = (1-\theta)\cdot u_{(t)} + \theta\cdot u_{(t+\Delta t)} \tag{3.48}$$

with the time weighting (“collocation“) factor θ determining the weighting in time. Though it can be chosen freely, three choices are of special interest:

- θ = 0.0: Forward-Euler or fully explicit scheme.
- θ = 0.5: Crank-Nicolson or central differences scheme.
- θ = 1.0: Backward-Euler or fully implicit scheme.

The time derivatives are computed by evaluating:

$$\frac{\partial u}{\partial t} = \frac{u_{(t+\Delta t)} - u_{(t)}}{\Delta t} \tag{3.49}$$

In different prior works (e.g. *Gärtner,* 1987) it was shown by a Taylor series expansion that the choice of θ = 0.5 leads to the most accurate numerical scheme, as it is of second order accuracy. Experience shows that values of θ between 0.6 and 0.7 are a good trade-off between accuracy and stability.

In this work all physical processes are discretized using the same time step. This is not strictly necessary, in so-called staggered time stepping schemes this is avoided and suitable time steps can be selected for each single process. Due to the tight nonlinear coupling of the manifold of regarded processes, a staggered time stepping is difficult to handle and therefore was not used.

3.3.1 Stability based time stepping criteria

The time stepping for each of the coupled PDEs is limited by additional criteria. These criteria influence the stability and the accuracy of the numerical scheme. The numerical scheme is called “stable”, if small perturbations of the solution (e.g. by rounding due to limited accuracy of the computer) vanish over time rather than grow over all bounds. The actual accuracy of the scheme cannot be determined, as the “true” solution is not known in most cases. Thus the scheme can only be tested against a variety of simple test cases with known solutions. Anyhow if the scheme is *consistent* (i.e. truncation errors tend to zero with temporal and spatial discretization length going to zero), accuracy can be evaluated by reducing the discretization length. It should be pointed out that stability is a weak criterion in comparison to accuracy.

Due to the nonlinear coupling of the PDEs “*it is difficult, if not impossible, to generate results on stability, convergence, and consistency for nonlinear PDE approximations*” (*Lapidus and Pinder*, 1999). Thus, a formal stability analysis cannot be employed to the system of PDEs. Anyhow, under the assumption that the stability of the local linearized scheme is an essential condition for the global scheme, some considerations can be employed for the linearized system.

3.3.1.1 Courant-Friedrichs-Lewy criterion

The advective transport time step in Eulerian formulation is characterized by the Courant-Friedrichs-Lewy (CFL) number, which specifies the way a fluid particle travels during a time step relative to the grid spacing (*Courant et al.,* 1967). Due to accuracy considerations it should be limited, so that the information travelling speed **V** multiplied with the time step Δt is smaller than the grid spacing Δh:

$$\mathrm{CFL} = |\mathbf{V}| \cdot \frac{\Delta t}{\Delta h} \le 1 \tag{3.50}$$

This concept is generally used only for the transport equation. It can be used to determine an upper bound for the time steps in respect to the saturation equation, too. Though this is formally possible, it is generally not useful. Convergence of the nonlinear iterative scheme is mostly only gained for much smaller time steps than those derived from the CFL criterion formulated for the saturation equation.

3.3.1.2 Von Neumann criterion

The time step for diffusive or dispersive transport is limited by the Neumann number Neu. The basic idea is that diffusive fluxes during a time step should be limited, so that they do not change the sign of the concentration gradient:

$$\mathrm{Neu} = \frac{2D\Delta t}{\Delta h^2} < 1 \tag{3.51}$$

The Neumann criterion can also be used to determine a maximal time step for solving the pressure equation for compressible fluids. Again the restrictions for convergence of the nonlinear scheme are much tighter in most cases, so that this additional criterion is not used.

3.3.1.3 Stability considerations

As stated above, it is difficult to specify general bounds for stability based on the CFL or Neumann numbers for nonlinear processes. Even if only the transport equation is regarded, the additional influence by the Peclet number Pe (Equation 2.31) and the time collocation factor (Equation 3.48) complicate the considerations. Furthermore the chosen numerical technique (mass-lumping, upwinding etc.) has an impact on the stability. Only for a very limited range of combinations of the factors mentioned above strict deductions of stability limits are available.

For explicit numerical schemes with a time collocation factor θ smaller than 0.5 the stability criteria must be fulfilled strictly, while numerical schemes with θ bigger than 0.5 can be stable even if the criteria are violated. *Gresho and Sani* (1998) suggested the following criterium for fully explicit (θ = 0) schemes:

$$\begin{aligned} &\mathrm{CFL} \le \mathrm{Pe}/3 \quad \mathit{for} \quad \mathrm{Pe} \le \sqrt{3} \\ &\mathrm{CFL} \le 1/\mathrm{Pe} \quad \mathit{for} \quad \mathrm{Pe} \ge \sqrt{3} \end{aligned} \tag{3.52}$$

These restrictions are not valid for implicit numerical schemes. Anyhow the CFL and Neumann criteria should not be violated largely or in large parts of the domain for implicit schemes due to accuracy considerations.

3.3.2 Heuristic time stepping criteria

In nonlinear coupled systems the CFL and Neumann criteria quite often lead to unreasonable big time steps, for which the nonlinear numerical scheme cannot reach convergence. Thus, it is necessary that additional limitations are derived on the basis of the global performance of the numerical algorithm. Several methods like

- neural networks,
- best performance search,
- genetic algorithms or
- desired value fit.

are suitable for this task. All of the methods mentioned above share the problem that the optimal time step can only be estimated on the basis of the performance of the algorithm for the last time steps. As several influences have an impact on the performance, which are not known a priori (changing boundary conditions, changing grid, transient processes reaching heterogeneities in the systems etc.), a formal optimisation is not feasible. In this work the "Best performance search" and "Desired value" schemes are used.

3.3.2.1 Desired value fit

A very simple scheme is the choice of the time step on the basis of results of the previous time step. It is possible to divide or multiply the time step Δt by a factor m if the number of iterations i in the nonlinear solver reaches critical limits (e.g. *Helmig* 1993, *Simunek et al*, 1999).

$$\begin{aligned} \Delta t_{new} &= \Delta t_{old} \cdot m \quad \mathit{if} \quad i \le i_{min} \\ \Delta t_{new} &= \Delta t_{old} / m \quad \mathit{if} \quad i \ge i_{max} \end{aligned} \tag{3.53}$$

A more flexible approach is a functional coupling between the actual number of iterations i, the desired optimal number of iterations i_{opt} and the time step Δt (*Thorenz*, 1996):

$$\Delta t_{new} = \Delta t_{old} \cdot f\left(i_{opt} / i\right) \tag{3.54}$$

In this approach the functions f can be chosen freely. Several functions were evaluated in the framework of this work; good results can be achieved with a simple exponential function:

$$f(x) = \begin{cases} x^{\alpha_1} & \mathit{if}\ x < 1 \\ x^{\alpha_2} & \mathit{if}\ x > 1 \end{cases} \quad \mathit{with}\ 0 < \alpha_1, \alpha_2 < 1 \tag{3.55}$$

In this formulation the parameters α_1, α_2 specify how aggressive the time stepping scheme reacts on time steps that diverge from the user-desired value for the number of iterations.

The methods described above can also be used to control the changes of primary variables, e.g. to keep changes of the phase saturations or tracer concentrations per time step below a certain value or in a given range (*Forsyth et al.*, 1995).

3.3.2.2 Best performance search

The user of a numerical model is in general interested in a good ratio between advance in simulation time and used real time. This ratio is in the following called the performance P of the scheme:

$$P = \frac{\Delta t_{\mathrm{simulation}}}{\Delta t_{\mathrm{reality}}} \tag{3.56}$$

Many modern computers are multi-user, multi-tasking system. Thus, the total system load can change during the simulation. Therefore, the real time should not be used to determine P, but instead the used CPU time of the simulation task should be regarded. Unfortunately the ANSI standard C library does not contain functions to query the CPU time. Multiple machine dependent implementations for this task are available, but they suffer from portability problems, limited accuracy or overruns of the counters. This situation can be circumvented on single-user systems, where the simulation can be run as the only CPU intensive task, so that the real time is used instead of the CPU time. This situation is expected to be enhanced with the next definition of the C standard, as it is planned to add better support for CPU time queries.

If one would have the possibility to test multiple time steps before actually choosing one, it would be possible to gather the best possible performance for the next time step on the basis of a formal optimisation. As the testing would be very costly, this is not a real choice.

Under the assumption that the optimal time step does not change too much from one time step to another, it is possible to collect the data of the last time steps and derive the estimated optimum from this pool of information. The most obvious way is a curve fitting on the basis of sampled data. For the fitted curve the maximum can easily be determined. This scheme works only if the sample data spreads sufficiently around the optimal time step. If all of the current sample data is quite near the optimal point, this scheme will fail. In this case it is easier to determine the slope of the performance curve for the last values. If the slope is negative, the time step should be reduced, if it is positive, the time step should be increased. Thus the time step is modified by:

$$\Delta t_i = \Delta t_{i-1} \cdot (1+\alpha) \ \textit{with} \ \alpha = \begin{cases} +\alpha_0 \ \textit{if} \ \dfrac{P_{i-1}-P_{i-2}}{\Delta t_{i-1}-\Delta t_{i-2}} \geq +\alpha_1 \\ -\alpha_0 \ \textit{if} \ \dfrac{P_{i-1}-P_{i-2}}{\Delta t_{i-1}-\Delta t_{i-2}} \leq -\alpha_1 \end{cases} \tag{3.57}$$

In this equation the index i denotes the current time step. The parameters a_0 and a_1 are user defined parameters. The performance of this time stepping control scheme is presented in Figure 3.1. It shows the development of the chosen time step during the simulation. The regarded system is based on the Buckley-Leverett test case (Chapter 4.3), with the enhancement of a source boundary condition that varies in time.

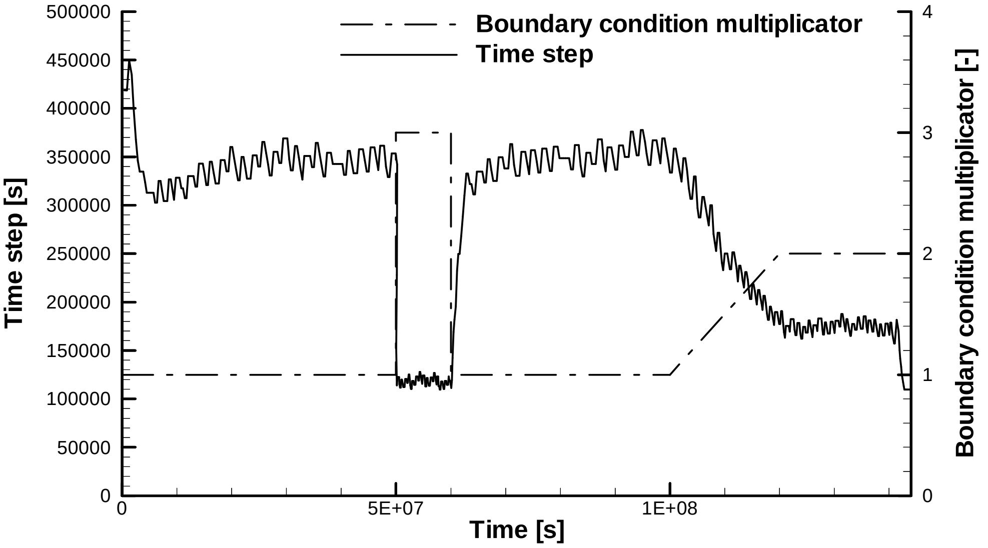


Fig. 3.1: The time step is dynamically adapted due to changing boundary conditions. The dashed-dotted line represents the multiplier for the fluid source, the solid line the automatically chosen time step.

The system quickly adapts the time step (solid line) according to the applied boundary conditions (dashed-dotted line). After $5 \cdot 10^7$ seconds the applied flux changes drastically. As no convergence can be found for the original time step length, the time step length is reduced and the time step is re-evaluated repeatedly until convergence is reached again. After $5 \cdot 10^7$ seconds the boundary flux returns to its initial value and the simulator quickly adapts the time step to the new situation. Near the end of the simulation (between $1 \cdot 10^8$ and $1.2 \cdot 10^8$ seconds) the flux is increased again, following a slope function. Again the time step is adapted adequately by the automatic time stepping scheme.

Evaluating the slope of the performance curve and normalizing it with the values for performance and time step can further enhance this method:

$$\alpha = \left(\alpha_0 \frac{P_{old_1} - P_{old_2}}{\Delta t_{old_1} - \Delta t_{old_2}} \cdot \frac{\Delta t_{old_1} + \Delta t_{old_2}}{P_{old_1} + P_{old_2}} \right)^{\alpha_1} \tag{3.58}$$
$$|\alpha| \le \alpha_2$$
$$\Delta t_{new} = \Delta t_{old_1} \cdot (1+\alpha)$$

In this scheme α_1, α_2 and α_3 are used as control parameters. The scheme acts smoother in varying the time step, but the differences in performance in comparison to the simple scheme are not remarkable.

3.4 Linearization

3.4.1 Iteration schemes

As already described in the prior chapters, a set of nonlinearly coupled PDEs is regarded. In order to resolve those nonlinearities, multiple schemes are feasible. The most prominent ones among them are the Newton method and the fixed point or Picard iteration. In this work a Picard type iteration was used, the convergence rate was enhanced by a simple linear predictor scheme:

$$\mathbf{u}_{predicted}^{t+\Delta t} = \mathbf{u}^{t} + s \cdot \frac{\Delta t}{\Delta t_{old}} \left(\mathbf{u}^{t} - \mathbf{u}^{t-\Delta t} \right) \tag{3.59}$$

In this equation **u** denotes the vector of node values of the regarded quantity, s is a scaling value between zero and unity. This expression is evaluated at the beginning of a time step for all variables. As the expression is node independent, it is not necessary to store an additional third time level of the node values. The ratio of current and old time step length is necessary to scale the predictor appropriately if the time step length changes.

For the Newton method a system of coupled nonlinear equations is set up. This system is obtained from the FEM formulation for all nonlinear coupled primary variables, i.e. the size of the equation system is equal to the number of nodes times the number of primary variables. Thus, the following equation must be solved:

$$\mathbf{f}(\mathbf{x}) = \mathbf{0} \tag{3.60}$$

In this expression **x** is a vector of nonlinear functions. The standard procedure requires setting up the Jacobian matrix $\mathbf{f}'(\mathbf{x})$ for this system. The Jacobian is obtained by evaluating the partial derivative of each column in respect to the corresponding variable:

$$\mathbf{f}'(\mathbf{x}) = \left(\partial_1 \mathbf{f}(\mathbf{x}), \ldots, \partial_n \mathbf{f}(\mathbf{x}) \right) \tag{3.61}$$

This procedure can be difficult, if in nonlinearly coupled systems the analytical derivatives are not available. In this case, the derivatives must be obtained by numerically approximating them. This will reduce the convergence speed and increase the numerical effort. The resulting numerical scheme is referred to as “Quasi-Newton” methods. The method after *Broyden* (1967) is a very successful member in this class of schemes.

After setting up the Jacobian, it is necessary to solve the linearized system of equations:

$$f'(\mathbf{x}_k) \cdot \mathbf{s}_k = -f(\mathbf{x}_k) \tag{3.62}$$

The iteration vector is updated by setting:

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{s}_k \tag{3.63}$$

Afterwards the procedure is started again with the recalculation of the Jacobian. If the coupling function are weakly nonlinear, it is not necessary to do this in each iterative step. Convergence of Newton’s scheme is only possible if the initial guess is already sufficiently near to the solution and if the Jacobian is non-singular. Thus, it is necessary that the coupling functions (Chapter 2.2) have non-singular derivatives.

To increase the convergence radius of the Newton scheme it is possible to use a Trust-Region scheme or Line-Search algorithms. Trust-Region schemes evaluate the size of the correction step and scale it appropriately to keep it in certain bounds. For Line-Search algorithms the residual of $f(\mathbf{x})_{k+1}$ is related to the residual of $f(\mathbf{x})_k$. If necessary, the correction vector is scaled until the new residual is smaller than the prior one.

If the determination of the Jacobian is difficult or impossible, so that a finite differences approach must be used, it should be considered to use Broyden’s approach instead. After initially estimating $f'(\mathbf{x})$ by finite differences it is automatically determined by the iterative procedure.

A more simplistic approach is the Picard iteration scheme. It involves solving multiple systems of linear equations subsequently, where each system has a number of unknowns that is equal to the number of nodes. After solving one of the systems, the matrix of the next system will be updated corresponding to the new values. Thus, instead of solving a single large system of linear equations like in the Newton scheme, multiple smaller systems are solved. This reduces the effort for solving the system of linear equations, as most linear solvers scale nonlinear with the number of nodes. Another advantage is the larger convergence radius compared to the Newton scheme. The drawback is that the convergence rate of the Picard scheme is lower than the one of the Newton scheme near the correct solution.

A Picard scheme was chosen because of the ease in adding physical processes into the iterative scheme, and because of the straightforward method of implementation. Although a Newton scheme often yields a better convergence rate, this advantage is only partly valid if the analytical derivatives of the coefficients are not known. If the derivatives must be computed during runtime by a finite difference method, the convergence suffers. *Lehmann et al.*

(1998) have shown that the Picard and Newton schemes demonstrate similar performance for the class of problems regarded in this work.

As the Picard scheme shows a larger convergence radius than the Newton scheme, it is supposed to use a combined scheme for further developments, i.e. to first perform one or a few Picard iteration steps and to switch afterwards to the locally faster converging Newton scheme.

3.4.2 Relaxation

The methods presented above share the problem that the computed correction vector or the values for the next iteration are not always optimally scaled. Sometimes a larger correction step would be possible, in other situations a smaller one would be preferable to increase the convergence rate. A so-called relaxation parameter λ can be introduced to scale the correction vector **c** before using it to update the vector of values **u** for the regarded quantity:

$$\mathbf{u}_{\text{iter}+1} = \mathbf{u}_{\text{iter}} - \lambda c \tag{3.64}$$

This enables the free scaling of the correction vector. The scaling factor λ should be chosen between zero and unity.

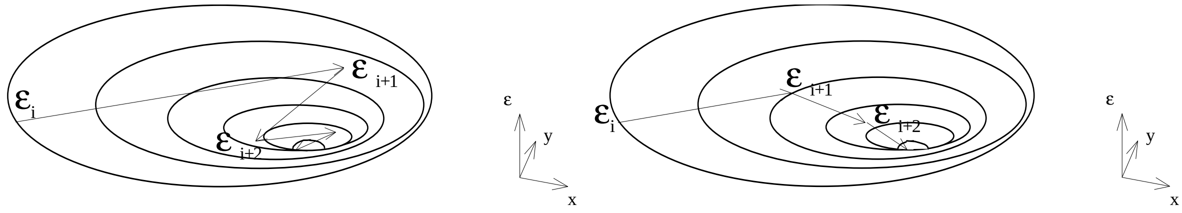


Fig. 3.2: Iterative behaviour for a system with two unknowns (x and y) and shallow outer slope of the error ε. The left picture shows the “unrelaxed” behaviour of the iterative process, the right picture “relaxed” behaviour.

To illustrate the problem a problem with only two unknown variables x and y is regarded. A simple iterative scheme is used (fastest decline), for which Figure 3.2 presents a fictitious plot of the error cone ε for a system with a shallow outer slope. The initial correction steps are chosen too big (left picture in Figure 3.2), so that a large number of iterative steps are needed. If each of the correction steps would be halved (right graphic in Figure 3.2) the convergence rate would be significantly better.

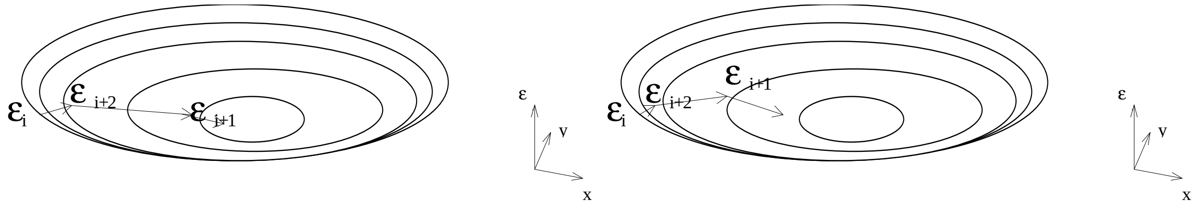


Fig. 3.3: Iterative behaviour for a system with two unknowns (x and y) and steep outer slope of the error ε. The left picture shows the “unrelaxed” behaviour of the iterative process, the right picture “relaxed” behaviour.

For other situations it would be unreasonable to reduce the computed correction. In the left picture of Figure 3.3 the unrelaxed behaviour for a steep outer slope is presented. In this case the relaxation (right picture of Figure 3.3) largely increases the number of needed iterations.

If a Newton iteration scheme is used, the so-called “line search” procedure can be applied to solve this problem. For the line search algorithm the correction vector will be reduced until it is guaranteed that the total system error is reduced. This procedure is not applicable for the Picard iteration scheme used in this work, thus another method is proposed.

The comparison of the two cases shows that a dynamic relaxation parameter would be helpful to increase the convergence. This task was solved with line search algorithms for systems with a modest nonlinear coupling between the equations. If the coupling is depending on multiple variables and can even change during the simulation a mathematical approach is difficult to handle. Thus a heuristic approach has been developed to keep oscillations of the iteration bounded. The scheme uses the directions of the entries in the correction vectors and compares them to the previous corrections. If to many of them change their direction, the relaxation factor will be reduced so that oscillations are depressed. Weighting the vector entries with their size can enhance this simple scheme.

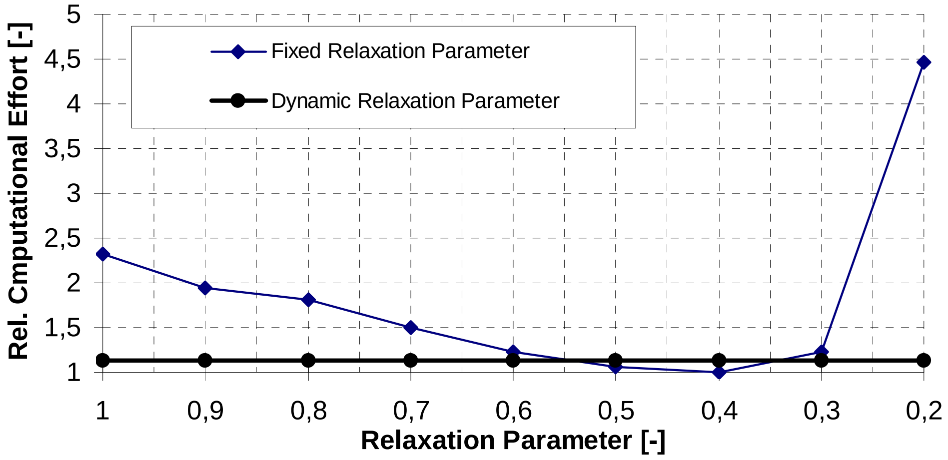


Fig. 3.4: Computational effort for different fixed and an automatically chosen dynamic relaxation parameter. Numerous trials are needed to find a fixed relaxation parameter that is slightly superior to the automatically chosen one.

This scheme was tested for multiple test cases. Figure 3.4 presents the computational effort for hand trimmed static relaxation and automatically chosen dynamic relaxation. The dynamically chosen relaxation parameters show a performance that is comparable to a hand tuned fixed relaxation parameter. It must be pointed out that hand tuning of the relaxation parameter is a very time consuming operation, as many simulation runs are necessary. Furthermore, this operation must be repeated if the set-up of the simulation is changed. The results for the dynamic relaxation parameter were obtained from a single run of the simulator and required no hand-tuning.

3.4.3 Accuracy evaluation

It is necessary to define error criteria to determine sufficient convergence in the nonlinear iteration. As multiple variables are used to describe the physical processes, it is necessary to track all of them for changes in the iterative process. On the one hand the primary variables (i.e. pressure, concentrations, saturations) can be tracked, on the other derived variables as the inner fluxes can provide an interesting view into the inner errors of the not yet converged system.

For the primary variable V the following criteria C, which are only based on the results of the new time level, are evaluated. The numbering is conforming to the one in RockFlow.

$$\text{C I:}\quad \varepsilon_P = \max_{1\le i\le n}\left|\operatorname{abs}\left(V_i^{old} - V_i^{new}\right)\right| \tag{3.65}$$

$$\text{C II:}\quad \varepsilon_P = \max_{1\le i\le n}\left|2\cdot\operatorname{abs}\left(\frac{V_i^{old} - V_i^{new}}{V_i^{old} + V_i^{new}}\right)\right| \tag{3.66}$$

$$\text{C III:}\ \varepsilon_P = \frac{\max\limits_{1\le i\le n}\left|\operatorname{abs}\left(V_i^{old} - V_i^{new}\right)\right|}{\max\limits_{1\le i\le n}\left|\operatorname{abs}\left(V_i^{new}\right)\right|} \tag{3.67}$$

$$\text{C IV:}\ \varepsilon_P = \frac{\max\limits_{1\le i\le n}\left|\operatorname{abs}\left(V_i^{old} - V_i^{new}\right)\right|}{\max\limits_{1\le i\le n}\left|\operatorname{abs}\left(V_i^{new}\right)\right| - \min\limits_{1\le i\le n}\left|\operatorname{abs}\left(V_i^{new}\right)\right|} \tag{3.68}$$

$$\text{C VII:}\qquad \varepsilon_P = \left[\sum_{i=1}^{n}\left(V_i^{t+\Delta t^{old}} - V_i^{t+\Delta t^{new}}\right)^2\right]^{1/2} \tag{3.69}$$

All of the above methods share the problems that the errors in the nonlinear iteration will accumulate if the time step is reduced. I.e. halving the time step can double the error in the nonlinear iteration, as the same error can occur twice in the same time span. Thus a set of criteria is developed, each of them takes into account the changes during a time step and therefore works on the two time levels t and t+Δt.

$$\text{C V:}\quad \varepsilon_P = \max_{1\le i\le n}\left|\text{abs}\left(\frac{V_i^{t+\Delta t^{old}} - V_i^{t+\Delta t^{new}}}{V_i^{t+\Delta t^{new}} - V_i^{t}}\right)\right| \tag{3.70}$$

$$\text{C VI:}\ \varepsilon_P = \frac{\max\limits_{1\le i\le n}\left|\text{abs}\left(V_i^{t+\Delta t^{old}} - V_i^{t+\Delta t^{new}}\right)\right|}{\max\limits_{1\le i\le n}\left|\text{abs}\left(V_i^{t+\Delta t^{new}} - V_i^{t}\right)\right|} \tag{3.71}$$

$$\text{C VIII:}\ \varepsilon_P = \frac{\left[\sum\limits_{i=1}^{n}\left(V_i^{t+\Delta t^{old}} - V_i^{t+\Delta t^{new}}\right)^2\right]^{1/2}}{\left[\sum\limits_{i=1}^{n}\left(V_i^{t} - V_i^{t+\Delta t^{new}}\right)^2\right]^{1/2}} \tag{3.72}$$

It is impossible to give a general recommendation for the type of error criterion to be chosen. For highly dynamic nonlinear processes the criteria C VI and C VIII produce very stable results, which do not have the tendency of error accumulation. For processes that are already near a steady state the criteria of the first group can be more suitable, as their denominator does not reach zero for steady state. The criteria C VI and C VIII do also work for systems that are near or in steady state, but they are less effective, as at least two iterations are performed in each time step. Furthermore, the limited accuracy of the linear solver should be considered, i.e. the denominator in C VI and C VIII should be limited to values above a user definable critical value, otherwise the system will spent useless iteration time in the nonlinear solver.

Another possibility to verify the accuracy of the iteration process is the evaluation of the inner fluxes, i.e. the residuum of the pressure equation if the last pressure iteration vector is applied upon the assembled system of equations for the next iteration. The values of the residual vector correspond to erroneous fluxes into and out of the system.

3.5 Model adaptive methods

The coupled systems which are regarded in this work can be described by a set of nonlinear coupled PDEs as described in Chapter 2.1:

- A single pressure equation (p unknown).
- Multiple saturation equations (S_α unknown).
- Multiple transport equations ($c_{\alpha,\beta}$ unknown).

These PDEs result in large systems of linear equations, as described in Chapter 3.1. Due to the nonlinear coupling (Equation 3.73), these systems of linear equations have to be rebuilt multiple times in each time step, an extremely costly procedure.

$$\begin{aligned} & k_{r_\alpha} = f\left(S_0,\ldots,S_{\text{phases}-1},\mathbf{V}_\alpha\right),\ p_{c_{0,\alpha}} = f\left(S_0,\ldots,S_{\text{phases}-1},\right) \\ & \mu_\alpha = f(c_{\alpha,\beta},p_\alpha),\ \rho_\alpha = f(c_{\alpha,\beta},p_\alpha) \end{aligned} \tag{3.73}$$

Setting up the systems of linear equations is expensive compared to solving them. This is due to the size of the PDEs on the one hand and the advances in solver technology (BiCG solvers, preconditioners, multi grid methods etc.) on the other. The following approach has the goal to reduce the time to set up those equation systems.

3.5.1 A priori adaptation of the physical model

The first step is to determine a “holistic” physical model, which captures all effects the modeller wants to take into account. This is a very important step, as already at this point the physical system can be over- or undersized. In the following some simplified formulations of the discretized field equations (Chapter 3.1.7.2) will be outlined, which demonstrate the capability to adapt the model to simplified physical problems.

3.5.1.1 Saturated single-phase flow of an incompressible fluid

This is the simplest case of groundwater flow simulation. Only the flux matrix **C** for a single-phase has to be multiplied with the pressure vector. On the right hand side the known gravity term **D** and sinks/sources **E** are added. It is important to note that this formulation is not stable for a time collocation factor $\Theta \le 0.5$. It is recommended to use a fully implicit scheme with $\Theta = 1$.

$$\begin{aligned} & \Theta\mathbf{C}_0^{t+\Theta\Delta t}\mathbf{p}_0^{t+\Delta t} = \\ & -\left(1-\Theta\right)\mathbf{C}_0^{t+\Theta\Delta t}\mathbf{p}_0^{t} + \mathbf{D}_\alpha^{t+\Theta\Delta t} + \mathbf{E}_0^{t+\Theta\Delta t} \end{aligned} \tag{3.74}$$

3.5.1.2 Saturated single-phase flow of a compressible fluid

For this formulation the storage matrix **A** is introduced. In this case it describes the fluid compressibility. The introduction of the storage term increases the stability of the solution, so that the most accurate time collocation factor $\Theta = 0.5$ can be used.

$$\left(\frac{1}{\Delta t}\mathbf{A}_{0_{I}}^{t+\Theta\Delta t} + \Theta\mathbf{C}_{0}^{t+\Theta\Delta t}\right)\mathbf{p}_{0}^{t+\Delta t} = \\ \frac{1}{\Delta t}\mathbf{A}_{0_{I}}^{t+\Theta\Delta t}\mathbf{p}_{0}^{t} - (1-\Theta)\mathbf{C}_{0}^{t+\Theta\Delta t}\mathbf{p}_{0}^{t} + \mathbf{D}_{0}^{t+\Theta\Delta t} + \mathbf{E}_{0}^{t+\Theta\Delta t} \tag{3.75}$$

3.5.1.3 Saturated single-phase flow in an aquifer with storativity

Quite similar is the formulations for the flow of an incompressible fluid in an aquifer with storativity. In this case the compressibility of liquid, matrix and residual gas saturation is lumped into the soil storativity within matrix **A**.

$$\left(\frac{1}{\Delta t}\mathbf{A}_{0_{II}}^{t+\Theta\Delta t} + \Theta\mathbf{C}_{0}^{t+\Theta\Delta t}\right)\mathbf{p}_{0}^{t+\Delta t} = \\ \frac{1}{\Delta t}\mathbf{A}_{0_{II}}^{t+\Theta\Delta t}\mathbf{p}_{0}^{t} - (1-\Theta)\mathbf{C}_{0}^{t+\Theta\Delta t}\mathbf{p}_{0}^{t} + \mathbf{D}_{0}^{t+\Theta\Delta t} + \mathbf{E}_{0}^{t+\Theta\Delta t} \tag{3.76}$$

3.5.1.4 Unsaturated single-phase flow in an aquifer with storativity

Unsaturated flow in an aquifer can be described by multiple formulations of the continuity equations. The first is the classical *Richards'* formulation, in which the storage matrix **A** additionally contains the relationship between saturation and capillary pressure:

Pressure formulation

$$\left(\frac{1}{\Delta t}\mathbf{A}_{0_{II,III}}^{t+\Theta\Delta t} + \Theta\mathbf{C}_{0}^{t+\Theta\Delta t}\right)\mathbf{p}_{0}^{t+\Delta t} = \\ \frac{1}{\Delta t}\mathbf{A}_{0_{II,III}}^{t+\Theta\Delta t}\mathbf{p}_{0}^{t} - (1-\Theta)\mathbf{C}_{0}^{t+\Theta\Delta t}\mathbf{p}_{0}^{t} + \mathbf{D}_{0}^{t+\Theta\Delta t} + \mathbf{E}_{0}^{t+\Theta\Delta t} \tag{3.77}$$

This formulation *requires* that the derivatives of the capillary pressure functions are defined over all of the possible suction pressures. If this is not the case, mass losses will occur in this formulation. For Finite-Difference formulations these mass losses even occur if the capillary pressure function is well behaved. This was the main reason for the development of the schemes presented in the following.

An alternative formulation was proposed by *Celia et al.* (1990). It does not use the derivative of the capillary pressure–saturation relation in the mass storage term, but describes the mass storage due to saturation changes directly by using the saturation mass matrix B to describe changes of the saturation. Thus, it is simply the standard conservation equation for a single fluid under partially saturated conditions.

$$\textit{Mixed formulation:} \tag{3.78}$$

$$\left(\frac{1}{\Delta t}\mathbf{A}_{0_{II}}^{t+\Theta\Delta t}+\Theta\mathbf{C}_{0}^{t+\Theta\Delta t}\right)\mathbf{p}_{0}^{t+\Delta t}=$$

$$\frac{1}{\Delta t}\mathbf{A}_{0_{II}}^{t+\Theta\Delta t}\mathbf{p}_{0}^{t}-\frac{1}{\Delta t}\mathbf{B}^{E}\left(\mathbf{S}_{0}^{t+\Delta t}-\mathbf{S}_{0}^{t}\right)-(1-\Theta)\mathbf{C}_{0}^{t+\Theta\Delta t}\mathbf{p}_{0}^{t}+\mathbf{D}_{\alpha}^{t+\Theta\Delta t}+\mathbf{E}_{0}^{t+\Theta\Delta t}$$

This formulation will show no mass losses, even if the capillary pressure function is not defined or the gradient is equal to zero or infinity for a certain suction pressure. But in general this formulation is not first choice, as the convergence in the nonlinear iteration scheme is poorer. This can be explained by the fact that changes of the saturation no longer are handled implicitly in the system of equations, but are handled as a right hand side parameter.

For very dry situations the saturation formulation of the Richards' equation is well suited:

$$\textit{Saturation formulation:} \tag{3.79}$$

$$\frac{1}{\Delta t}\mathbf{B}\,\mathbf{S}_{0}^{t+\Delta t}=$$

$$\frac{1}{\Delta t}\mathbf{A}_{0_{II}}^{t+\Theta\Delta t}\left(\mathbf{p}_{0}^{t}-\mathbf{p}_{0}^{t+\Delta t}\right)+\frac{1}{\Delta t}\mathbf{B}\,\mathbf{S}_{0}^{t}-\mathbf{C}_{0}^{t+\Theta\Delta t}\mathbf{p}_{0}^{t+\Theta\Delta t}+\mathbf{D}_{\alpha}^{t+\Theta\Delta t}+\mathbf{E}_{0}^{t+\Theta\Delta t}$$

It cannot be used in saturated areas of the system. To avoid this problem the primary variable switching technique (PVST) was proposed by *Forsyth et al.* (1995). In this approach, the system parts with a very low saturation are handled with the saturation formulation, while the rest of the system is treated by one of the other formulations. This approach is not implemented yet, as it would require a major rework of the assemble strategy for the system of linear equations. *Diersch and Perrochet* (1999) have shown promising results for the performance of this scheme.

3.5.1.5 Multiphase flow

For multiphase flow problems a common pressure field for all phases is computed. Thus, the discretized pressure field equation for multiphase flow is very common to the general superset equation (3.22). Differences arise only in the storage terms of matrix **A**:

$$\begin{aligned}
&\text{Pressure field :} && (3.80)\\
&\sum_{\alpha=0}^{\text{phases}-1}\left(\frac{1}{\Delta t}\mathbf{A}_{\alpha_I}^{t+\Theta\Delta t}+\Theta\mathbf{C}_{\alpha}^{t+\Theta\Delta t}\right)\mathbf{p}_{ref}^{t+\Delta t}=\\
&\sum_{\alpha=0}^{\text{phases}-1}\left(\frac{1}{\Delta t}\mathbf{A}_{\alpha_I}^{t+\Theta\Delta t}-(1-\Theta)\mathbf{C}_{\alpha}^{t+\Theta\Delta t}\right)\mathbf{p}_{ref}^{t}+\\
&\sum_{\alpha=0}^{\text{phases}-1}\left(\frac{1}{\Delta t}\mathbf{A}_{\alpha_I}^{t+\Theta\Delta t}\left(\left(\mathbf{p}_{c_{0,\alpha}}^{t+\Delta t}-\mathbf{p}_{c_{0,ref}}^{t+\Delta t}\right)-\left(\mathbf{p}_{c_{0,\alpha}}^{t}-\mathbf{p}_{c_{0,ref}}^{t}\right)\right)\right)+\\
&\sum_{\alpha=0}^{\text{phases}-1}\left(\mathbf{C}_{\alpha}^{t+\Theta\Delta}\left(\mathbf{p}_{c_{0,\alpha}}^{t+\Theta\Delta t}-\mathbf{p}_{c_{0,ref}}^{t+\Theta\Delta t}\right)\right)+\\
&\sum_{\alpha=0}^{\text{phases}-1}\left(\mathbf{D}_{\alpha}^{t+\Theta\Delta t}+\mathbf{E}_{\alpha}^{t+\Theta\Delta t}\right)
\end{aligned}$$

For multiphase flow in heterogeneous systems the choice of the reference phase can be crucial. As described by the *extended pressure condition* by *Duijn et al.* (1995), the phase pressures are not necessarily continuous over material boundaries. Continuity is not given, if the saturation of a phase drops below the residual saturation. In this situation the capillary pressures are no longer defined. For many natural soil systems it can be assumed that water is the most-wetting fluid (this not true for materials based on organic components, e.g. coal) and that the water saturation never drops below the residual saturation if no phase transitions (i.e. drying) take place. Thus, the water pressure can be used as reference pressure in most natural heterogeneous systems. As the influence of the capillary pressures on both sides of a material border shall correctly be treated in this situation, it is necessary to evaluate the node capillary pressures element-by-element. This means that the influence of the capillary pressures on the pressure field is correctly evaluated according to the materials assigned to the elements at each side of the material border. If a node-by-node approach is used instead, the capillary pressures on the material border cannot be described adequately because of the jump of the material properties.

As already outlined in Chapter 3.1.7.2, additionally phases-1 saturation field equations must be solved after solving the pressure field equation:

$$\begin{aligned}
&\textit{Saturation fields :} && (3.81)\\
&\frac{1}{\Delta t}\mathbf{B}\,\mathbf{S}_{\alpha}^{t+\Delta t}=\frac{1}{\Delta t}\mathbf{B}\,\mathbf{S}_{\alpha}^{t}\\
&-\frac{1}{\Delta t}\mathbf{A}_{\alpha_I}^{t+\Delta t}\left(\left(\mathbf{p}_{ref}^{t+\Delta t}-\mathbf{p}_{ref}^{t}\right)-\left(\mathbf{p}_{c_{0,\alpha}}^{t+\Delta t}-\mathbf{p}_{c_{0,ref}}^{t+\Delta t}\right)+\left(\mathbf{p}_{c_{0,\alpha}}^{t}-\mathbf{p}_{c_{0,ref}}^{t}\right)\right)+\\
&-\mathbf{C}_{\alpha}^{t+\Delta t}\left(\mathbf{p}_{ref}^{t+\Theta\Delta t}-\mathbf{p}_{c_{0,\alpha}}^{t+\Theta\Delta t}+\mathbf{p}_{c_{0,ref}}^{t+\Theta\Delta t}\right)+\\
&\mathbf{D}_{\alpha}^{t+\Delta t}+\mathbf{E}_{\alpha}^{t+\Delta t}
\end{aligned}$$

For heterogeneous systems the saturation can be discontinuous across the material boundaries. This behaviour should be taken care for by assigning multiple saturations to a single node on the boundary, each belonging to one of the surrounding materials. Additionally, constraints for the saturations must be extracted from the capillary pressures of the surrounding materials and incorporated into the system of equations. This procedure is not implemented yet.

The formulation presented above is suited for advection dominated systems, but can suffer from convergence problems for systems that are governed by capillary forces. This is due to the explicit handling of the capillary pressures in the saturation field equations (Equation 3.23). It is supposed to switch to another formulation by using the following substitution:

$$\mathrm{grad}\ p_\alpha = \mathrm{grad}\left(p_{ref} - p_{c_{0,\alpha}} + p_{c_{0,ref}}\right) = \mathrm{grad}\ p_{ref} - \frac{\partial p_{c_{0,\alpha}}}{\partial S_\alpha}\mathrm{grad}\ S_\alpha + \frac{\partial p_{c_{0,ref}}}{\partial S_{ref}}\mathrm{grad}\ S_{ref} \tag{3.82}$$

It was assumed that this substitution is constant over space within an element. This leads to the following expression to describe the saturation field:

$$\begin{aligned}
&\mathit{Saturation\ \ fields:} \\
&\sum_{E=0}^{elements-1}\left(\frac{1}{\Delta t}\mathbf{B}^E - \Theta\left(\frac{\partial p_{c_{0,\alpha}}}{\partial S}\right)^E \mathbf{C}_\alpha^{E^{t+\Theta\Delta t}}\right)\mathbf{S}_\alpha^{t+\Delta t} = \\
&\sum_{E=0}^{elements-1}\left(\frac{1}{\Delta t}\mathbf{B}^E + (1-\Theta)\left(\frac{\partial p_{c_{0,\alpha}}}{\partial S}\right)^E \mathbf{C}_\alpha^{E^{t+\Theta\Delta t}}\right)\mathbf{S}_\alpha^{t} + \\
&\sum_{E=0}^{elements-1}\left(\frac{1}{\Delta t}\mathbf{B}^E + \left(\frac{\partial p_{c_{0,ref}}}{\partial S}\right)^E \mathbf{C}_\alpha^{E^{t+\Theta\Delta t}}\right)\mathbf{S}_{ref}^{t+\Theta\Delta t} + \\
&\sum_{E=0}^{elements-1}\left(-\frac{1}{\Delta t}\mathbf{A}_{\alpha_I}^{E^{t+\Theta\Delta t}}\left(\left(\mathbf{p}_{ref}^{t+\Delta t} - \mathbf{p}_{c_{0,\alpha}}^{t+\Delta t} + \mathbf{p}_{c_{0,ref}}^{t+\Delta t}\right) - \left(\mathbf{p}_{ref}^{t} - \mathbf{p}_{c_{0,\alpha}}^{t} + \mathbf{p}_{c_{0,ref}}^{t}\right)\right)\right) + \\
&\sum_{E=0}^{elements-1}\left(-\mathbf{C}_\alpha^{E^{t+\Theta\Delta t}}\mathbf{p}_{ref}^{t+\Theta\Delta t} + \mathbf{D}_\alpha^{E^{t+\Theta\Delta t}} + \mathbf{E}_\alpha^{E^{t+\Theta\Delta t}}\right)
\end{aligned} \tag{3.83}$$

This formulation is only valid if the local changes of the saturation within an element are small. If this is not the case, a new element matrix type must be used, within which the derivative of the capillary pressure function is incorporated into the element integration loop. Furthermore, the capillary pressure function must have a finite derivative and must be constant within the regarded part of the domain. If the material properties change, additional treatment of the internal boundaries can be helpful to enhance the solution procedure. The easiest method would be to switch the used formulation for the elements near the inner boundaries. A more sophisticated approach would allow multiple saturations on a single node, each of them belonging to one of the different materials of the connected elements. Due to the significant impact on the program structure this was not realized yet.

For further developments it is suggested to solve the pressure field and the saturation fields in a combined system of linear equations for all primary variables. In this case, the discretized equations can share the primary variables. In this case, a local Taylor development can be used to achieve a Quasi-Newton scheme for the matrices of type **C** and **D**, which will result in better convergence near the solution. Here, the procedure is outlined for the matrices of type **C**:

$$
\begin{aligned}
&{}^{i+1}\mathbf{C}_\alpha^E \cdot {}^{i+1}\mathbf{p}_\alpha^E \approx \mathbf{C}_\alpha^E \cdot {}^{i+1}\mathbf{p}_\alpha^E \\
&+\left(\int_{V_E} \mathrm{grad}\,\boldsymbol{\varphi}^E\, \frac{\partial k_{r_\alpha}}{\partial S_\alpha}\frac{\mathbf{k}}{\mu_\alpha}\mathrm{grad}\,p_\alpha\,dV_E - \int_{V_E}\boldsymbol{\varphi}^E\mathrm{grad}\,\rho_\alpha\, \frac{\partial k_{r_\alpha}}{\partial S_\alpha}\frac{\mathbf{k}}{\rho_\alpha\mu_\alpha}\mathrm{grad}\,p_\alpha\,dV_E\right)\left({}^{i+1}\mathbf{S}_\alpha^E-{}^{i}\mathbf{S}_\alpha^E\right)
\end{aligned}
\tag{3.84}
$$

In this equation an approximation for matrix **C** at iteration level i+1 is proposed to be used while setting up the system of linear equations. It is estimated from the standard matrix **C** on the old iteration level i and additional terms, which use the partial derivative $\partial k_{r_\alpha}/\partial S_\alpha$ and the change of the phase saturation ${}^{i+1}\mathbf{S}_\alpha^E-{}^{i}\mathbf{S}_\alpha^E$ between the iteration steps. These terms can be added to the iterative scheme if the solution is accurate enough to switch to a Quasi-Newton scheme.

3.5.2 Runtime adaptation of the physical model

In Figure 3.21 a typical coupled nonlinear system is presented. It includes multiphase flow as well as density driven flow effects.

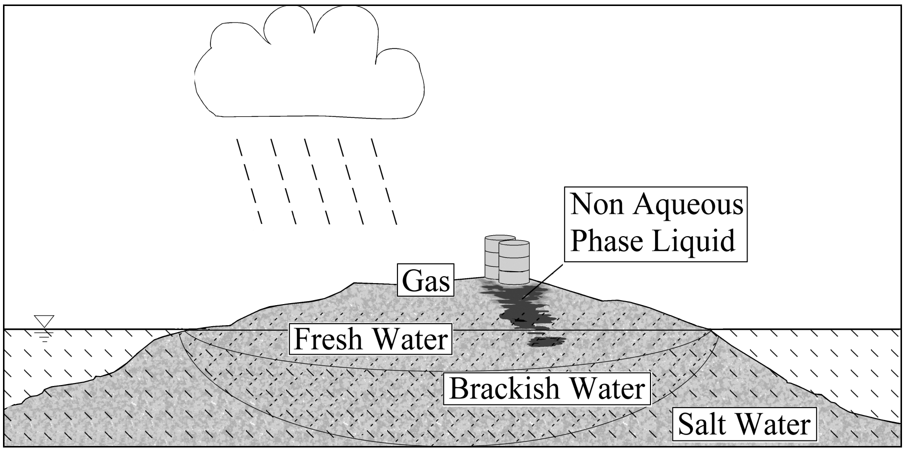


Fig. 3.21: An island as an example for a nonlinear subsurface flow system. Multiphase flow and density driven flow must be regarded to describe the system.

The PDE to describe the fluid pressure for this example results from Equation 2.17, which has to be formulated for multiphase flow of three phases (Equation 3.80). It is assumed that the modeller decides that the compressibility of fluids and matrix are negligible for the processes of his objective. Furthermore the modeller has decided to use phase 0 as reference phase for the pressure field. Thus, the following subset of the multiphase pressure equation (3.80) for a three-phase system is to be solved:

$$\begin{aligned}
&-\frac{1}{\rho_0}\operatorname{div}\left(\rho_0\frac{k_{r_0}\mathbf{k}}{\mu_0}\left(\operatorname{grad}(p_0)-\rho_0\mathbf{g}\right)\right)\\
&-\frac{1}{\rho_1}\operatorname{div}\left(\rho_1\frac{k_{r_1}\mathbf{k}}{\mu_1}\left(\operatorname{grad}\left(p_0+p_{c_{0,1}}\right)-\rho_1\mathbf{g}\right)\right)\\
&-\frac{1}{\rho_2}\operatorname{div}\left(\rho_2\frac{k_{r_2}\mathbf{k}}{\mu_2}\left(\operatorname{grad}\left(p_0+p_{c_{0,2}}\right)-\rho_2\mathbf{g}\right)\right)\\
&=Q_0+Q_1+Q_2
\end{aligned}\tag{3.85}$$

This is a subset of the superset pressure PDE 2.17 and is valid in the entire domain. The same equation can be written in matrix notation for the FEM formulation (Chapter 3.1):

$$\begin{aligned}
&\sum_{E=0}^{\text{elements}-1}\left[\sum_{\alpha=0}^{2}\left(\Theta\mathbf{C}_{\alpha}^{E^{t+\Theta\Delta t}}\right)\right]\cdot\mathbf{p}_0^{t+\Delta t}=\\
&\sum_{E=0}^{\text{elements}-1}\left[\sum_{\alpha=0}^{2}\left((1-\Theta)\mathbf{C}_{\alpha}^{E^{t+\Theta\Delta t}}\mathbf{p}_0^{t}+\right.\right.\\
&\left.\left.\mathbf{C}_{\alpha}^{E^{t+\Theta\Delta t}}\Theta\mathbf{p}_{c_\alpha}^{t+\Delta t}+\mathbf{C}_{\alpha}^{E^{t+\Theta\Delta t}}(1-\Theta)\mathbf{p}_{c_{0,\alpha}}^{t}+\mathbf{D}_{\alpha}^{E^{t+\Theta\Delta t}}+\mathbf{E}_{\alpha}^{E^{t+\Theta\Delta t}}\right)\right]
\end{aligned}\tag{3.86}$$

In the chosen formulation the index 0 denotes the non-wetting phase (in this case the gas), index 1 the non-aqueous liquid, index 2 water.

It is obvious that the fluid mobilities for some phases are equal to zero in parts of the domain, as the saturation of the appropriate phase is below the residual saturation. As a consequence it is absolutely legitimate to ignore the matrices of type **C** (Equation 3.15) and **D** (Equation 3.16) in the appropriate part of the domain for these phases if the saturation drops below the residual saturation. The matrices of type **E** are treated as nodal loads and are therefore not regarded here.

This evaluation is performed as a first step, resulting in a largely simplified PDE, with a limited set of associated matrices (Figure 3.22) in parts of the domain. This process is equivalent to a *downsizing of the physical model*, thus adapting it to the relevant processes.

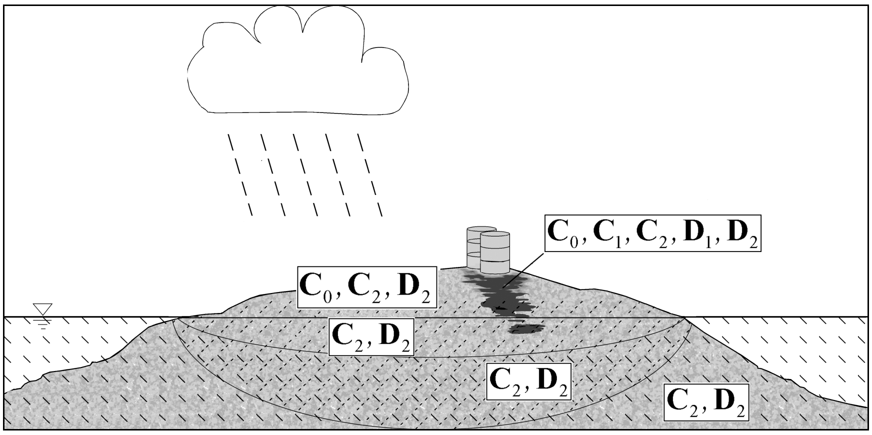


Fig. 3.22: Matrix types belonging to relevant processes in the system. In large parts of the system a limited subset is sufficient to describe the relevant processes.

When employing the model-adaptive methods, it is important to note that the system of linear equations does not degenerate by neglecting terms of the PDEs, because only terms that do not have an impact on the PDE are neglected. For the saturation PDE this can result in a system that is only determined by the mass lumped storage matrix, thus only the diagonal of the linear equation system is filled.

3.5.3 Caching of system matrices

Due to the nonlinear coupling, the matrices that correspond to the terms of Equation 3.85 are rebuild in each iterative step. As the physical model was already adapted by ignoring irrelevant parts of the PDE, the regarded PDE is now spatially differing. This is not problematic, as the primary variable will stay continuous over the whole domain. A further look on the example PDE shows that only parts of the PDE are involved in the nonlinear coupling. This, again, differs spatially. During the simulation some of those matrices have to be rebuilt, others are static. In Figure 3.23 the matrices of Equation 3.86 are presented which change drastically enough to justify a costly matrix recalculation.

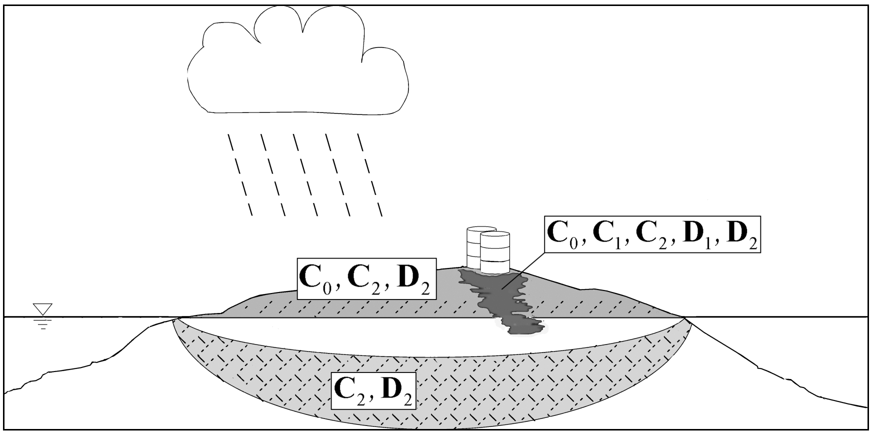


Fig. 3.23: Distribution of PDE parts belonging to matrices, which have to be rebuilt. In the grey areas a subset of the matrices must be rebuilt, in the white areas no recalculations are necessary.

In the saturated zone the capillary pressure can be neglected and the relative permeability can be assumed to be constant. In the zone of brackish water, the density changes due to salt transport. Therefore in a large part of the system (white soil areas in Figure 3.23) the resulting matrices can be regarded as constant, some matrices have to be rebuilt due to changes of the density (matrices $\mathbf{C}_2$, $\mathbf{D}_2$ in the light grey area). In the unsaturated soil areas (medium grey areas) the flux matrices for gas and water ($\mathbf{C}_0$, $\mathbf{C}_2$, $\mathbf{D}_2$) have to be rebuilt. Furthermore, the element capillary pressure vectors must be refreshed. In the NAPL contaminated area multiple matrices ($\mathbf{C}_0$, $\mathbf{C}_1$, $\mathbf{C}_2$, $\mathbf{D}_1$, $\mathbf{D}_2$) must be recomputed is necessary.

Only parts of the matrices really have to be rebuilt. The necessity is checked during a time step or iterative step (according to Figure 3.23). Thus the old system matrices of the grey areas are *cached,* together with the matrices of the white areas.

The simulator has to store the matrices together with characteristic values (e.g. relative permeabilities, densities etc.), which have determined the situation in which these matrices were built. These characteristic values are evaluated during the runtime and only if the differences diverge over a critical limit, the matrices will be triggered for a recomputation. Small changes of phase mobilities, densities or velocities can be neglected, if they are below a user specified threshold. The stored matrices for flow and transport can be reused in this case.

As characteristic values the

- phase mobilities,
- phase densities,
- phase velocities and
- phase saturation

in the middle of each element are used. The recomputation of matrices can be triggered by absolute values of the reference variables, absolute changes of the reference variables or relative changes of the reference variables. This means that the error ε is evaluated from the reference variable u for each matrix type M that is marked as relevant for phase α in Element E:

$$\begin{aligned}&\textit{Absolute values:}\\ &\varepsilon^{\text{abs.}}_{\text{M,E},\alpha} = u_{\text{M,E},\alpha}\\[1em] &\textit{Absolute changes:}\\ &\varepsilon^{\text{abs.}}_{\text{M,E},\alpha} = \left| u_{\text{M,E},\alpha} - u_{\text{M,E},\alpha_{\text{ref}}} \right|\\[1em] &\textit{Relative changes:}\\ &\varepsilon^{\text{rel.}}_{\text{M,E},\alpha} = \left| 2 \cdot \frac{u_{\text{M,E},\alpha} - u_{\text{M,E},\alpha_{\text{ref}}}}{u_{\text{M,E},\alpha} + u_{\text{M,E},\alpha_{\text{ref}}}} \right|\end{aligned} \tag{3.87}$$

The current value of the characteristic reference variable u is compared to the stored reference value u_{ref} for each active matrix type of each phase for each element. The calculation of a matrix is triggered if a critical constraint for these values is violated:

$$\begin{aligned}&\textit{Deterministic conditions for matrix rebuild:}\\ &1)\quad \textit{Absolute changes}\quad \varepsilon^{\text{abs.}}_{\text{M,E},\alpha} \big/ \varepsilon^{\text{crit}}_{1} \geq 1\\ &2)\quad \textit{Relative changes}\quad \varepsilon^{\text{rel.}}_{\text{M,E},\alpha} \big/ \varepsilon^{\text{crit}}_{1} \geq 1\\ &3)\quad \textit{Absolute and relative changes}\quad \left(\varepsilon^{\text{abs.}}_{\text{M,E},\alpha} \big/ \varepsilon^{\text{crit}}_{1} \geq 1\right) \wedge \left(\varepsilon^{\text{rel.}}_{\text{M,E},\alpha} \big/ \varepsilon^{\text{crit}}_{2} \geq 1\right)\end{aligned} \tag{3.88}$$

By this means the errors are limited to a specified range. This approach has the disadvantage that for stationary problems the optimal solution can not be reached, as all matrices will tend to stay at a certain error, if this error is below the critical limit.

An enhancement in this situation is a probability based approach. The condition for the matrix recomputation is enhanced by a random function r and a weighting factor λ:

$$\begin{aligned}&\textit{Probabilistic conditions for matrix rebuild:}\\ &1)\quad \textit{Absolute changes}\quad \varepsilon^{\text{abs.}}_{\text{M,E},\alpha} \big/ \varepsilon^{\text{crit}}_{1} \geq r^{\lambda} \quad ;0 \leq r \leq 1\\ &2)\quad \textit{Relative changes}\quad \varepsilon^{\text{rel.}}_{\text{M,E},\alpha} \big/ \varepsilon^{\text{crit}}_{1} \geq r^{\lambda} \quad ;0 \leq r \leq 1\\ &3)\quad \textit{Absolute and relative changes}\quad \left(\varepsilon^{\text{abs.}}_{\text{M,E},\alpha} \big/ \varepsilon^{\text{crit}}_{1} \geq r^{\lambda}\right) \wedge \left(\varepsilon^{\text{rel.}}_{\text{M,E},\alpha} \big/ \varepsilon^{\text{crit}}_{2} \geq r^{\lambda}\right) \quad ;0 \leq r \leq 1\end{aligned} \tag{3.89}$$

The random number r is generated on the basis of a Poisson distribution in the range between zero and unity. This relates the relative frequency for a matrix re-calculation to the deviation from the reference value of the characteristic variable. Figure 3.24 shows the probability function for rebuilding a matrix as a function of the relative error $\varepsilon_{\text{M,E},\alpha} \big/ \varepsilon^{\text{ref}}$ and the form factor λ. For λ equal to zero, the standard behaviour of the deterministic scheme is gained again.

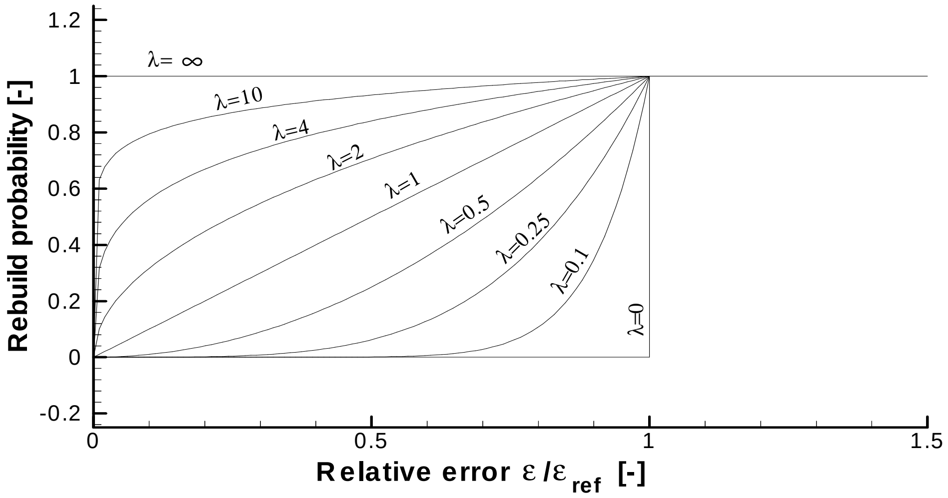


Fig. 3.24: Matrix rebuilding probabilities as a function of relative error $\varepsilon_{M,E,\alpha}/\varepsilon^{ref}$ and form factorλ. For small values of λ the behaviour is equivalent to the deterministic rebuilding scheme, for very large λ it is equivalent to the standard approach with no matrix caching.

If a certain matrix is triggered for a recalculation by changes of the corresponding variable, the new variable will be stored together with the reference value. As a single matrix can be triggered for a recalculation by multiple variables, it is important that all reference values are refreshed for the matrix, even if they did not cause this specific recomputation.

3.5.4 Indexing criteria for matrix recalculations

A large set of indexing criteria can be set up, though it seems useful to limit the number, as the computational effort for evaluating the criteria should be kept low compared to the effort to recalculate the matrices. The following indexing criteria are proposed:

For Richards' approach the pressure of the wetting phase determines the water saturation, thus if the pressure is bigger than the reference pressure, the regarded part of the system is fully saturated and therefore most of the matrices are static. If density effects are taken into account, the recalculation of the gravity matrix must be triggered if significant density changes occur. In the rest of the domain the soil is unsaturated, thus the impact of changing saturations can trigger all matrices for a recalculation. If transport processes are taken into account, the matrices of the transport kernel have to be recomputed if the fluid velocity or the saturation diverge too much.

For the full multiphase flow model the situation is similar. The relative phase mobility k_{r_α}/μ_α is used to qualify if a matrix recalculation is necessary. Furthermore, the fluid density is important for density driven flow.

Changes of the saturation have an impact on multiple matrices. As it is easier to qualify the error that is introduced via neglected changes of the phase mobilities and the capillary pressure, it is recommended to use these as indicators for the recalculation. The saturation is a good indicator for the recalculation of the storage matrix, if compressible fluids or transport processes are regarded. In Table 3.3 the matrix identifiers defined in Equation 3.13-3.17 are used.

Table 3.3: Dependencies for matrix recalculations in the flow model.

Reference variable of phase α	Matrices, Vectors
Saturation	$\mathbf{A}_\alpha$, $\mathbf{p}_{c_{0,\alpha}}$
Density (function of pressures, concentrations and temperatures)	$\mathbf{A}_\alpha$, $\mathbf{C}_\alpha$, $\mathbf{D}_\alpha$
Mobility (function of saturations, viscosity and for nonlinear flow regimes the velocity)	$\mathbf{C}_\alpha$, $\mathbf{D}_\alpha$

For the tracer transport model a more simplistic approach was chosen, as the tracer transport model is not an integral part of this work. Thus, only a very limited subset of the approach presented above was realized for the tracer transport model. The rebuild of all matrices was related to changes of the fluid saturation and the fluid velocity.

In order to visualize the considerations presented above, the system presented in Chapter 4.6 will be chosen as a test case. Please refer to that chapter for the system description. While all of the calculations presented in Chapter 4.6 are performed in a steady state flow field, here the transient behaviour in the beginning of the experiment for set-up B is chosen. This is the more difficult test case, as the transient flow field undergoes severe changes during the simulation.

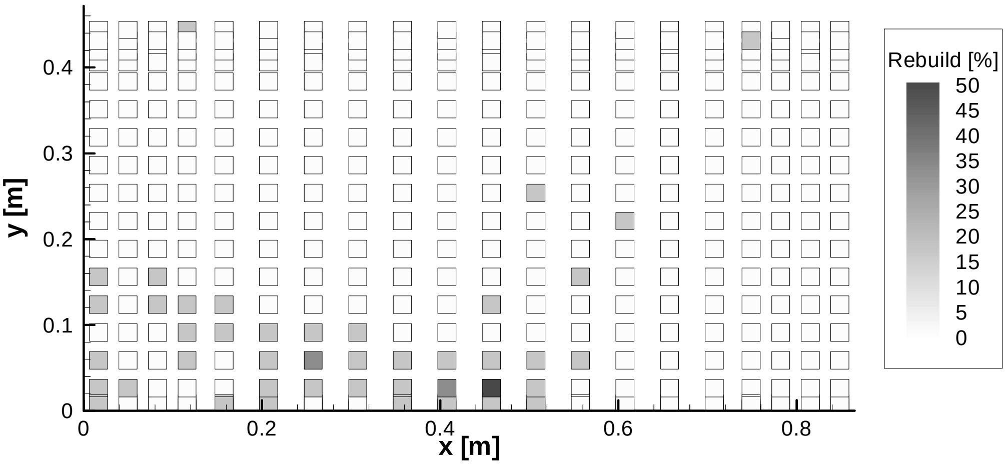


Fig. 3.25: Distribution of performed matrix recalculations for a single time step, relative to the number of recalculations for the standard approach. Mainly matrices near the head of the saltwater front are recalculated.

In Figure 3.25 the percentage of performed matrix recalculations in the current time step is presented. This is the situation after 80000 seconds of saltwater intrusion. The system is not yet in steady state, as can be seen from the number of matrix recalculations around the position of the saltwater tongue. In the plot the recalculations of all matrix types are incorporated by using an arithmetic mean, they are not weighted with the effort to build them. Obviously the unsaturated zone is already in state, as the number of recalculations is very low. In the saturated zone the recalculations concentrate on the area around the advancing saltwater tongue.

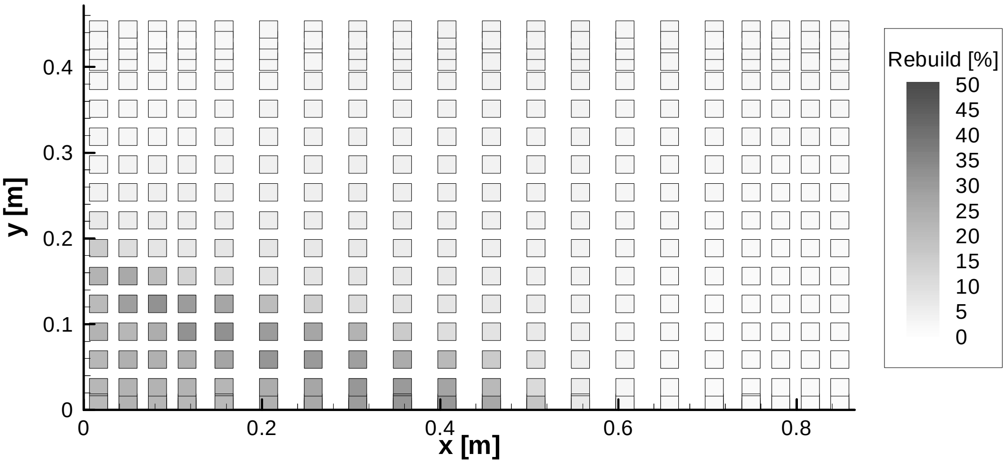


Fig. 3.26: Distribution of accumulated matrix recalculations for all time steps, relative to the number of recalculations for the standard approach. In large parts of the domain only a small number of recalculations was performed, they concentrate on the area of the saltwater tongue.

An additional possibility to evaluate the results is presented in Figure 3.26. In this plot the effort for rebuilding the system matrices was additionally averaged over all previous time steps. It becomes obvious that the computational effort is successfully focused onto the system areas where large changes of the nonlinear regime took place. For the example presented above, the total computational effort is halved in comparison to the standard approach.

3.5.5 Accuracy considerations

The errors introduced by not setting up some matrices in parts of the system where the saturations are below the residual saturations (i.e. the model adaptive approach) are equal to zero, as for those areas the model adaptive approach is a fast a priori estimation to determine if the flux matrix will completely be equal to zero.

More difficulties arise from estimating the errors that are introduced by the second step, the conditional matrix recalculations. For the flow model, the rebuilding criteria based on the fluid mobility are directly related to mass flux errors of the regarded phase. The criteria based on saturation changes lead to equivalent errors in the storage of the regarded phase. Furthermore, nonlinear errors are introduced via the capillary pressure function and the density of the fluid. The inaccuracies introduced by not rebuilding the tracer transport matrices are equivalent to phase speed variations on the tracer front. These errors are equal to the relative values of the recalculation criteria for those matrices.

Generally it is important to specify a useful range for the recalculation criteria in comparison to the errors introduced by the linear solver and the nonlinear iteration loop:

$$\varepsilon_{\mathrm{lin}} \le \varepsilon_{\mathrm{reb}} \le \varepsilon_{\mathrm{non-lin}} \tag{3.90}$$

with

$\varepsilon_{\mathrm{lin}}$ *: Accuracy criterion for linear solver*

$\varepsilon_{\mathrm{reb}}$ *: Rebuild criterion*

$\varepsilon_{\mathrm{non-lin}}$ *: Accuracy criterion for non - linear solver*

It is obvious that a recalculation criterion smaller than the accuracy of the linear solver is useless, as the reliability of the results is not given in this range of values. Furthermore, the nonlinear solver will suffer from convergence problems if the inner results of the linear results are not accurate enough. By setting the recalculation criterion to smaller values than the nonlinear error criterion, it can be guaranteed that the total error is not determined by the conditional rebuilding of matrices, but by the nonlinear solver.

3.5.6 Efficiency considerations

After profiling multiple simulations, it became obvious that for coupled systems setting up the system matrices can be much more costly than solving the resulting system of linear equations. This is especially true for systems with elements of higher dimensions. While the integration of 1D-elements is performed very fast, the integration of 2D and even more 3D elements is quite expensive, so that the overhead for evaluating the recalculation criteria can pay off later by not rebuilding some matrices. The necessary effort to evaluate the indicators should be related to the possible gain from not rebuilding the corresponding matrices. The effort is determined by the effort to determine the saturation, density and mobility in the middle of an element.

As an example, the profiling results for a simple test case will be presented here. The example by *Förster and Bruck* (Chapter 4.2) was discretized with 1D, 2D and 3D elements in order to compare the effort in the different parts of the numerical models. An instationary simulation was performed, modelling the initial flooding of the system with gas. In Table 3.4 the distribution of CPU time spent in different parts of the simulator is compared for the different discretizations. Obviously the effort is shifted towards setting up the system matrices for higher element dimensions.

Table 3.4: Comparison of relative CPU time spent in different parts of the simulator. The numbers do not sum up to 100%, because some overhead for I/O, generating post-processing data etc. is necessary.

	1D	2D	3D
Setting up matrices	**10%**	**35%**	**66%**
Assembling matrices	**3%**	**3%**	**3%**
Solving system of linear equations	**85%**	**61%**	**30%**

The gain-to-effort ratio for the model adaptive methods would be quite bad for one-dimensional elements, where one indicator value must be evaluated to possibly prevent the integration over two node values. For a two-dimensional element (in the following it is assumed that three Gaussian points for each direction are used), this ratio is already much better, as in total nine integration points must be considered. For a three-dimensional element with 27 Gaussian integration points, the additional effort pays of easily. If the Group-FEM approach (Equation 3.3) is not chosen, on each integration point several variables (saturation, density and mobility) must be evaluated. Furthermore several matrix transformations must be performed, resulting in a substantial additional effort.

It is important to note that the model adaptive downsizing approach (i.e. the automatic downsizing of the physical model) shows good performance, even if heavy interactions between the different PDEs take place via nonlinear coupling functions. In contrast, the conditional recalculation scheme will have a negative impact on the convergence of the nonlinear iteration scheme for these systems, so that the advantage from not rebuilding the system matrices must be carefully balanced against the disadvantage in the nonlinear solving procedure.

For weak or non-existing nonlinear couplings, the conditional recalculation scheme shows significant advantages in comparison to the standard scheme. The same is true for nonlinear systems, if they are near their steady state.

If additionally a grid adaptive scheme is used, the grid adaptive scheme minimizes the number of elements in regions with low *spatial* changes of the primary variables, the caching methods presented above try to minimize the number of matrix recalculations, if the *temporal* changes are low. Due to the nature of the regarded systems, spatial and temporal changes are often related to each other, so that both approaches hinder each other. This is no longer true if *multiple* coupled processes are regarded. Due to the additive behaviour of the grid adaptation scheme, the grid will already be refined if only one of the processes shows large spatial variability (compare Figure 4.18, the grid is refined due to changes of salinity, saturation and pressure). This does not necessarily imply that the other processes require this refinement, too. Anyhow the corresponding system matrices for all processes will be rebuilt during the subsequent time steps in these regions. This will be circumvented by using the proposed model adaptive approach with conditional matrix recalculation, resulting in better performance of the simulator.

3.6 Global Algorithm

The global algorithm of the simulator is described in the following. In order to simplify the representation, it is divided into hierarchical ordered points of objective. The global algorithm is presented in Figure 3.27, it must be pointed out that the global algorithm is non-static; it can be switched on the basis of the information in the input files and thus is “polymorph”. This polymorphism is realized by using function pointers in the code. Furthermore, it must be stressed that only a very rough overview of the functionality can be provided her, due to the complexity of the code.

```
Read input files

Construct FE data structures
Initialize chosen model
  - initialize model node and element values
  - initialize functions pointers
  - initialize computational kernels
Loop over all time steps
    Predict results on new time level
    A priori grid adaptation
            If strong non-linear coupling: Calculate pressure
            If strong non-linear coupling: Calculate saturations
            If strong non-linear coupling: Calculate concentrations
        Repeat until convergence is reached
        If weak non-linear coupling: Calculate pressure
        If weak non-linear coupling: Calculate saturations
        If weak non-linear coupling: Calculate concentrations
        Estimate length of next timestep
    A posteriori grid adaptation
    Write results
Destruct all data structures
```

Fig. 3.27: Pseudo code for the global algorithm.

The time loop in the algorithm contains several single computational kernels for pressure, saturations and concentrations. These can either be incorporated into an iterative loop to resolve nonlinearities or can be evaluated once per time step for weakly coupled or linear processes. Within each FEM-kernel several tasks are performed. The structure of the pressure calculation kernel is outlined in Figure 3.28 as a typical example.

```
Loop over all rebuild criteria
    Loop over all elements
        Evaluate current values of reference variables
Loop over all rebuild criteria
    Loop over all elements
        Compare stored values of reference variables to current ones
        Evaluate error criterion and mark matrix:
            -1: Delete matrix
             0: Keep old matrix
             1: Rebuild matrix
Loop over all rebuild criteria
    Loop over all elements
        Store new reference values for matrix if necessary
        Rebuild matrix if necessary
Loop over all elements
    Incorporate matrices into global system of equations
Apply sources and sinks to global system of equations
Eliminate irregular nodes from global system of equations
Apply Dirichlet boundary conditions to global system of equations
Solve system of linear equations
```

Fig. 3.28: Pseudo code for the algorithm within
a finite element kernel.

Within the kernels the main workload is done in the functions which recalculation the system matrices. Furthermore the solver for the system of linear equations can require significant CPU time.

4 Test cases and applications

In the following a set of test cases is presented, which is supposed to give an overview over the capabilities of the developed numerical model. The examples where chosen to demonstrate the possibility to transform the complex multiphase flow model into simpler ones, i.e. to demonstrate the model adaptivity. The chosen examples cover the following typical problem classes:

- saturated single-phase flow of an incompressible fluid with storativity
- saturated single-phase flow of a compressible fluid
- unidirectional two-phase flow
- multidirectional two-phase flow
- tracer transport in fracture networks
- coupled density driven and unsaturated-saturated flow (with laboratory experiment)
- two-phase flow in a fracture network with tracer transport
- three-phase flow in a porous medium

4.1 Pressure drawdown around a well

This is a classical test case for numerical groundwater models. The transient pressure draw down around a well during a pump test in an infinite two-dimensional domain is regarded. It is presented here as a very simple application for the downsizing methods of the model.

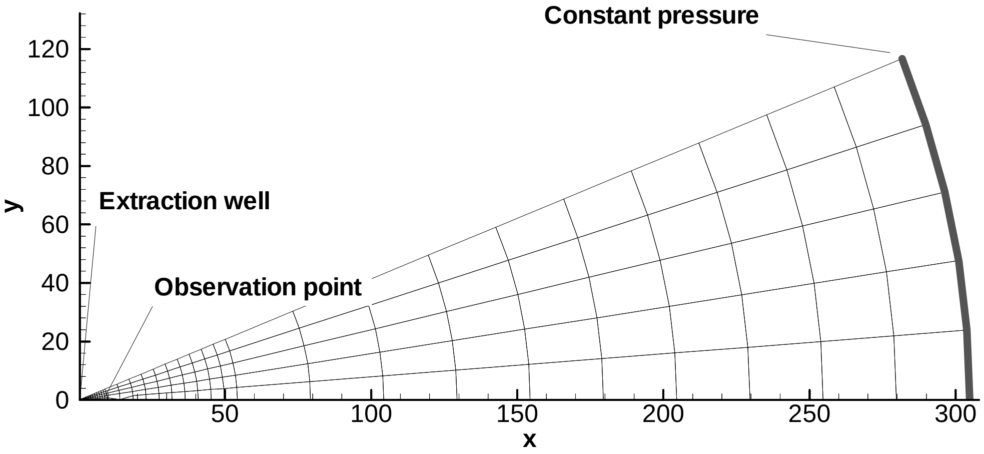


Fig. 4.1: FE-mesh for Theis' example. A limited part of the domain is sufficient for the simulation, as the system is symmetric.

In Figure 4.1 the geometry for this example is presented. Due to the symmetry of the system, only a sector of the total domain is regarded. The sides of the sector are impermeable. A constant flux is prescribed on the well. The outer boundary is simulated with a constant pressure and subsequently with a closed border in order to quantify the impact of the chosen boundary condition type on the results. The system is discretized with 40 nodes in radial direction; the axial nodes are used to check for boundary effects in the solution (which are not observed).

Table 4.1: Physical properties for the simulation of the pressure drawdown in a well.

Discharge rate	8.849e-04 m^3/s
Storativity	1.e-7 ms^2/kg
Permeability	9.2903e-11 m^2
Pressure	$p(t = 0) = 0$ Pa
	$p(t, r = 304.8) = 0$ Pa
Radius	$0.3048m < r < 304.8m$

The analytical solution for this test case assumes an infinite domain. In the results of the numerical model the influence of the boundary becomes visible for the last compared time step (Table 4.2), as the results for prescribed pressure and closed boundary start to differ.

In total the results show a good agreement between computational and analytical solution, the small initial error smoothes out very fast. A further refinement of the mesh reduces the initial error significantly, after bisecting the mesh the initial result enhanced to a pressure difference of 20 Pa, a further bisection to 11 Pa and a last one to 9 Pa. Thus, grid convergence is reached for a result sufficiently near the analytical solution.

Table 4.2: Comparison of numerically and analytically calculated drawdown in a well.

Time [s]	Pressure [Pa] analytical	Pressure [Pa] numerical, pressure prescribed on boundary	Pressure [Pa] numerical, closed boundary
10	-367.88	-309.65	-309.65
100	-12664.	-12371.	-12371.
1000	-37562.	-37569.	-37569.
10000	-64687.	-65040.	-65043.

4.2 Gas flow through a rock sample

In this test case the flow of a compressible fluid through a homogeneous porous medium under isothermal conditions is regarded (*Förster and Bruck,* 1994). The system is a one-dimensional rock column. It is completely saturated with gas. The prescribed pressures at the boundaries of the system induce the flow process. Due to the compressibility, a nonlinear pressure field can be observed in the sample. The impact of gravity can be neglected in comparison to the applied pressures, as the vertical extent of the sample is very small. This and other examples for gas flow are extensively covered in *Shao et al.* (1995).

4.2.1 Analytical solution

Under the assumption of single-phase flow with no sinks or sources in the domain, it is possible to simplify the mass conservation Equation 2.3 for the one-dimensional case. Integration leads to:

$$\mathrm{div}(\rho q) = 0 \Rightarrow \rho q = \mathit{constant} \tag{4.1}$$

I.e. the mass flux is constant in each cross-section of the system. Together with Darcy's law (2.4) and the equation of state for ideal gases we obtain for isothermal conditions and neglected influence of gravity:

$$p \;\mathrm{grad}\; p = \mathit{constant} \tag{4.2}$$

Integration delivers the pressure reduction curve in the sample:

$$p_2 = \sqrt{\left(\frac{p_3^2(x_2 - x_1) + p_1^2(x_3 - x_2)}{x_3 - x_1}\right)} \tag{4.3}$$

If the pressures p_1 at position x_1 and p_3 at x_3 are known (i.e. the boundary conditions), the pressure p_2 at an arbitrary position x_2 in between can be computed from the above equation.

4.2.2 Comparison of analytical and numerical solution

It is seen from the derivation of the analytical solution that the soil parameters are not relevant to the solution. The boundary conditions are chosen as in the experiments by *Förster and Bruck* (1994):

$p(x = 0\ \mathrm{m}) = 1695500\ \mathrm{Pa}$, $p(x = 0.0488\ \mathrm{m}) = 95500\ \mathrm{Pa}$

The sample has a length of 4.88 cm, it is discretized with 36 adaptively placed nodes, resulting in a minimal element length of 0.04 cm in the area with the strongest curvature of the pressure field. Under those conditions the results of Figure 4.2 are achieved. The difference between analytical and numerical solution is very small, only directly near the boundary condition a significant error is encountered. By defining $\varepsilon = \max_i((P_{ana}-P_{num})/P_{ana})$ the maximal error ε is below 4%.

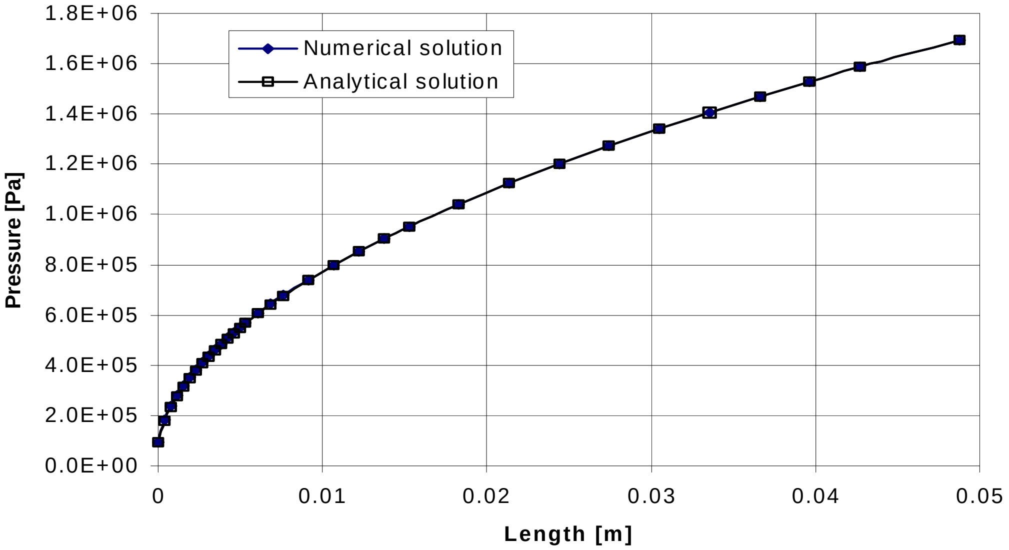


Fig. 4.2: Pressure decrease for gas flow through a rock sample under high pressure. The results of the analytical and the numerical solution are indistinguishable.

4.3 Unidirectional two-phase flow

4.3.1 Introduction

The object of interest is the displacement of two immiscible fluids in porous media. Buckley & Leverett stated their semi-analytical solution for the simplified one-dimensional problem in 1941 (*Buckley & Leverett*, 1941). Their problem is a classic benchmark, which has been widely used to test the handling of self-sharpening fronts in multiphase flow simulation programs (e.g. *Huyakorn*, 1978; *Helmig*, 1993). It originates in examinations that were performed by the oil industry. In a one-dimensional set-up the initial oil filling is extracted on one side and water is entering the system on the other side. Thus, a moving front of water can be examined (Figure 4.3). Due to the shape of the permeability functions a shock front is establishing. This shock front is a severe test for numerical models, as it is difficult to catch the curvature and slope of the saturation in the numerical model. In natural systems the capillary pressures would smoothen the front. Some authors claimed the necessity to introduce artificial capillary pressure in order to enable a numerical solution. Capillary pressure between the two phases acts in a similar way as diffusion does in transport processes, it smoothes the front and therefore simplifies the numerical solution.

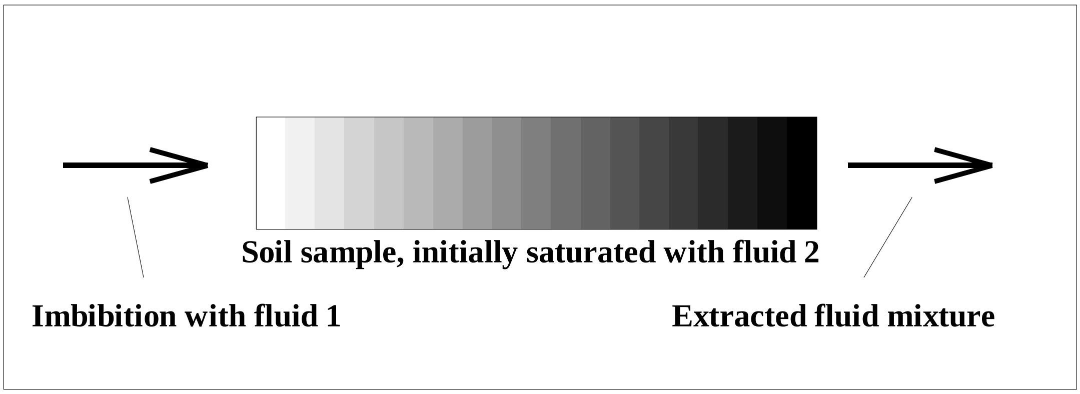


Fig. 4.3: System set-up for the Buckley-Leverett test case.

4.3.2 Analytical solution

The literature offers multiple possible solutions to this problem. The one originally proposed by Buckley & Leverett will be presented here.

The mass conservation equation (2.3) can be simplified to

$$n\frac{\partial S_\alpha}{\partial t} = -\operatorname{div} q_\alpha \tag{4.4}$$

under the assumptions that the fluid is incompressible, the flow field is one-dimensional, gravity can be neglected and that no sinks or sources are inside the domain. The *fractional-flow-functions* describe the flux of one phase related to the total flux of all phases:

$$f_\alpha = \frac{q_\alpha}{q_{total}},\ \sum_\alpha f_\alpha = 1 \qquad \Rightarrow\ f_1 = \frac{q_1}{q_1+q_2} = \frac{\frac{k_1}{\mu_1}}{\frac{k_1}{\mu_1}+\frac{k_2}{\mu_2}} = \frac{1}{1+\frac{k_2}{\mu_2}\frac{\mu_1}{k_1}} \tag{4.5}$$

The conservation equation can be expressed in terms of the total flux:

$$\frac{\partial S_\alpha}{\partial t} = -\frac{q_{total}}{n}\frac{\partial f_\alpha}{\partial x} \tag{4.6}$$

The relative permeabilities are not yet used in this expression. Buckley & Leverett used the following expressions, which they extracted from laboratory experiments:

$$k_1 = \frac{\left(S_1 - S_{1_r}\right)^2}{\left(1-S_{1_r}-S_{1_r}\right)^2},\ k_2 = \frac{\left(1-S_1-S_{2_r}\right)^2}{\left(1-S_{1_r}-S_{2_r}\right)^2} \tag{4.7}$$

In those equations the sub-index r denotes the residual saturation of a phase. Incorporating the fractional flow function f_1 and the relative permeabilities k_1 and k_2 into the conservation equation and some reordering leads to:

$$\Delta x = -\frac{q_{\text{total}}}{n} \frac{\partial \left(\dfrac{1}{1 + \dfrac{k_2}{\mu_2} \dfrac{\mu_1}{k_1}} \right)}{\partial S_1} \Delta t \tag{4.8}$$

Now we can derive the saturation curve presented in Figure 4.4. The saturation curve is non-unique and has multiple possible saturations for one coordinate. The real saturation curve is gained by establishing a shock front in a way that the areas A and B are equal.

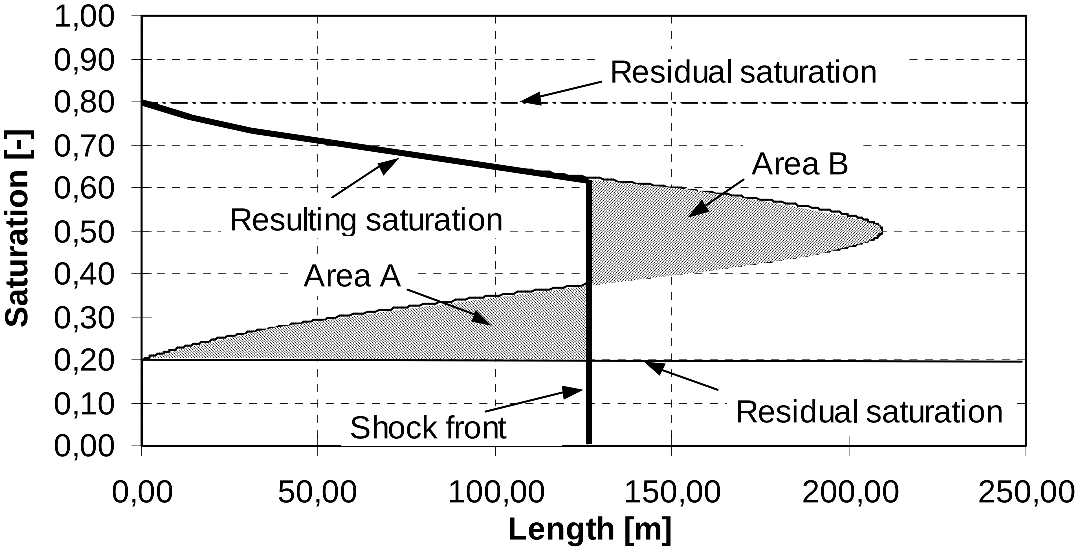


Fig. 4.4: Graphical determination of the shock front position for the Buckley-Leverett example. The shock front is established at a position for which Area A is equal to Area B.

4.3.3 Numerical solution

The standard test case is extended to a two- and three-dimensional problem. For the two-dimensional case one side is shifted upwards as a test for the gravitational terms in the numerical model (Figure 4.5).

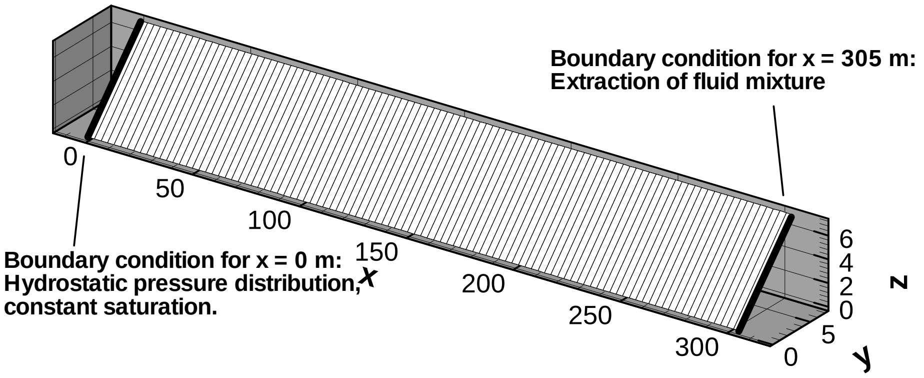


Fig. 4.5: Geometry for the Buckley-Leverett problem, discretized with two-dimensional planar elements.

Thus, the system is evaluated with one-dimensional, two-dimensional and three-dimensional discretization. The meshes for the one-dimensional and three-dimensional discretization are not presented here; they use the same spacing as the two-dimensional system.

Table 4.3: Physical specifications for the Buckley-Leverett test case.

Length	301.95 m
Area of cross-section	10 m^2
Simulation time	969 days
Porosity	0.2 [-]
Permeability	No influence
Rel. permeability	$k_{r_\alpha} = \left(\frac{S_\alpha - 0.2}{0.6}\right)^2$
Viscosity ratio	1 [-]
Extraction rate at x = 301.95m	$1.505 \cdot 10^{-6}$ m^3/s
Initial saturation of first phase	0.2 [-]
Saturation of first phase at x = 0 m	0.8 [-]

The impact of different numerical effects was already presented in Chapter 3.1.6, thus they will not be repeated here. The most important parameters for the numerical model are collected in the following table.

Table 4.4: Numerical parameters for the Buckley-Leverett test case.

Number of nodes in x-direction	100
Number of elements	99
Element lengths	3.05 m
Mass lumping	Yes
Upwinding	Mobility
	1D: α = 0.6; 2D, 3D: α = 0.8
Time collocation	Θ =0. 5
Time step length	412528.266 s
Final simulation time	82505653 s

For the simulations the final time of 969 days is reduced by the time that is necessary to fill the patch that belongs to the first node. This is necessary, because this patch is already filled at the beginning of the simulation due to the boundary condition. The computed saturation distributions (Figure 4.6) show a good agreement to the analytical results. The smoothing of the front depends on the choice of the upwind coefficient. Using upwinding is necessary for this example, as otherwise unphysical behaviour is encountered (compare to Chapter 3.1.6.1). Sizing the upwind parameter is always a compromise between good reproduction of the front steepness and beginning oscillations. In this case, the upwind parameter was chosen a little larger for 2D and 3D elements, because the upwinding is internally reduced for some points of the Gaussian integration (Chapter 3.1.6.1). Finalizing the quality of the results is good, the front advancing speed is reproduced correctly and the smoothing of the front is in a tolerable range.

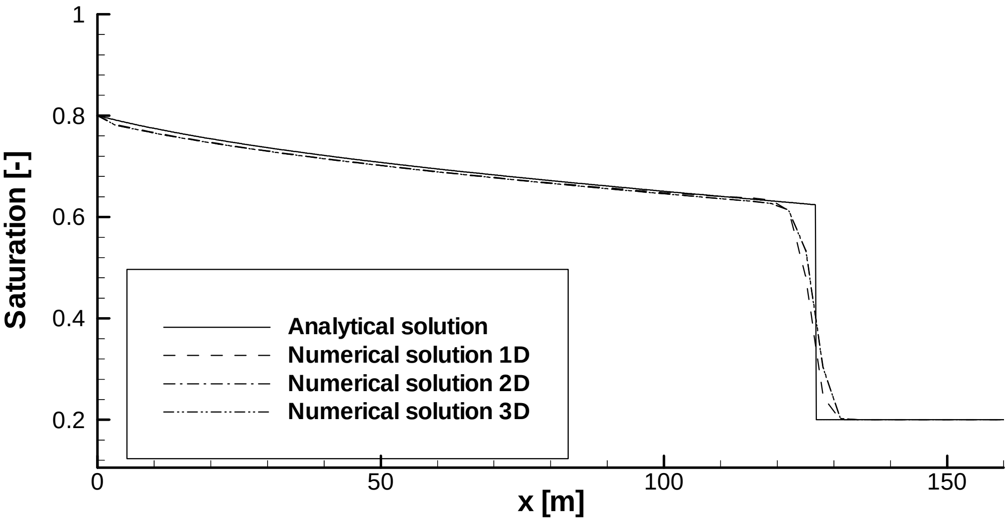


Fig. 4.6: Calculated saturation distribution for the Buckley-Leverett example. The solution for 2D and 3D elements are indistinguishable, the 1D solution differs slightly because of the different integration scheme for 1D elements.

4.3.4 Impact of the model-adaptive methods

The Buckley-Leverett example is chosen as a test case for the model-adaptive methods because an analytical solution is available and thus the impact of the methods on the accuracy can be evaluated. For these tests the same parameters as in the prior chapter are used.

The computations are performed with elements of all three dimensions. Additionally a reference case with no model-adaptation is regarded. The model-adaptive runs are performed with different values for the matrix recalculation parameters, in order to study the impact on both accuracy and speed in execution.

The model-adaptive downsizing is used for all system parts where one of the fluid mobilities dropped to zero. Furthermore, the matrix recalculation is connected to a relative change ε of the fluid mobilities. The following table shows the impact on the computation time.

Table 4.5: Impact of conditional matrix recalculation on the execution time for different accuracies.

Element dimension	Standard approach	Conditional recalculation. Accuracy:			
		$\varepsilon = 0.$	$\varepsilon = 10^{-3}$	$\varepsilon = 10^{-2}$	$\varepsilon = 10^{-1}$
1	100%	61%	55%	50%	39%
2	100%	42%	35%	32%	24%
3	100%	36%	27%	24%	17%

It is obvious that already for $\varepsilon = 0$ a substantial decrease in computation time is encountered. This is solely due to the downsizing of the physical model, i.e. ignoring irrelevant parts of the pressure PDE with no errors introduced. With increasing ε, an increasing number of matrices is kept from being rebuilt, correspondingly the execution time drops further. For even further increased values of ε the numerical scheme loses its stability. Figure 4.7 shows the results for the different recalculation scenarios.

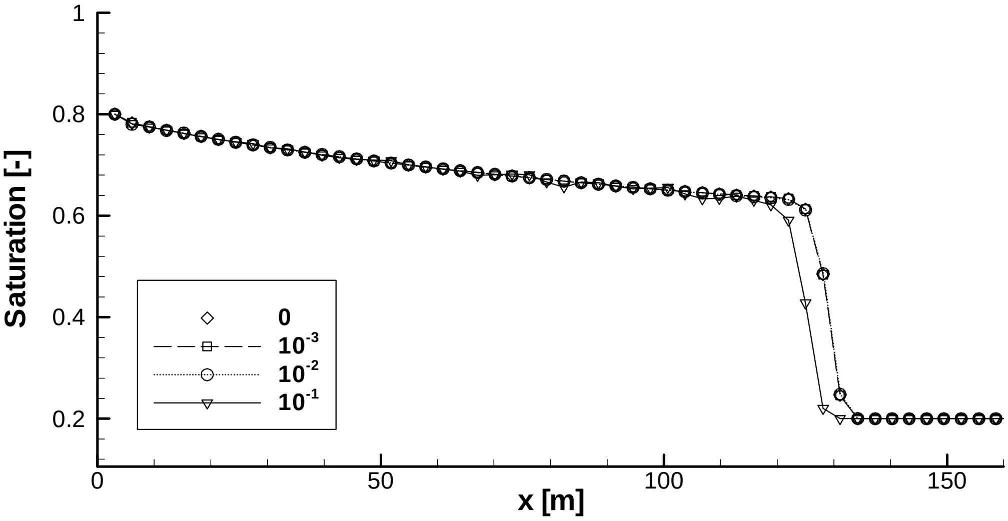


Fig. 4.7: Calculated saturation distributions for varying matrix recalculation parameters ε. For $\varepsilon = 10^{-3}$ and $\varepsilon = 10^{-2}$ the results cannot be distinguished from the standard solution $\varepsilon = 0$.

The results of the model-adaptive computation show the expected behaviour, as already discussed in Chapter 3.5. As presented in Figure 4.7, the results are excellent for both $\varepsilon = 10^{-3}$ and $\varepsilon = 10^{-2}$. For $\varepsilon = 10^{-1}$ the saturation front shows significant divergence from the other results. It is important to note that this divergence is in the 10% error margin that is introduced by setting ε to 10^{-1} (or 10%).

The prior statements show that the model-adaptive methods can significantly reduce the computation time, while the resulting errors are transparent to the modeller.

4.4 Multidirectional two-phase flow

4.4.1 Introduction

The so-called “5-spot problem” is a two-dimensional test case that originates in the oil industry (*Muskat*, 1949). A large oil field with a checkerboard of oil extraction and water injection wells is regarded. Here, the geometry and flow rates are adapted from the originally published units to the metric system. From symmetry considerations it becomes clear that two possible subsystems can be extracted from this field (Figure 4.8). The extracted sub-systems have impermeable boundaries and are driven by the oil extraction and water injection. The two systems should be numerically equivalent, but mesh effects can cause the results to be different.

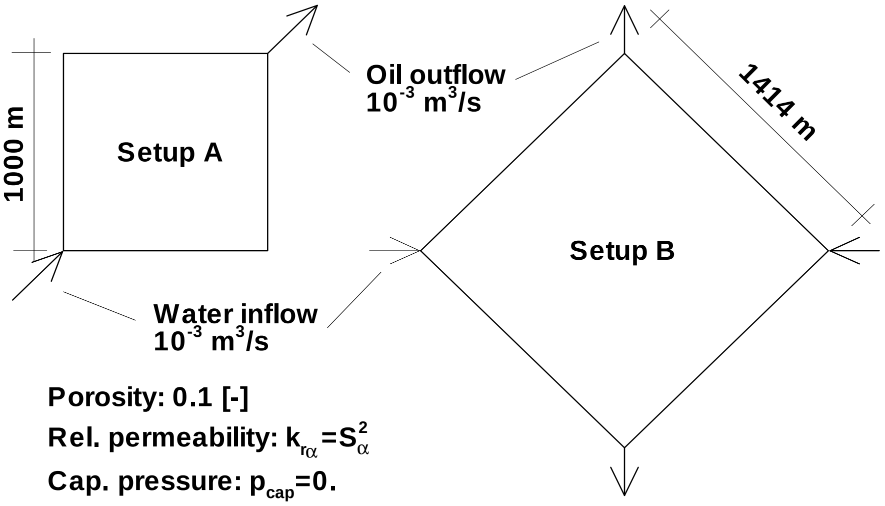


Fig. 4.8: Problem definition and extracted subsystems for the 5 spot problem. Both set-ups are equivalent due to the symmetry of the system.

4.4.2 Numerical simulation

The regarded systems represent a slice of one meter thickness, which was cut from the oil reservoir. The simulations are performed with fixed pressure for the extraction well(s), while a constant volume flux of water is injected into the other well(s). The fluid properties of water are assumed to be equal to the properties of oil. The numerical properties are chosen as described in the following table.

Table 4.6: Numerical parameters for the 5 spot test case.

Number of elements	Coarse: 64 Medium: 256 Fine: 4096
Node spacing	Set-up A: coarse 125 m, medium 62.5 m, fine 15.6 m Set-up B: coarse 176.8 m, medium 88.4 m, fine 22.1 m
Mass lumping	Yes
Upwinding	Full mobility upwinding
Time collocation	$\Theta = 0.5$
Time step length	Initial 1000 s, dynamically adapted
Final simulation time	$2 \cdot 10^7$ s

The results of the simulations for both set-ups are presented in Figure 4.9. Even the coarse grid simulation produces a useable solution, though the results on the finer grids are considerably better. It is oscillation free and shows a good mass balance. The reproduction of the front steepness is considerably better on the finer grid.

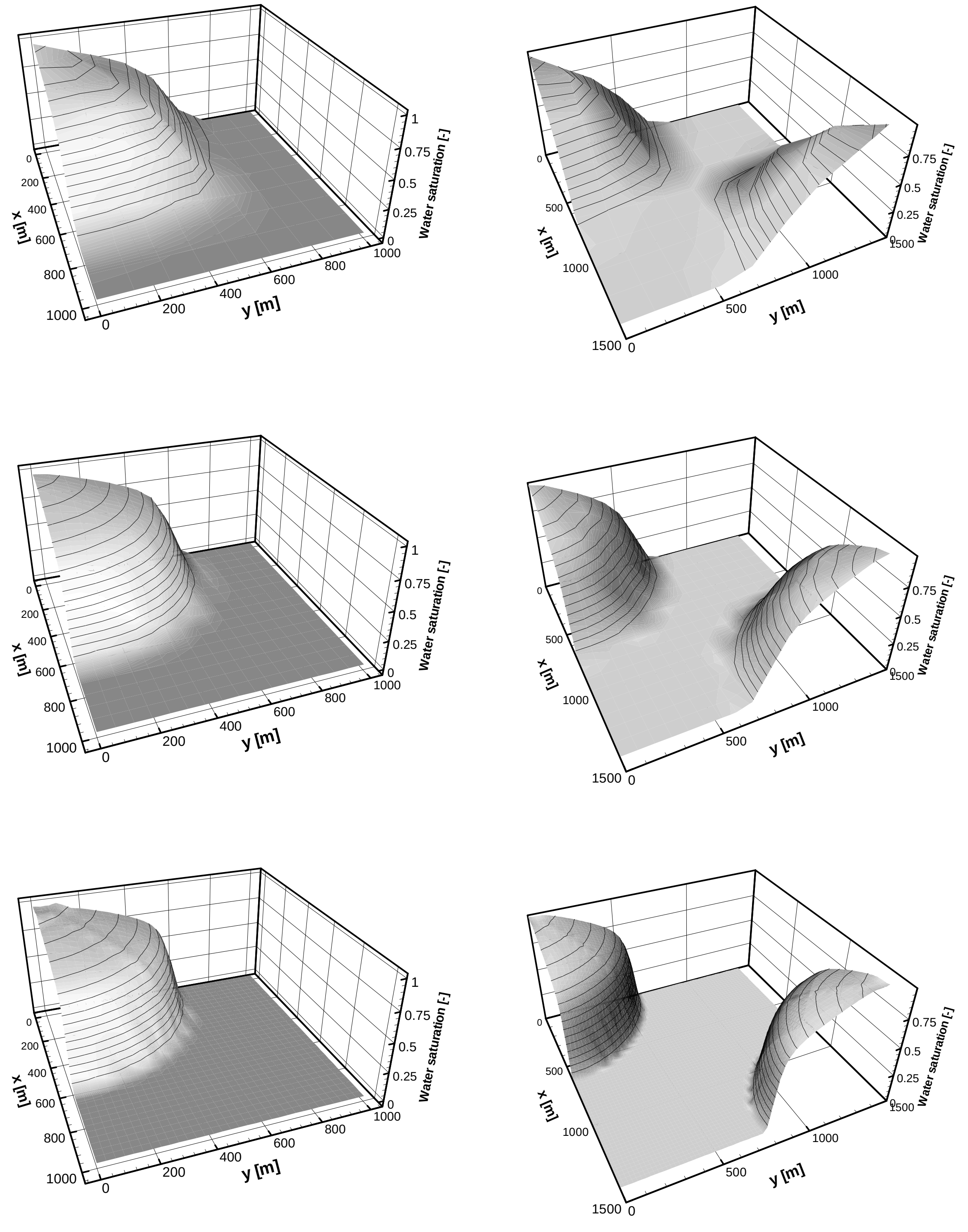


Fig. 4.9: Results for the 5-spot problem. The results for set-up A are shown on the left side, the results for set-up B on the right side. The upper two pictures show the results for grids with 64 elements, the middle ones with 256 elements and the lower ones with 4096 elements.

4.4.3 Grid orientation dependency

As a grid dependency test, the results for a cross-section along one of the boundaries for set-up A are compared to the diagonal cross-section for set-up B. Due to symmetry considerations, the results should be equal for both set-ups if the grids are sufficiently refined. As shown in Figure 4.10, a good agreement is reached, though the grid is turned by 45° between the two simulations.

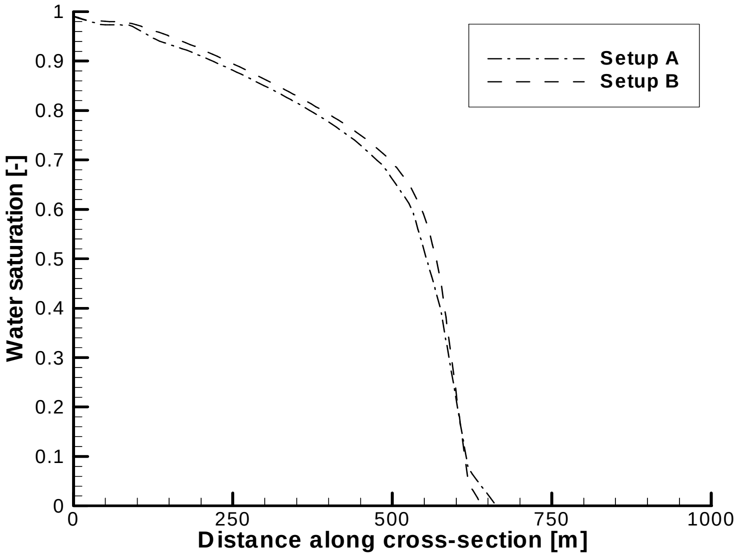


Fig. 4.10: Comparison of cross-sectional results for 5-spot problem on fine grid for both set-ups. The results are close to each other, thus the solution is independent from the grid orientation.

4.4.4 Impact of upwinding schemes on the results

As already pointed out in Chapter 3.1.6, the upwinding procedures can have significant impact on the results. Naturally the modeller on the one hand wants to achieve stable results, while on the other the numerical results should not be smeared too much. In the following set-up A will be examined on different mesh resolutions and with different upwinding schemes. The simulations are performed on grids with 256, 1024 and 4096 elements, respectively.

The comparison of the Gauss-Point-Upwinding (GPU) with the Fully-Upwinding (FU) scheme (Chapter 3.1.6) shows differences in numerical stability and quality. For the GPU scheme it is necessary to distribute the singular water injection point onto a larger area, because otherwise a single peak in the saturation field near the injection point is encountered. For this reason, a source with a radius of 100m is applied. The injected stream is distributed with a weighting $w = (1-d/100)^{0.25}$ as a function of the distance d from the proposed injection point.

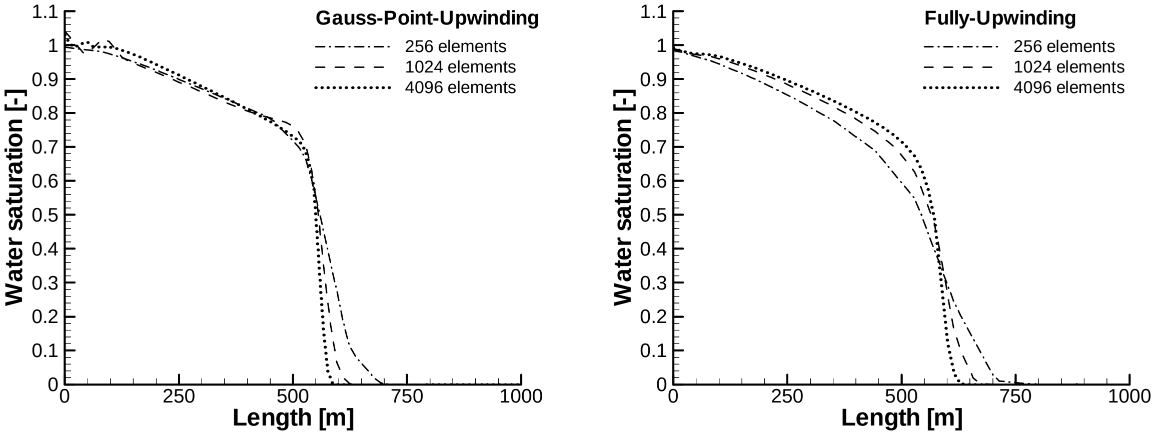


Fig. 4.11: Comparison of cross-sectional results for 5-spot (set-up A) with different upwinding schemes.

In Figure 4.11 the results of the different upwinding schemes are compared on different grid sizes. The GPU scheme still shows some small oscillation problems around the injection area for the medium size grid, while the FU scheme produces absolutely smooth results. This excessive smoothing on the other hand is the reason why the coarse grid results are better for the GPU scheme in the region behind the front. Finalizing, it should be investigated in future if it is useful to switch between the upwinding schemes dynamically.

4.4.5 Model adaptive methods vs. grid adaptation

In order to compare the results of the model adaptive methods and the grid adaptation, the simulations for set-up A are performed on a fixed grid with 1024 elements with and without the model adaptive methods and on an adaptively refined grid with 16 start elements.

The model adaptive matrix recalculation is triggered by relative changes of 1% for the phase mobilities, thus allowing approximately 1% error in front advancement speed. The grid adaptive methods are run in “default” mode, a maximum refinement depth of 3 levels is allowed, resulting in a minimal grid spacing that is equal to the grid spacing that is used for the model adaptive methods.

Table 4.7: Numerical effort for different methods

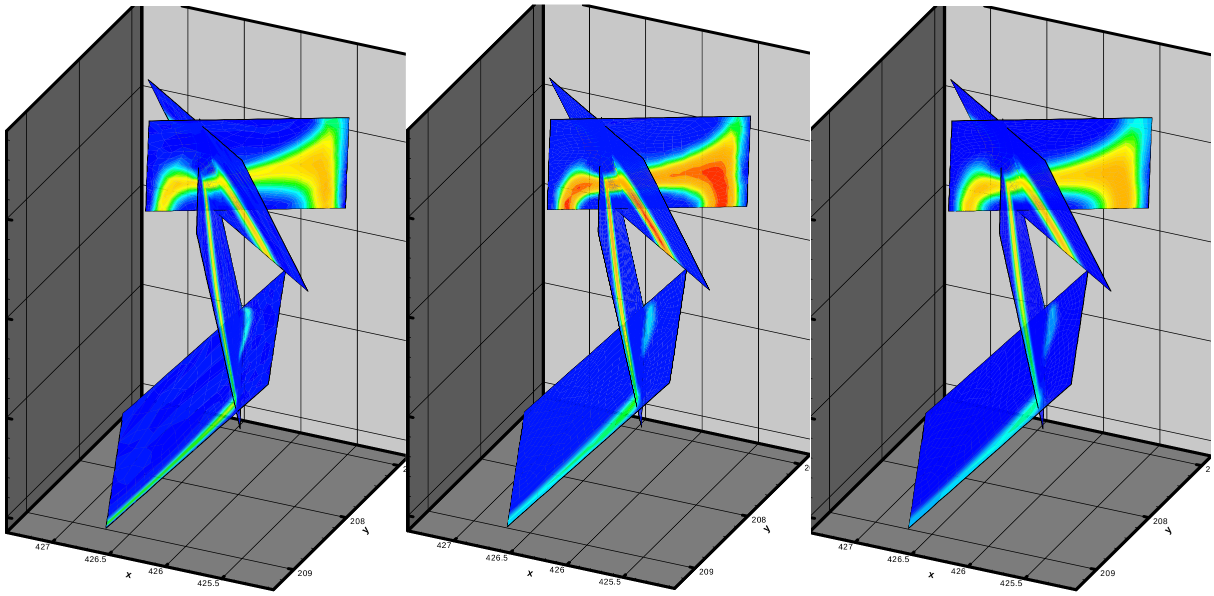
Numerical method	Relative CPU time
Standard	100%
With predictor	81%
With model adaptivity and predictor	26%
With grid adaptivity and predictor	21%
With grid adaptivity, model adaptivity and predictor	12%

Table 4.7 compares the computational effort for the standard method to the different adaptive methods. Obviously both adaptive approaches result in a significant decrease of numerical effort. The quality of the results is unchanged for both adaptive approaches, thus they are not presented here. Both methods seem to harmonize not too bad with each other, as the combination results in a further decrease of the required CPU time. Though the very substantial decrease in computational effort for the single application of the methods cannot be extrapolated to using both of them, the combination results in an even further reduction of computational effort.

4.5 Advective tracer transport in fracture networks

As test applications for the method of characteristics scheme presented in Chapter 3.2, fracture data from two different sites is used. The first example is based on the geometry data of a network of small fractures in the near field of a tunnel in the Grimsel Hard Rock Laboratory (Switzerland). It consists of four fractures, the geometry data was provided by Dr. H. Shao. The second example is based on data from the Münchehagen site (Germany). It consists of eleven fractures, the geometry data was provided by Dr. G. Kosakowski. The simulations presented here do not resemble any actual experiments, but are supposed to demonstrate the performance of the numerical scheme in comparison to established ones.

Figure 4.12 shows tracer distributions of an injected tracer pulse in the system of 4 fractures. The system is discretized with 3327 nodes. The tracer is injected at the lower left corner and an outlet is set at the upper right corner of the network. The dispersion length is set to very small values, taking into account that the main dispersion effects do not result from the single fracture dispersion, but from the network effects.



Distribution after 500s Operator-Splitting
FEM (SUPG)
dt = 0.9s
c_{min}/c_0=-0.14

Distribution after 500s
Method of Characteristics
dt = 25s
c_{min}/c_0=0.

Distribution after 500s
Operator-Splitting
Characteristics + FEM (PG)
dt = 25s
c_{min}/c_0=-0.09

Fig. 4.12: Comparison of tracer breakthrough for a 4-fracture system (Grimsel) with different numerical methods.

All methods give a similar distribution pattern, but the tracer peak is a little too dispersed in the operator splitting scheme. In the finite element simulation significant undershoots of up to 14% of the maximal concentration are observed. In order to reduce these undershoots a finer mesh resolution would be required, resulting in larger computation times. The method of characteristics scheme shows no undershoots at all, while the mixed operator splitting scheme again produces some undershoots. The tracer breakthrough and the total distribution pattern are very similar for all methods.

Figure 4.13 shows the finite element mesh for a fracture system based on the Münchehagen data. The example consists of eleven fractures and is discretized with 10922 nodes. Again the tracer transport is computed with the different numerical methods.

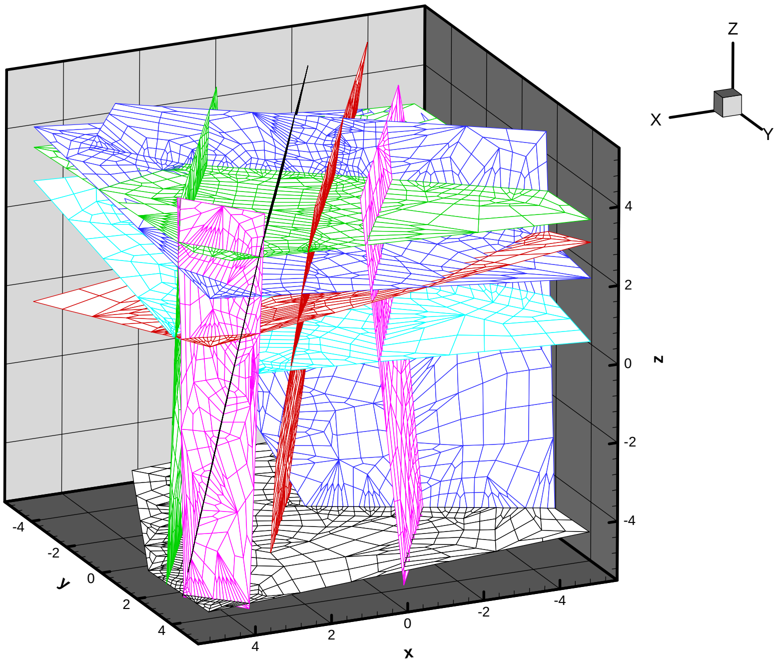


Fig. 4.13: Mesh for a system of eleven fractures (Münchehagen).

Figure 4.14 allows the comparison of results for the peak breakthrough. The purely advective Lagrangian method computes very sharp peaks, whereas the FEM solution smoothes the front. The operator splitting scheme shows results between the two, as it is expected. Interestingly the Lagrangian scheme can reproduce one of the typical transport effects within fracture networks, the tailing effect (i.e. a non-gaussian type of breakthrough curve), though no dispersion is regarded for the single fractures. This means that the network dispersion by macro scale mixing processes starts to dominate over the neglected dispersion in the single fractures. For large fracture network systems this point of view, and therefore the MOC, might be sufficient for the characterization of the dominant physical processes.

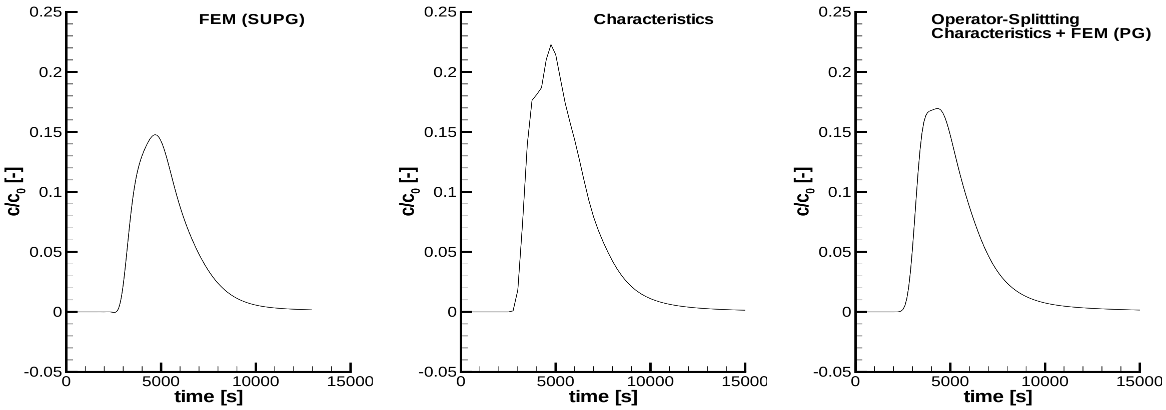


Fig. 4.14: Tracer breakthrough at three neighbouring outflow nodes for a system of fractures (Münchehagen) computed with different numerical methods.

The transport through the systems considered is highly advective ($Pe_{max} \approx 36$ for the first example, $Pe_{max} \approx 220$ for the second one). The Peclet numbers are computed on the basis of local (single fracture) dispersivities, because the much larger total dispersivity results from macro scale mixing in the network. For this kind of highly advective problems an operator splitting scheme is very effective. Table 1 compares the necessary CPU times for the simulations mentioned above.

Table 4.8: Comparison of relative CPU time for tracer transport simulations with different numerical methods.

Numerical method	4 fractures, 3327 nodes, steady flow	4 fractures, 3327 nodes, unsteady flow	11 fractures, 10922 nodes, steady flow
FEM (SUPG)	100%	100%	100%
Characteristics	3.6%	9.5%	1.4%
Operator splitting: FEM (PG) + characteristics	5.5%	9.8%	2.5%

The results demonstrate that the presented extended MOC scheme is suitable for transport simulations in fracture networks. Macro scale (i.e. network) mixing processes in the network can be reproduced and by the combination with the FEM an operator splitting scheme is obtained, which is able to simulate diffusion, dispersion, sorption and decay in an efficient way. For highly advective problems (Pe >> 1) the MOC is very efficient. For transport processes of tracers in large fracture networks the macroscopic mixing dominates over local diffusion and dispersion in single fractures, so that the very fast characteristics scheme is sufficient to describe the main processes. If local fracture dispersivities, retardation or sinks or sources for the tracer shall be regarded, it is necessary to use the operator splitting scheme.

4.6 Density driven flow in unsaturated media

4.6.1 Introduction

Clearly, there is a lack of experimental measurements on density-driven flow in partially saturated and coupled saturated/partially-saturated systems, which are suitable for verifying numerical models. Here an effort to address this problem is presented, by combining experimental and numerical investigations of density-driven flow patterns in coupled saturated/partially-saturated systems. First, a series of laboratory-scale experiments is considered to measure saltwater-freshwater interactions, particularly in the interface region between the saturated and partially saturated zones. The boundary conditions used in the experiments are selected on the basis of preliminary simulations from the numerical model. The experimental measurements are then used to test the numerical model. Particular emphasis is placed on checking the validity of flow phenomena predicted by the model in the partially saturated region above the water table, and along the interface between the saturated and partially saturated regions.

4.6.2 Experimental set-up

The laboratory experiment employed a perspex flow cell with internal dimensions of 0.958 m (length) x 0.478 m (height) x 0.105 m (width). Figure 4.15 presents the system before it was filled with sand.

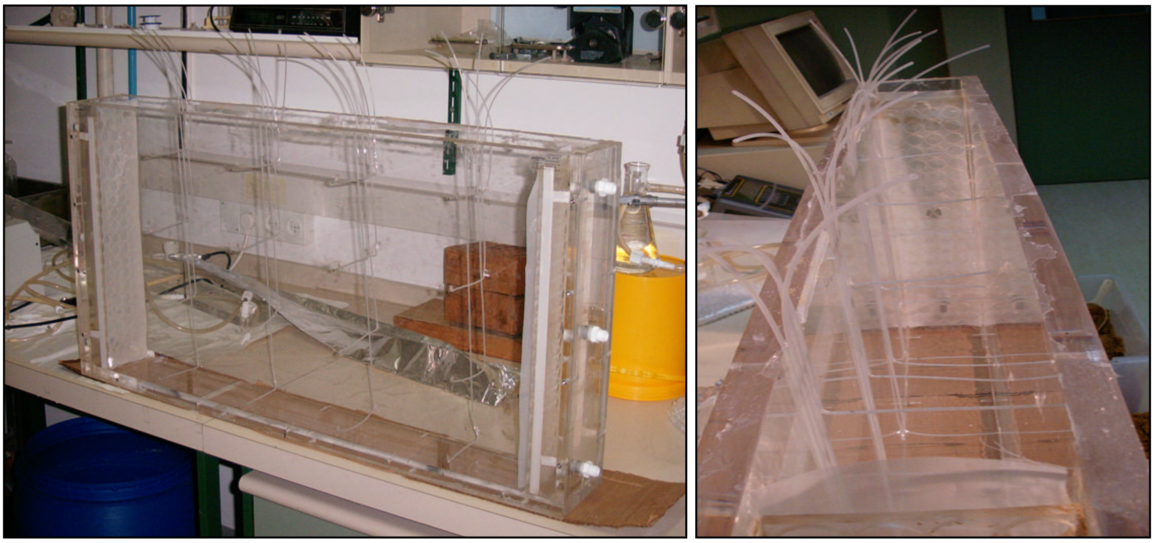


Fig. 4.15: Sandbox before filling with sand. The right picture shows the injection tubes through which samples were taken and the tracer was injected into the system.

It was equipped with two perforated sidewalls so that an inner length of 0.865 m remained. Sixteen tubes of 1 mm inner diameter were placed horizontally throughout the system to allow for tracer injection and sample extraction (Fig. 4.16). Each of the tubes was closed at one end, perforated along a length of 10 cm and bent upwards at an angle of 90 degrees. Thus a relatively equal injection over the system thickness could be achieved and, together with the symmetry arising from the impermeable front and back plate, the system can be simplified to a 2D system for the numerical investigations.

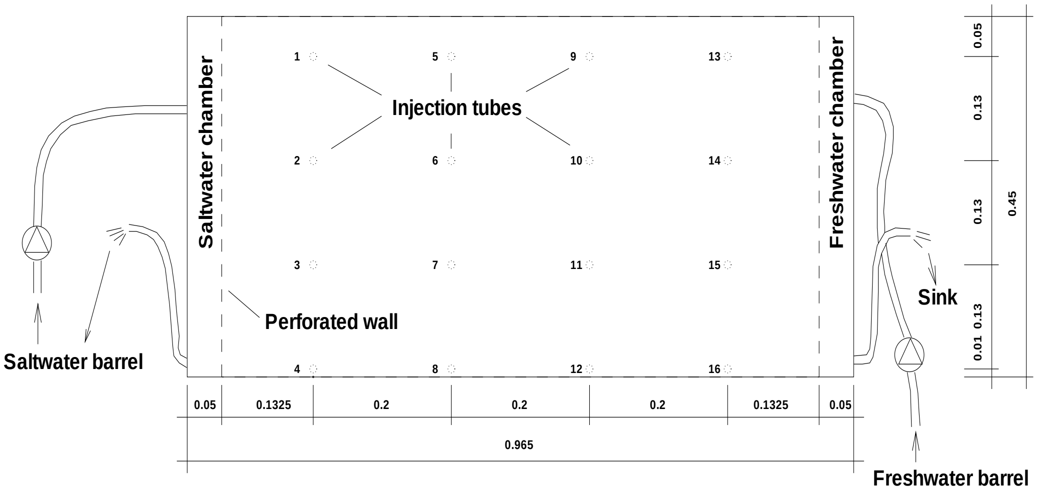


Fig. 4.16: Geometry of the sandbox.

The filling material was a clean, natural, silica sand local to the Rehovot (Israel) area. The sand was sieved to a range of 0.25 mm to 0.50 mm, and then washed and oven-dried at 95ºC. In order to achieve a relatively homogeneous packing, the flow cell was filled with dry sand “increments” of 2 cm thickness. Each of the increments was homogenized with those below to a depth of 10 cm, by using a mixing rod to disrupt any possible layering, and then compacted mechanically to reduce consolidation during the experiments. The 16 sampling/injection tubes were emplaced during this packing process. The flow cell walls bent outwards slightly because of the filling material, introducing a minor error of up to 2 mm for the internal width of the flow cell. After completion of sand filling, the flow cell was flooded with CO_2 through the lower row of injection tubes. Subsequent slow water saturation of the system from the bottom led to a virtually bubble free system, as CO_2 dissolves easily in water. To minimize hysteresis effects, the flow cell was then run through multiple imbibition-drainage cycles. Finally, the sand was covered to avoid evaporation across the upper boundary, to mimic a system at greater depth. The room temperature was held constant at 25ºC.

The porosity of the sand was measured to be 0.36 by taking samples of water-saturated sand and comparing the weights before and after oven-drying for a fixed volume. The intrinsic permeability of the sand was estimated to be about 4.7×10^{-11} m^2, by measuring the flux for a small gradient under saturated conditions. Samples of the sand at different heights above the fixed water table were taken to measure the water content. From these estimates, a capillary pressure – saturation relationship could be fitted, on the basis of the well-known *van Genuchten* (1980) curve (Fig. 4.17).

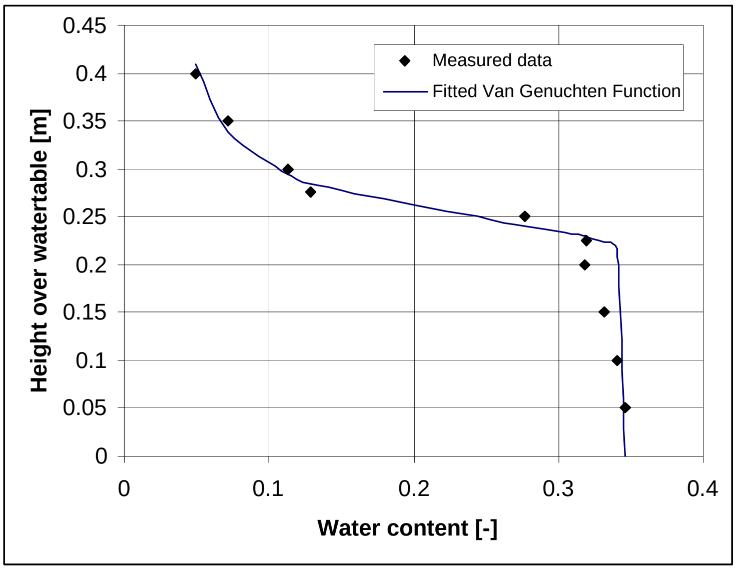


Fig. 4.17: Water content versus height for the sand used in the experiments.

Constant head boundary conditions were maintained for the experiments by filling the intake chambers of the flow cell with constant flux (100 ml/min) peristaltic pumps and using an overflow system. During the experiments this set-up produced stable hydraulic heads, with measurable head fluctuations of about ±0.5 mm. Several plastic barrels were used as water reservoirs. Regular tap water was used for all of the experiments; the barrels served as buffers to homogenize possible changes in the properties of the water (density of the freshwater was measured to be 1.067 kg/m^3). The overflow of the intake chamber connected to the saltwater inflow was recycled, whereas the overflow on the freshwater side was not reused, due to possible density changes. An electrical conductivity meter was attached to the recycled saltwater as an on-line monitor of the salinity (density of the saltwater was measured to be 1.097 kg/m^3). After adjusting the density once, the reference conductivity was used as a standard for further adjustments.

Tracer injections (red dye) were used to visualize the flow pattern. The dye was tested against a chloride tracer and showed no measurable retardation. Injection of dye in the two inlet/outlet chambers showed significant stratification, due to the saltwater inflow and to the boundary conditions used in the inlet and outlet chambers. On the saltwater side this stratification was eliminated by use of a recycling system within the inlet flow reservoir. On the freshwater side a small peristaltic pump was used to extract any bottom layer of saltwater that might develop.

All tracer injections were performed in a way that minimized the impact on the stabilized flow field. The density of the tracer was matched to the density in the flow cell prior to any injection. This was achieved by using a mixture between tracer matched to the density of freshwater and tracer matched to the density of saltwater. A small amount of water was extracted through each of the injection tubes and the density was estimated from the electrical conductivity of each sample. Following each injection, the tubes leading to the perforated front ends were cleaned by injecting water of the appropriate density. In order to have a minimal impact on the equilibrium in the partially saturated zone, the amount of previously extracted water equalled the total amount of liquid injected afterwards. This procedure virtually eliminates the far-field impact of the injection, but as the extracted water is preferentially drawn from below the injection tubes (because of the higher saturation and therefore higher hydraulic conductivity and lower capillary pressure below the injection point), the injected tracer will settle in equilibrium below the injection tubing.

During the experiments time-lapse pictures of the tracer movement were taken with a programmable digital still camera. It was therefore possible to take pictures at fixed or random intervals with a prescribed set of exposure settings. For each series of pictures, a reference picture of the flow cell with no dye in it was taken. After the experiments the images were processed through several enhancement steps. The first image was inverted and blended into the following ones, so that only the change in the informational content was preserved. For this procedure a very steady lighting situation was required; the room was completely darkened to avoid the impact of sunlight on the pictures. Only the green channel of the images was used, as this channel showed the best signal-to-noise ratio against the red dye.

4.6.3 Experiments

In an *a priori* investigation of the flow system, using the numerical model described below in Section 4, five principal flow patterns were identified, depending on the water table heights (boundary conditions) at the two sides of the system. Prescribing the saltwater and freshwater inlet chambers as being on the left and right sides of the flow cell, respectively, the following five cases can be considered:

- the pressure is much higher on the right side of the flow cell over at all heights, so that the freshwater is flushed through the system;
- the pressure is uniform at the bottom on both sides of the flow cell, so that the saltwater forms a tongue which does not reach the freshwater side;
- the pressure is uniform at the midpoint between the base of the flow cell and the water table, on both sides of the flow cell, so that the saltwater reaches the freshwater side, and there is a counterflow of freshwater;
- the water table is approximately uniform on both sides of the flow cell, so that the saltwater breaks through the system, but the freshwater is not totally flushed out;

- the pressure is much higher on the left side of the flow cell over at all heights, so that the saltwater is flushed through the system.

The two (first and last) extreme cases were neglected, as they lead to trivial cases where only freshwater or saltwater remains within the system. The focus is therefore on three experimental set-ups, corresponding to the second, third and fourth cases:

- Set-up A: The pressure is uniform at the bottom on both sides of the flow cell.
- Set-up B: An intermediate state between Set-up A and Set-up C.
- Set-up C: The water table is uniform on both sides of the flow cell.

The experiments were conducted after approximately steady state conditions were reached. In each experiment, a series of conductivity measurements was performed along the lower rows of sampling tubes, in order to determine the movement of the saltwater. The upper two sampler rows were not used for this purpose, as they did not come into contact with the saltwater. The system was assumed to be in steady state if no further increase in electrical conductivity was observed in the sampling tubes after waiting for an additional period equal to 10% of the total time the experiment was running.

4.6.4 Numerical simulations

The numerical simulations of the experiments discussed in Chapter 4.6.3 incorporated parameter values that could be measured directly from the experimental set-up (Table 4.9).

Table 4.9: Overview of parameter values of the laboratory experiment.

Porosity	$n = 0.36$
Permeability	$k = 0.5 \times 10^{-10}\ \mathrm{m}^2$
Residual saturation	$S_{res} = 0.15$
Maximum saturation	$S_{full} = 1.0$
Relative permeability	$k_r = 0 \le \left(\frac{S - S_r}{S_{max} - S_r} \right)^4 \le 1$
Diffusion constant (salt)	$D_m = 4.0 \times 10^{-10}\ \mathrm{m}^2/\mathrm{s}$
Diffusion constant (tracer)	$D_m = 4.0 \times 10^{-10}\ \mathrm{m}^2/\mathrm{s}$
Longitudinal mass dispersion length	$\alpha_l = 5 \times 10^{-4}\ \mathrm{m}$
Transverse mass dispersion length	$\alpha_t = 5 \times 10^{-5}\ \mathrm{m}$

Fluid viscosity	$\mu = 1 \times 10^{-3}$ kg/(ms)
Fluid density (freshwater)	$\rho_{fresh} = 1067$ kg/m^3
Fluid density (saltwater)	$\rho_{salt} = 1097$ kg/m^3

Other parameters, including the longitudinal and transverse dispersivities, fluid viscosity and the relative permeability function, were assigned values based on commonly-reported estimates appearing in the literature. The effective diffusivity (i.e., the product of free water diffusivity and tortuosity) was fitted numerically to observations of diffusive tracer spreading of the tracer during a separate experiment with no water movement. The diffusivity of the salt was assumed to be equal to the tracer diffusivity. The diffusivities were assumed to be constant in the range of the concentrations used (*Reid et al.*, 1988). The longitudinal dispersivity was set to the diameter of the largest sand grains, while the transverse dispersivity was assumed to be 1/5 of this diameter. Linear interpolation was used to quantify the dependency between salt concentration and fluid density (*Herbert et al.,* 1988). The viscosity was assumed to be constant in the range of salinities used in the experiments (*Spitz,* 1985). The dye concentration was very low, so the impact on the viscosity was neglected.

Table 4.10: Overview of numerical parameter values for the simulation of the laboratory experiment.

Number of nodes	36 nodes on initial grid, dynamically refined, limited to 15000 nodes
Time collocation	1 for pressure, 0.5 for transport
Time stepping	Fixed
Upwinding	SUPG for advection, $\alpha = 1$

The simulations were performed using, additionally, the numerical parameters shown in Table 4.10. An adaptive mesh refinement strategy was employed, which decreases the computational effort significantly. The starting grid was built with 25 elements, and a maximal refinement of 7 steps (each bisecting the element length) was allowed. The maximum number of nodes was restricted to 15000, which resulted in systems containing about 14000 elements. For comparison, a fixed mesh of comparable spacing would require about 400000 elements. Multiple simulations demonstrated that a refinement depth of 6 steps (resulting in approximately 6000 nodes) was sufficient to reach grid convergence.

The sharp interface that becomes established in Set-up B (see below) required an additional adjustment of the grid, using a streamline orientation based on initial simulations. The upper picture in Figure 4.18 shows the mesh for Set-up B after a simulation time of 37900 s.

The lower picture shows the rectangular mesh that was used for the Set-up A after a time of 21000 s. The mesh for Set-up C is not presented here, as it is similar to the mesh for Set-up A, though the local refinement is different. The grids were refined to accommodate the curvature of the pressure field, the saturation field, the salt concentrations and, with higher priority, the tracer field.

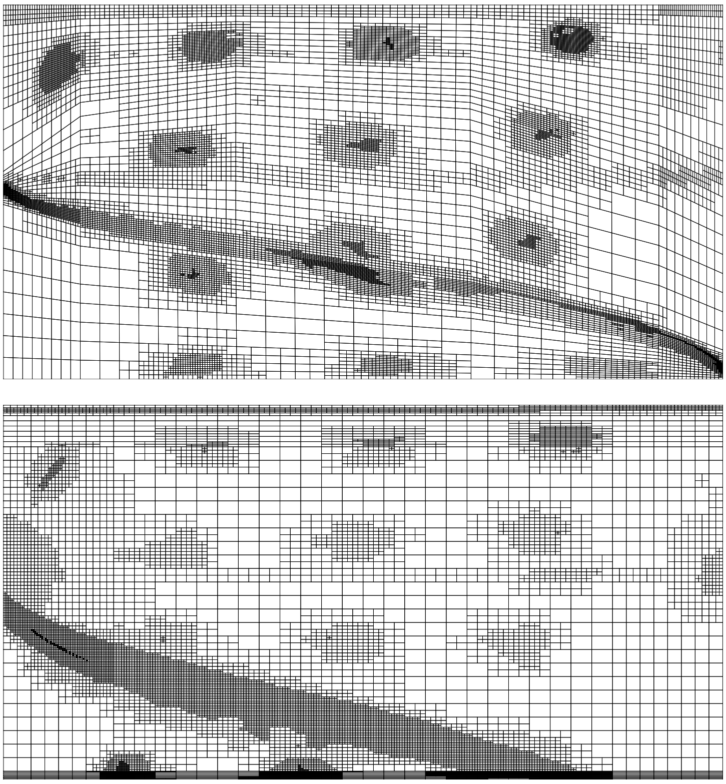


Fig. 4.18: Dynamically adapted finite-element mesh. The upper picture shows the streamline adapted grid for Set-up B, the lower picture the standard grid for the other set-ups.

The error criterion for changes in the variables considered in the nonlinear iteration was normalized according to the change in the variables in each time step. This approach avoids the accumulation of errors which might otherwise occur if the scheme chooses multiple smaller time steps instead of larger ones. The condition for the steady state form of the flow field was determined using the expression given in Table 4.10.

Dirichlet-type boundary conditions (BC's) were specified for the fluid pressures and the tracer concentrations (Table 4.11). These BC's were coupled to two additional conditions. First, the pressure boundary conditions were eliminated for all nodes where a pressure that equals the reference pressure of the Richards' approach does not result in a flux which leaves the system. In this way, the physically correct behaviour of a fluid at an open boundary can be reproduced. Second, the BC's for tracer concentrations were eliminated for nodes where fluxes leave the system. For those nodes the advective tracer transport fluxes are handled implicitly in the FEM. The tracer injections were treated by using short time Dirichlet BC's with a radius of 1 cm around the injection point; this value equals the amount of tracer injected through the tubes during the laboratory experiments.

Table 4.11: Applied Dirichlet-type boundary conditions.

	Set-up A	Set-up B	Set-up C
Pressure for x =0 m	2582.8 Pa for z = 0 m, declining linearly to 0 Pa at z = 0.24 m. 0 Pa for z>0.24 m if this results in a flux leaving the system	same as Set-up A	Same as Set-up A
Pressure for x = 0.865 m	2582.8 Pa for z = 0 m, declining linearly to 0 Pa at z = 0.247 m.	2543.5 Pa for z = 0 m, declining linearly to 0 Pa at z = 0.243 m.	2512.1 Pa for z = 0 m, declining linearly to 0 Pa at z = 0.240 m.
Salt concentration for x = 0 m	30 kg/m^3 if advective fluxes into the system occur	same as Set-up A	same as Set-up A
Salt concentration for x = 0.865 m	0 kg/m^3 if advective fluxes into the system occur	same as Set-up A	same as Set-up A

4.6.5 Experimental and numerical results

Presented below is a direct comparison of the results of the experiments and the associated numerical simulations. In the numerical simulations, tracer is injected at 16 points in the domain that match those used in the laboratory experiments. The three sets of flow conditions (Set-ups A, B and C; recall Section 4.6.3) are analysed in two ways: (1) by qualitative comparison of the tracer migration patterns from the injection points, at six different times, from photographs of the laboratory flow cell and corresponding “snapshots” from the numerical simulations; and (2) by quantitative analysis of the streamline patterns, which were extracted from the photographs and simulated snapshots. In all cases, the numerical simulations are seen to closely match the experimental results.

4.6.5.1 Set-up A

The system was defined so that the pressures at the bottom of the two inlet chambers were the same. This was achieved by adjusting the water table heights to 24.0 cm in the left (salt water) chamber and 24.68 cm in the right (fresh water) chamber ($p = \rho g h = 2583$ Pa; recall the fluid densities given in Table 4.9).

A detailed view of the observed and simulated flow patterns is presented in Figure 4.19 (left column the experiment, right column for the numerical results), which shows the movement of the injected tracer plumes for a total period of 30 hours. The numerical simulations resemble closely the behaviour observed in the flow cell experiment. The flow field can be separated into two main regions. In the lower region, the development of the saltwater tongue is clearly visible, along with a convection cell that develops. In the upper region, a significant counterflow area, which transports fresh water through both the saturated and partially saturated zones, can be observed. The flow velocities are quite small in the saltwater tongue, even in comparison to the partially saturated zone where a significant flow takes place.

Streamlines were extracted from the movement of the injected tracer in order to give an overview of the flow field dynamics (Figure 4.20) for the first 26 hours of the experiment. Comparison of the streamline plots (upper figure for the experiment, lower figure for the numerical results) shows that the behaviour of the saltwater tongue in the saturated zone and the freshwater counterflow in the partially saturated zone are simulated extremely well. The position of the front of the saltwater tongue can be estimated from the movement of the tracers injected in the tubes 4, 8, 12 and 16 (see Figure 4.16). In particular, the partial convection cell formed by the movement of the tongue is reproduced by the simulations. Also noticeable is the significant degree of flow in the partially saturated zone; the level of the water table is indicated on the figure.

The rate of tracer movement in the partially saturated zone is slightly overestimated in the numerical simulations (Figure 4.19). A sensitivity analysis with additional numerical simulations indicated that the location of the upper row of injection points relative to the assigned water content is a crucial parameter for the tracer movement. As is clear from Figure 4.17, the sharp transition in water content begins at heights above the water table in the range of 15 cm. Therefore, changing the height of the upper row of injection points by as little as 1-2 cm, and variations of a few centimetres for the transition zone, can have a significant effect on the predicted tracer pattern. The error during the experimental placement of the injection rows and changes of the transition zone during the multiple experiments are in this range. Additional analysis demonstrated that it is possible to get a closer match between experimental and numerical results by changing these values. These parameters do not change the principal behaviour of the system, and are therefore not discussed further.

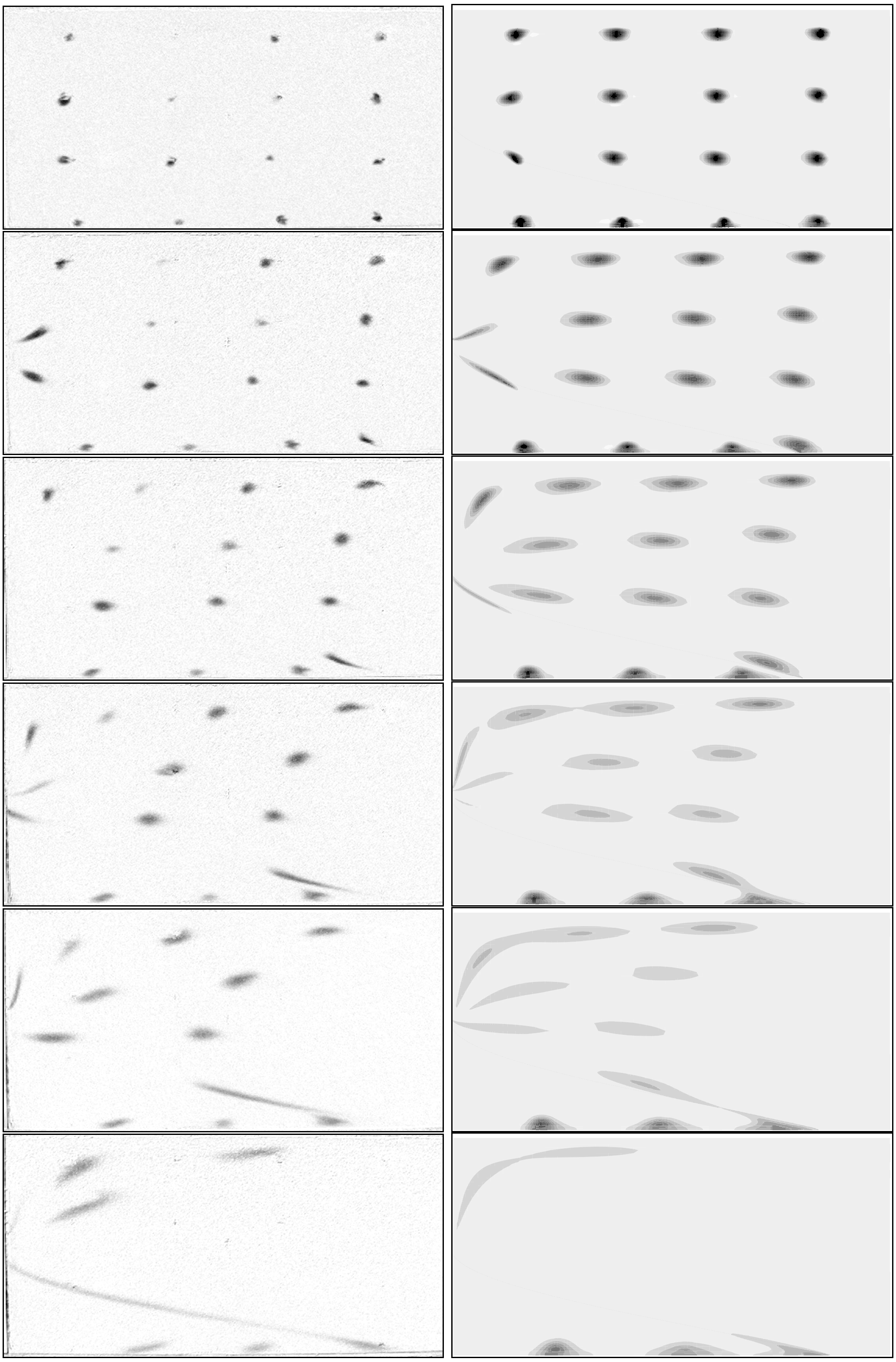


Fig. 4.19: Experimental (left column) and numerical (right column) tracer distributions at 0.75 h, 2.75 h, 5.75 h, 10.75 h, 16.75 h, 30.75 h after the experiment began. Horizontal boundary conditions: equal pressures at the bottom of the two inlet chambers (Set-up A). The dotted isoconcentration line shows computed 50% isosalinity.

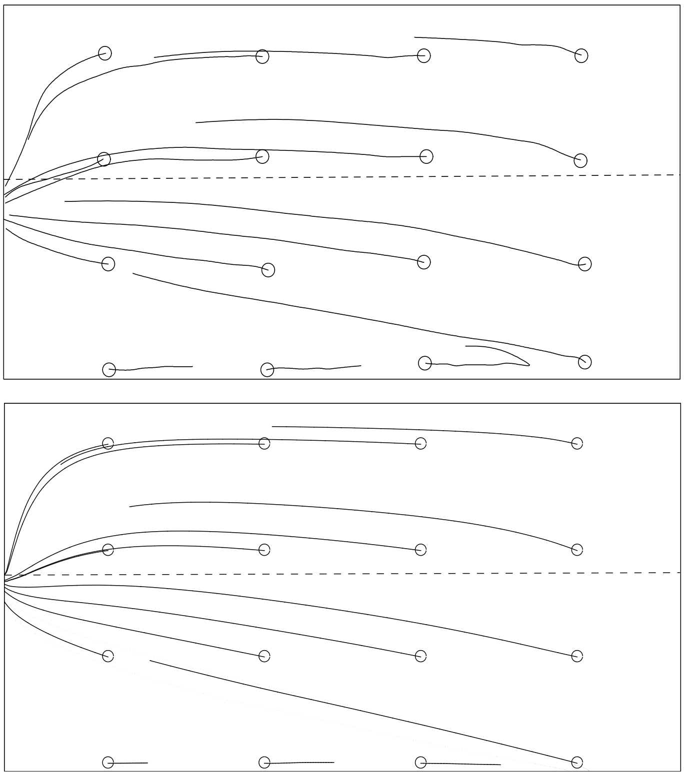


Fig. 4.20: Experimental (upper picture) and numerical (lower picture) results showing streamlines extracted from tracer movement shown in Figure 4.19. The location of the water table is marked by a dashed line. The dotted isoconcentration lines show computed 10% and 90% isosalinities.

4.6.5.2 Set-up B

For this intermediate case, the heads were adjusted to be 24.0 cm in the left (salt water) chamber and 24.3 cm in the right (fresh water) chamber. This results in an equal pressure at 13.3 cm above the base of the flow cell, on both boundaries:

$$\begin{aligned} p_{l,\,z=0.133\mathrm{m}} &= 9.81\,\mathrm{m/s^2}\cdot(0.240\,\mathrm{m}-0.133\,\mathrm{m})\cdot 1097\,\mathrm{kg/m^3} = 1151\,\mathrm{Pa} \\ p_{r,\,z=0.133\mathrm{m}} &= 9.81\,\mathrm{m/s^2}\cdot(0.243\,\mathrm{m}-0.133\,\mathrm{m})\cdot 1067\,\mathrm{kg/m^3} = 1151\,\mathrm{Pa} \end{aligned} \tag{4.9}$$

The observed and simulated tracer migration patterns are presented in Figure 6 (left column the experiment, right column for the numerical results). Fluid within the lower salt water tongue moves to the right (tracer injection points 3, 4, 7, 8, 12 and 16 in Figure 4.21), while the fresh water advances to the left; no convection cells develop. This interplay is especially clear from the tracer migrating from injection points 7 and 11: these two plumes nearly collide. In addition, significant flow through the partially saturated zone, as well as interaction at the interface between the saturated and partially saturated zones, can be observed. The numerical model reproduces these behaviours very well.

Streamlines were extracted from the movement of the injected tracer, for a period of 14 hours (Figure 4.22); these streamlines show that a counterflow regime with high velocities is established, leading to a “shearing” of the flow field. The simulated tracer migration within the saltwater-freshwater interface region showed substantial numerical dispersion problems on the rectangular grid. Subsequent use of a streamline-oriented grid increased significantly the quality of the simulations, resulting in a very good reproduction of the sharp transition zone.

An interesting effect is the “tilt” which is observed on the plumes injected through the tubes along the upper rows (Figure 4.21). This result can be attributed to the relatively large vertical spread of the plumes, with the upper part of the plumes being at lower saturation (and therefore effective permeability) areas than the lower part. The numerical model fails to catch this behaviour; a possible reason can be either local heterogeneities of the sand packing or a small error in the capillary pressure measurements.

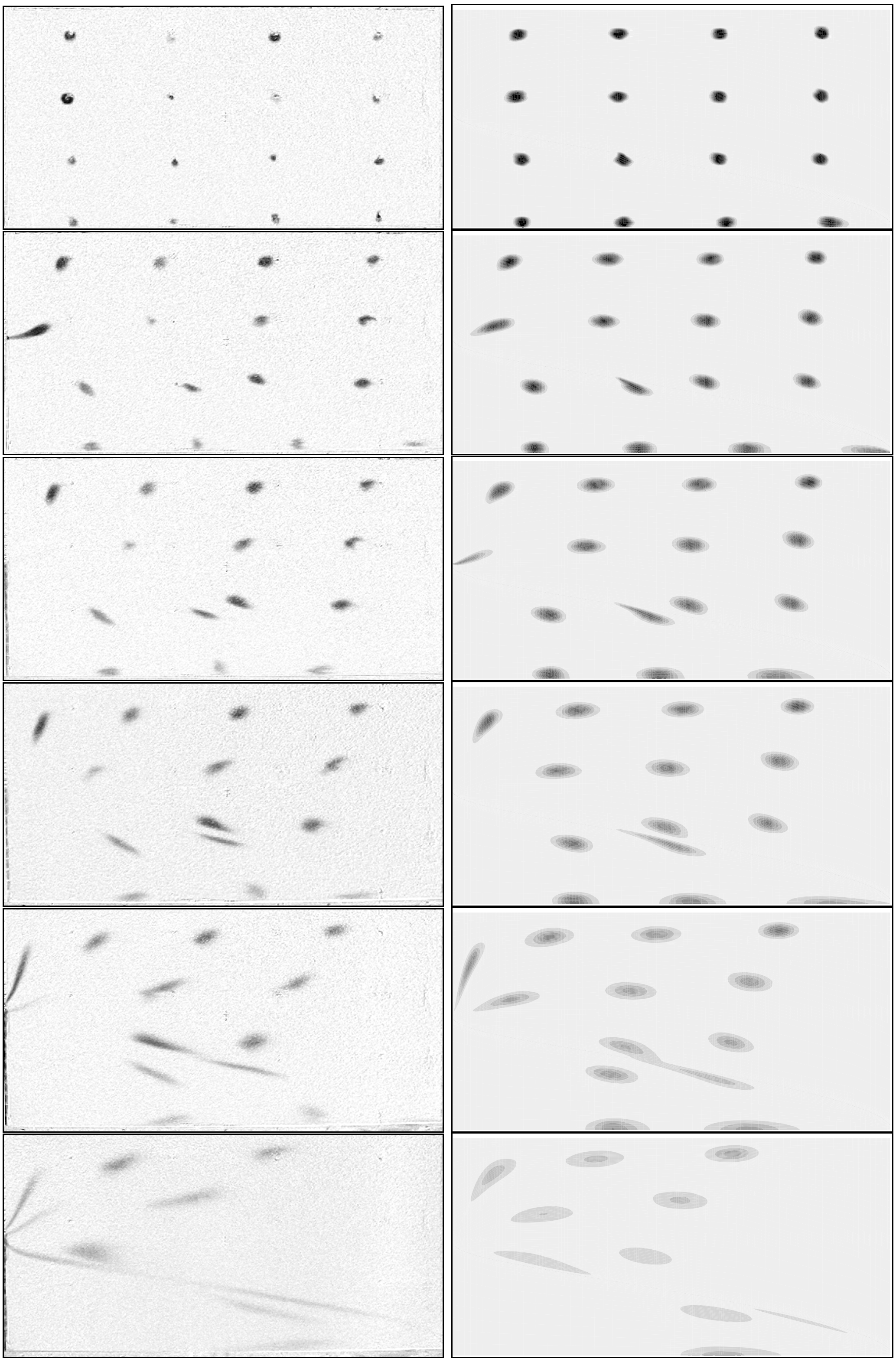


Fig. 4.21: Experimental (left column) and numerical (right column) tracer distributions at 0.25 h, 3.25 h, 6.25 h, 10.25 h, 17.25 h, 35.25 h after the experiment began. Horizontal boundary conditions: equal pressures at 13.3 cm above the base of the flow cell (Set-up B). The dotted isoconcentration line shows computed 50% isosalinity.

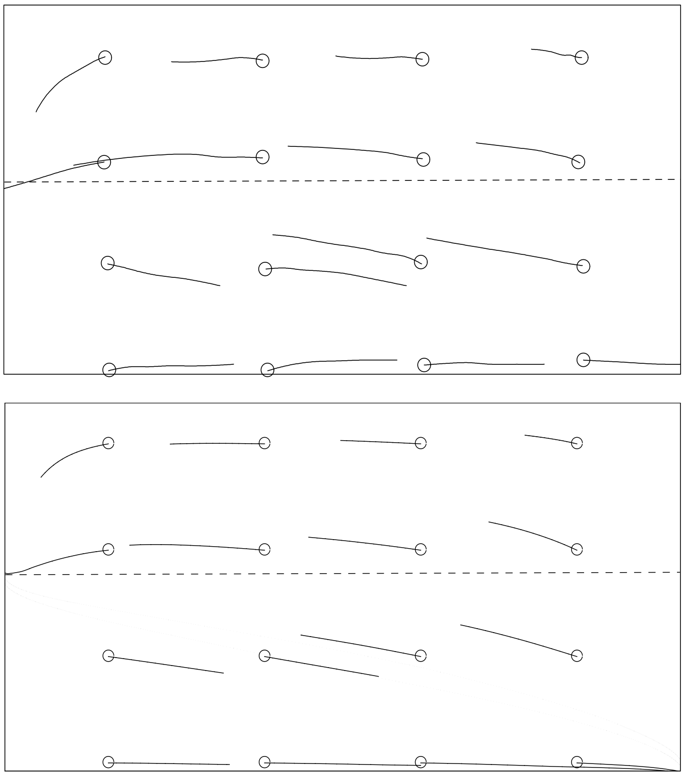


Fig. 4.22: Experimental (upper picture) and numerical (lower picture) results showing streamlines extracted from tracer movement shown in Figure 4.21. The location of the water table is marked by a dashed line. The dotted isoconcentration lines show computed 10% and 90% isosalinities.

4.6.5.3 Set-up C

In this case, the hydraulic heads were adjusted to heights of 24.0 cm on both boundaries of the flow cell. During the experiments, control of the heads was found to be critical, as even small changes in the heads can have a significant impact for this case. Some small fluctuations were evident during the course of the experiment, and the head was readjusted by +1 mm on the right side of the flow cell after 48 h. Furthermore the density adjustment of the tracer to the density of the fluid in the partially saturated zone proved to be crucial, as the flow velocities are very low and the impact of density differences was significant.

The observed and simulated tracer migration patterns are presented in Figure 4.23 (left column the experiment, right column for the numerical results), while streamlines extracted from the tracer movement patterns are given in Figure 4.24. The flow field is distinguished by two very different flow regions: whereas movement (mostly of salt water) in the saturated area is very fast, a slow anti-clockwise convection of fresh water develops in the partially saturated zone. The streamline plot of the experiment clearly shows that the flow field is separated into an area of high velocities in the saturated zone, and very slow movement in the partially saturated zone, into the opposite direction. After a period of about 24 h, most of the tracer in the saltwater toe was driven out of the flow cell, while very little movement in the freshwater area was visible.

In the experimental data the streamlines in the partially saturated zone initially point in a direction opposite to that predicted by the numerical model (compare the upper row of injection points in Figure 4.23). After the heads were re-adjusted at the flow cell boundaries (necessary because of minor fluctuations), the experimental and numerical results for the partially saturated zone are seen to match very well. Correspondence of the numerical simulations with the experimental results in the saturated zone is excellent. Some overestimation of the velocities in the partially saturated zone, by the numerical model, is evident (Figure 4.24). As noted above, this result can be attributed to the high sensitivity of the laboratory set-up to very small errors and variations in the heads at the boundaries. Tests with the numerical model confirm this sensitivity, and demonstrate the variability that can arise in this interface between the saturated and partially saturated zones.

With time, very slow movement in the partially saturated region can be observed, which forms one large convection cell (second and fourth rows of pictures in Figure 4.23). This result is interesting, as the water table height is equal on both sides of the flow cell, and no external pressure gradient induces this flow. It would appear that upward diffusion of salt on the left side of the flow cell, across the interface between the saturated and partially saturated zones, leads to a density-driven flow mechanism that allows the observed slow counterflow to develop.

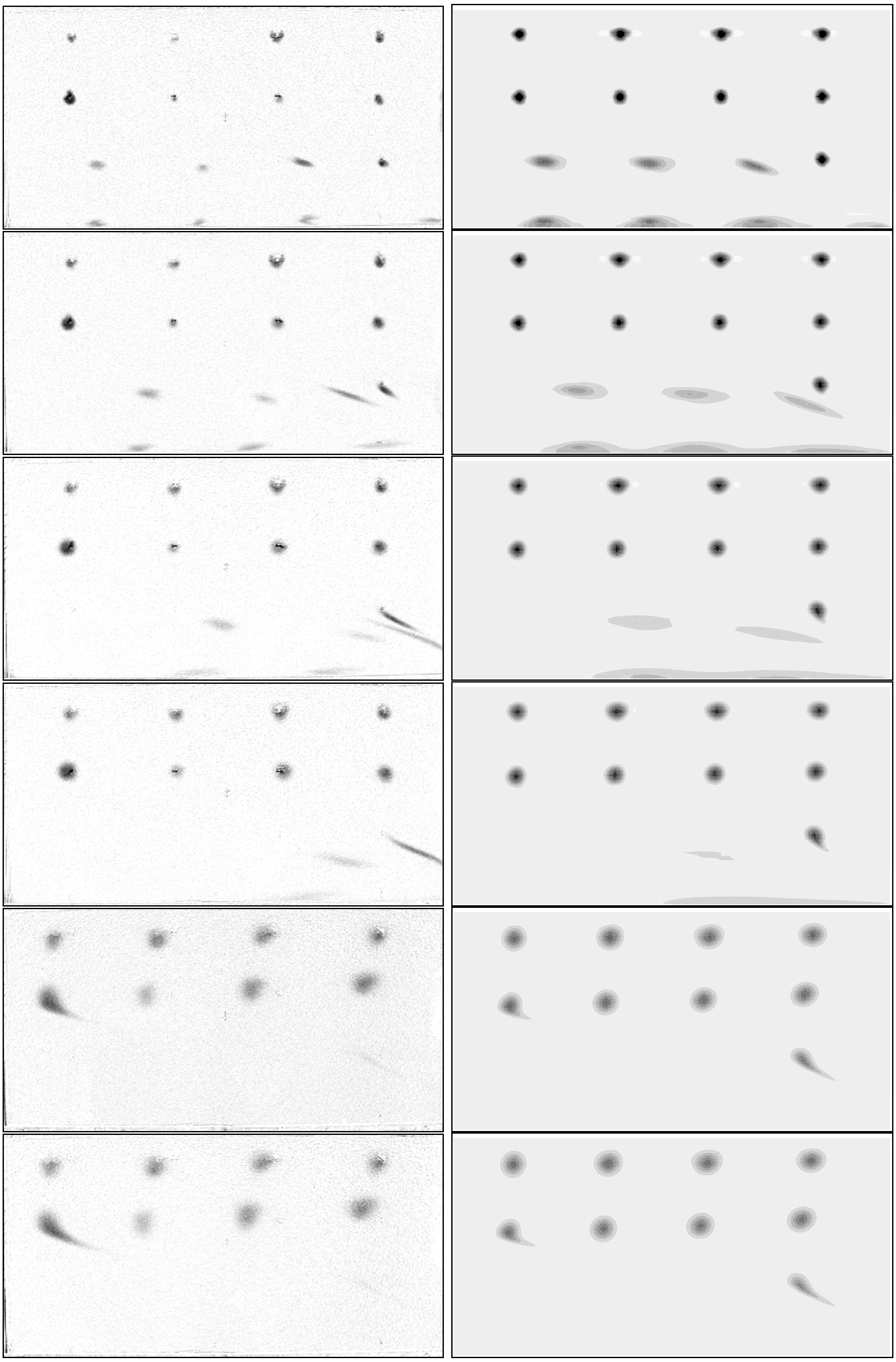


Fig. 4.23: Experimental (left column) and numerical (right column) tracer distributions at 2.5 h, 6.5 h, 11.5 h, 16.5 h, 44 h, 53 h after the experiment began. Horizontal boundary conditions: equal hydraulic heads (Set-up C). The dotted isoconcentration line shows computed 50% isosalinity.

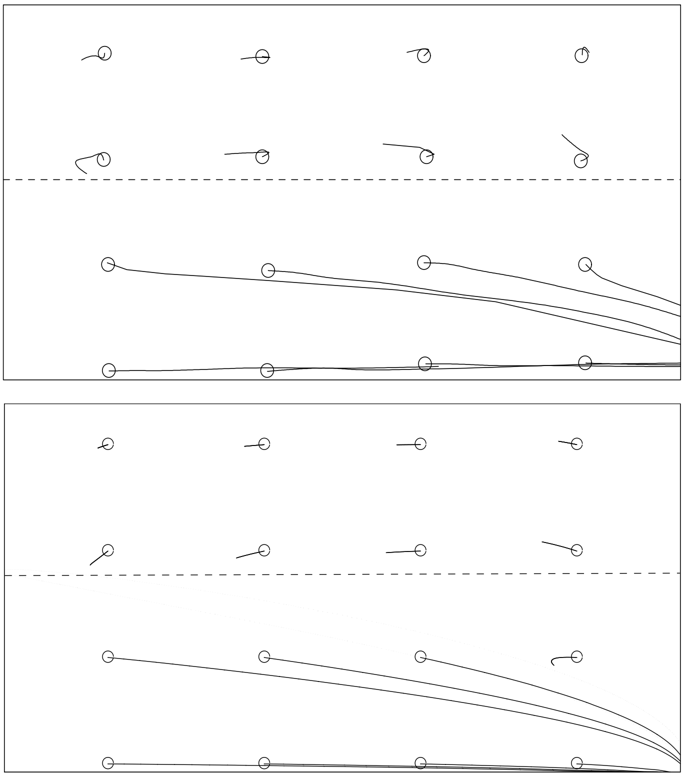


Fig. 4.24: Experimental (upper picture) and numerical (lower picture) results showing streamlines extracted from tracer movement shown in Figure 4.23. The location of the water table is marked by a dashed line. The dotted isoconcentration lines show computed 10% and 90% isosalinities.

4.6.6 Conclusions from the laboratory experiments

The laboratory experiments discussed here demonstrate clearly that a significant lateral flow, and coupled density-driven flow effects, take place in the partially saturated region above the water table, and at the interface between the saturated and partially saturated zones. Although it is difficult to extrapolate these small-scale experiments to the field scale, similar flow behaviours can be expected at the saltwater/freshwater interface in unconfined aquifers.

4.6.7 Impact of model and grid adaptivity on execution time and accuracy

In order to compare the influence of both model and grid adaptivity on the results, the calculations for one of the set-ups are repeated. Two different grid resolutions are regarded in order to evaluate the influence of different grid resolutions on the computation time and the accuracy. The coarse grid is obtained by applying four bisections on the original grid, thus resulting in 6400 elements. Equivalently four bisections of the original grid are allowed in the grid adaptive scheme, too. The fine grid consists of 25600 elements and is obtained after applying five bisections.

The results for the fine grid are not presented here, as all methods produced virtually equal results. In contrast, the computations on the coarse grid show significant differences. The results are presented in Figure 4.25. It is obvious, that the grid adaptive method suffers from numerical dispersion problems, while the standard approach can produce fair results even though a rather coarse grid is used. Even more astonishing is a comparison of the computation times (Table 4.12). The grid adaptive approach uses even more computation time than the standard approach. A closer look on the performance data during the simulation reveals, that the simulator has to perform two or three nonlinear iterations in each time step, because the grid adaptation has disturbed the previous equilibrium of the flow field. The model adaptive approach produces results that are indistinguishable from the standard approach, while significantly reducing the computation time. For the model adaptive and the standard approach the coarse grid results are already very close to the results on the finer grid, while the grid adaptive scheme requires further refinement of the grid.

Table 4.12: Comparison of computation time for the standard,
grid adaptive and model adaptive approaches.

Modelling approach	CPU-time (fine grid)	CPU-time (coarse grid)
Standard approach	100%	100%
Grid adaptive approach	110%	106%
Model adaptive approach	54%	51%

It must be stressed that these results can not be generalized. For finer meshes the results of the grid adaptive scheme are very good. Furthermore, the amount of needed computer memory is largely reduced with the grid adaptive approach.

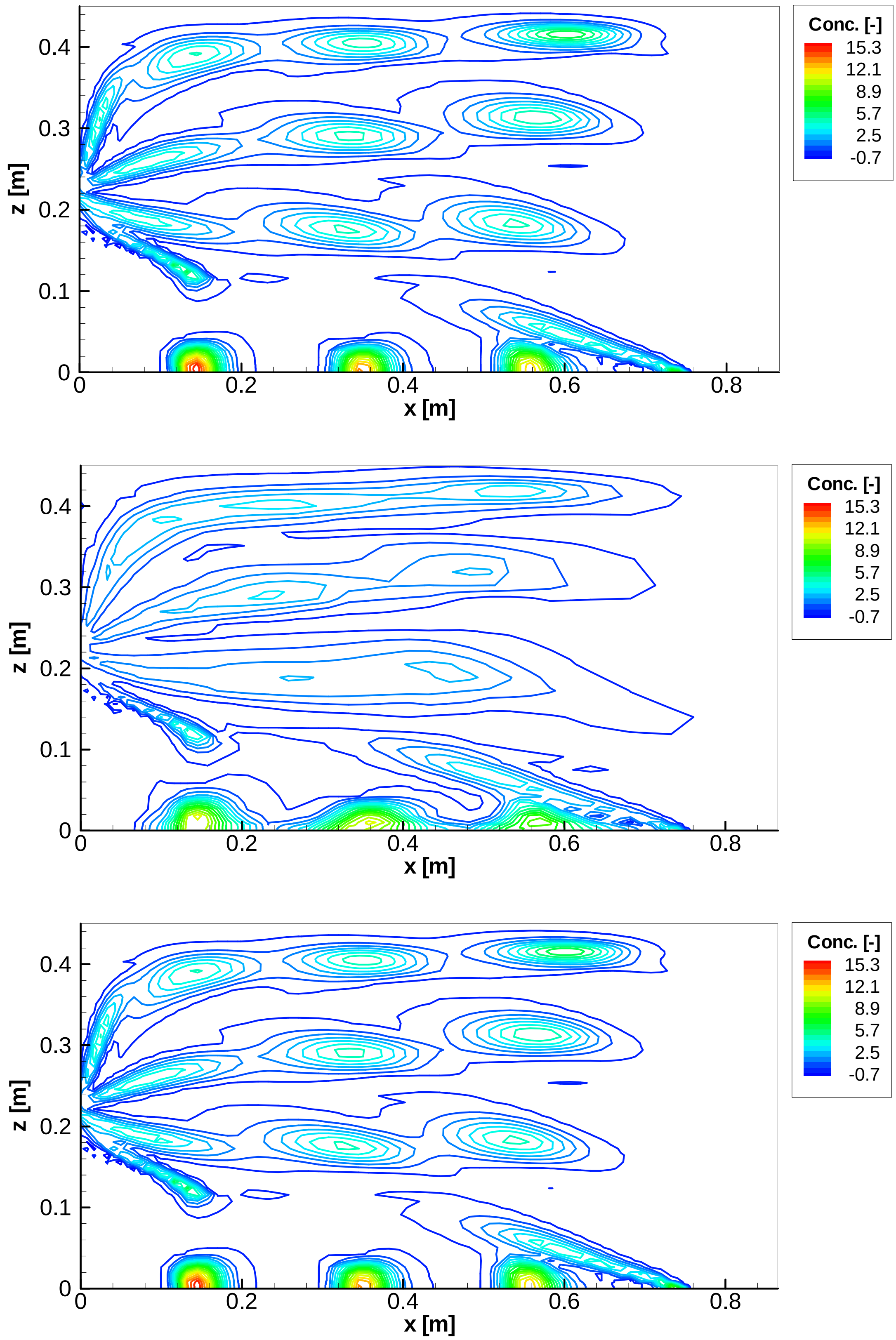


Fig. 4.25: These pictures show the computed tracer distribution for the coarse grid. The upper picture shows the result of the standard approach, the middle one the result with grid adaptivity and the lower one with model adaptivity.

4.7 Multiphase flow in a fracture network

This example presents the simulations of a packer test in a fracture network under multiphase flow conditions. The grid for this test case was generated by *Rother* (2001), based on geometry data that was supplied by Dr.-Ing. H. Shao (BGR). The physical data for the test are randomly chosen and do not reflect an actual test.

The physical behaviour of the regarded system is a severe test for the numerical model. Due to the extreme differences in the physical properties of the fluids (the density differs by three orders of magnitude and the viscosity by two orders of magnitude), any change in the saturation distribution results in significant changes in the pressure field. Subsequently, this results in severe convergence problems, which cannot easily be resolved. Anyhow the system is presented here, as it demonstrates the bounds for the useful application of the model.

Due to the low permeability of the surrounding rock matrix and the extremely high entry pressures for gas into the matrix, it can be assumed that gas does not enter the matrix. Thus, only the gas flow through the initially water filled fractures is regarded. The properties for the test case are presented in Table 4.13.

Table 4.13: Properties for the multiphase flow packer test

Fracture width	0.001 m
Porosity	1
Permeability	$8.3 \cdot 10^{-11}$ m^2
Initial water saturation	S(t = 0) = 99%
Rel. permeability - saturation relation	Linear for air, quadratic for water
Capillary pressure - saturation relation	Given by curve values

In the simulation it is assumed that the fractures are of limited extent and that the inflow and outflow only takes place at the dedicated boundary conditions. The position and the values for the boundary conditions are presented in Figure 4.26. The mesh is refined near the inlet and outlet in order to obtain a better reproduction of the applied boundary conditions. In the numerical model fluxes are applied as volumetric fluxes. Here, a mass flux is applied, because mass fluxes are very often used in experimental work. The mass fluxes are dynamically converted into volumetric fluxes in the model, according to the current pressure at the injection point.

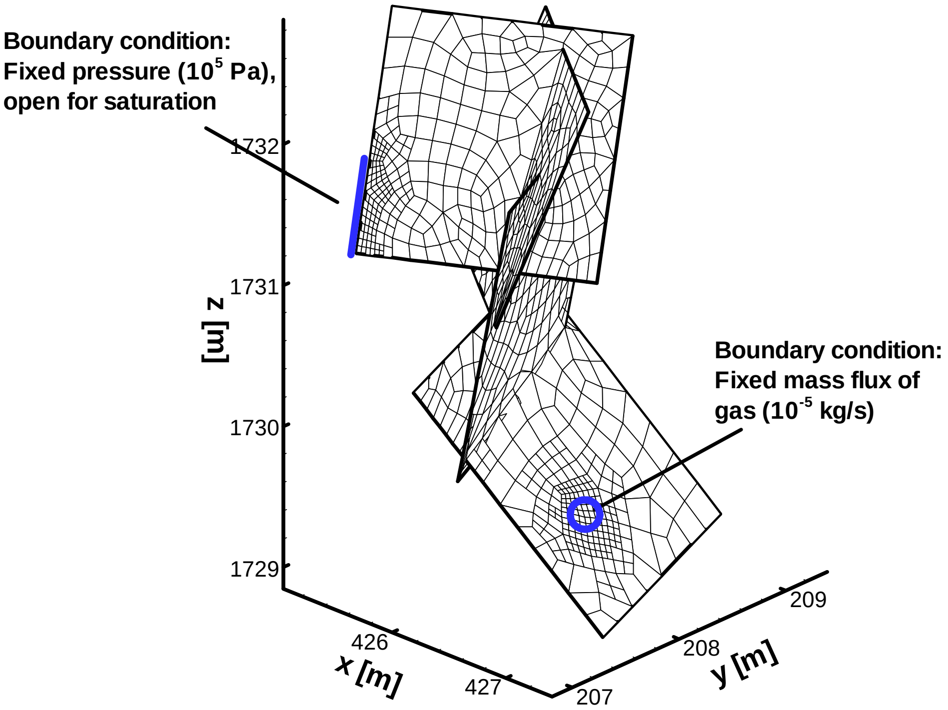


Fig. 4.26: Geometry and boundary conditions for the multiphase flow packer test.
The mesh consists of 1167 nodes and 1135 elements.

The results presented in Figure 4.27 show the saturations and Darcy velocities of gas in the system for different times. In the initial phase of the simulation, the gas is travelling on a straight way to the constant pressure boundary condition. After 100 s of simulation (first row of pictures) a pressure of 200 kPa is required to inject the gas at the desired rate. After approximately 150 s the gas has reached the outlet of the system. Thus, a sudden venting of the system takes place. The pressure at the injection point drops to 180 kPa. In the subsequent simulation (1000 s) the gas starts to ascend due to its low density and finally pools up on the boundaries of the fractures. This process must be regarded very critically, as the real extent of the fractures is often not known in real world applications. Now the gas follows a completely different flow path, but still reaches the outlet. In a real world situation the gas would probably have escaped into the far field.

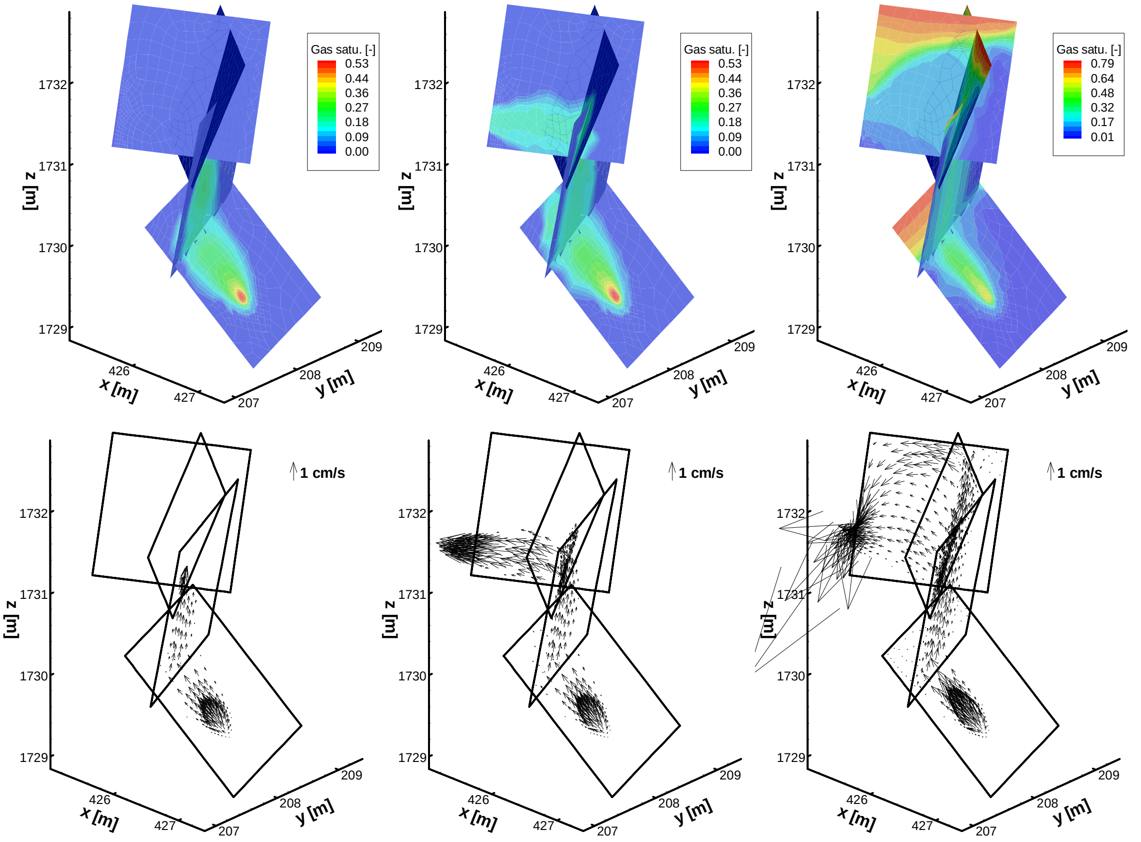


Fig. 4.27: Gas saturation (upper pictures) and Darcy velocities of the gas (lower pictures) in fractures during packer test after 100 s, 150 s and 1000 s (left to right).

Furthermore the spreading of a tracer is regarded. The tracer is added to the gas at the injection point for the period between 200 s and 225 s. The diffusivity is set to 10^{-6} m^2/s, the dispersion length is 0.2 m longitudinal and 0.01 m transverse. The resulting spreading of the tracer plume is presented in Figure 4.28. It should be pointed out that the boundary condition for the injection point is not switched to zero concentration after the tracer is added, but instead the correct dilution is computed.

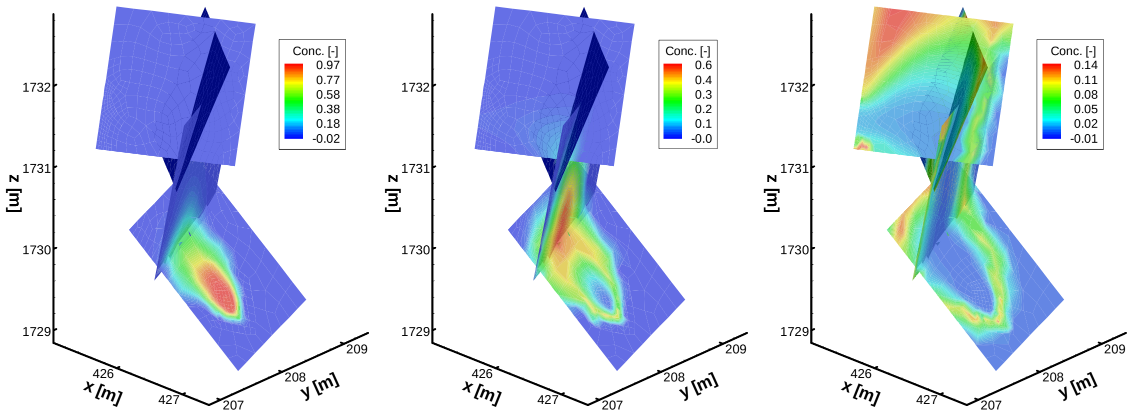


Fig. 4.28: Tracer concentration in the gas phase after 210 s, 250 s and 500 s.

Additionally to the distribution of the tracer in the network, the breakthrough on the outlet is a valuable information. In the current implementation of the simulator it is not possible to balance fluxes over geometrical units. Only single nodes can be observed. Thus, six nodes on the outlet are chosen and the weighted medium of the concentrations is regarded here. The tracer breakthrough is presented in Figure 4.29. Interestingly the breakthrough shows a double-peak and a very long tail. The double-peak can be related to two main flow paths of gas that establish in the system. The first breakthrough is caused by tracer that is following the straight line of gas which is connecting injection point and outlet. This can clearly be seen in the results for a single-node in the middle of the outlet. A sharp, fast arriving peak with a little tailing (dashed line in Figure 4.29) is observed here. In contrast, the result for a node on the upper bound of the outlet (dotted line in Figure 4.29) shows significant tailing. This is due to the storage of gas and tracer in the upper parts of the fracture network.

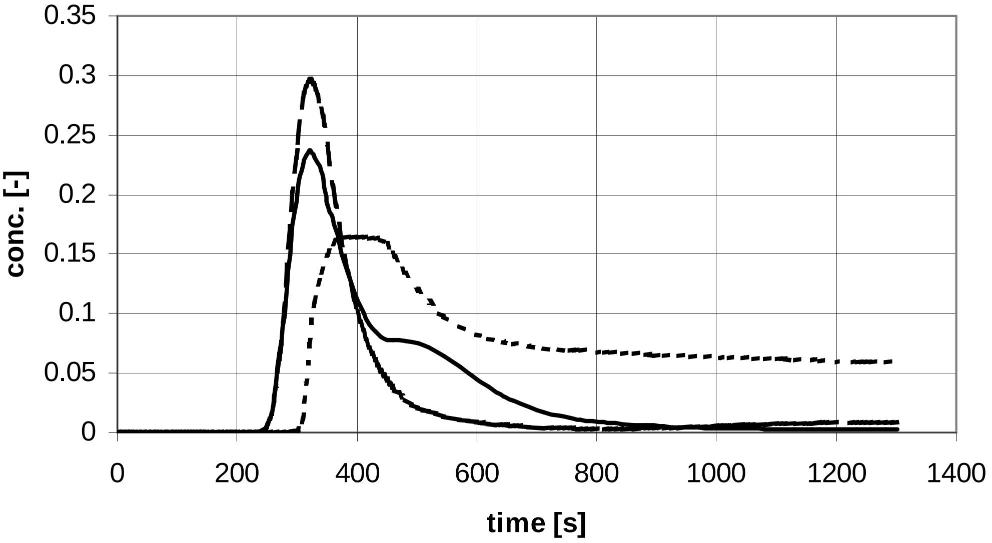


Fig. 4.29: Tracer breakthrough on the outlet. The solid line denotes the weighted medium of six nodal results. The dashed line denotes a single node in the middle of the outlet and the dotted line a single node on the upper bound of the outlet.

It should be pointed out that this simulation imposed severe problems on the stability of the numerical model. Due to the extreme variability of the physical parameters between the regarded fluid phases and because of the very transient flow field, convergence was hardly reached. Furthermore, it must be critically studied if the REV concept, which is underlying this type of model, is valid for gas injections in small scale fracture systems (compare *Persoff & Pruess*, 1995). Possibly a REV approach is not valid, because single pathways of large apertures in the fractures will first be entered by the gas and can then govern the flow field. Thus, models based on percolation theory (*Berkowitz and Balberg*, 1993) might be a better tool for gas migration simulations, especially if these are supposed to be used in risk analysis.

4.8 Three-phase flow in a three-dimensional domain-

The infiltration of non-aqueous phase liquids (NAPL) into natural soil systems results in a complex three-phase flow system. Here, an example for the simulation of a full three-phase flow problem is shown. The system is presented in Figure 4.30, it is a soil cube of 15 m length. Fixed pressure (175000 Pa) and saturation (water saturation equal to unity) boundary conditions are prescribed at the bottom of the system. Furthermore, a pressure and saturation boundary condition of small size is applied in one corner on the top of the system. It serves as a venting hole, so that the air can be driven out by the liquids that enter the system.

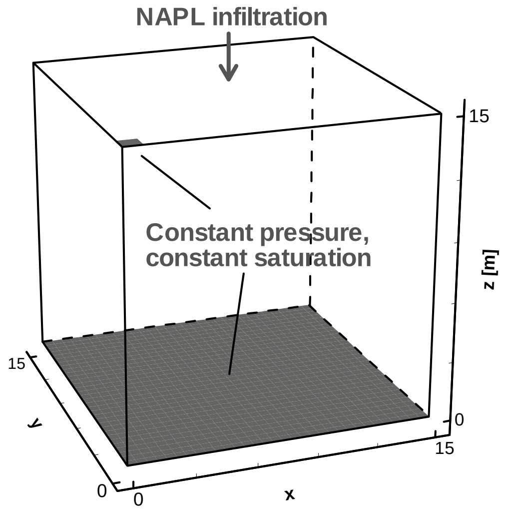


Fig. 4.30: System set-up for the three-phase flow test case. At the bottom both pressure and saturations are prescribed, at the top a constant rate NAPL infiltration takes place.

In the current implementation of the code a limited set of coupling functions for relative permeabilities and capillary pressures in three-phase systems is realised (Chapters 2.2.3 and 2.2.4). Due to the interface structure, these can later easily be extended to more sophisticated functions. Figure 4.31 presents the chosen functions for capillary pressures and relative permeabilities for this example. In Table 4.14 the material parameters for the test case are listed.

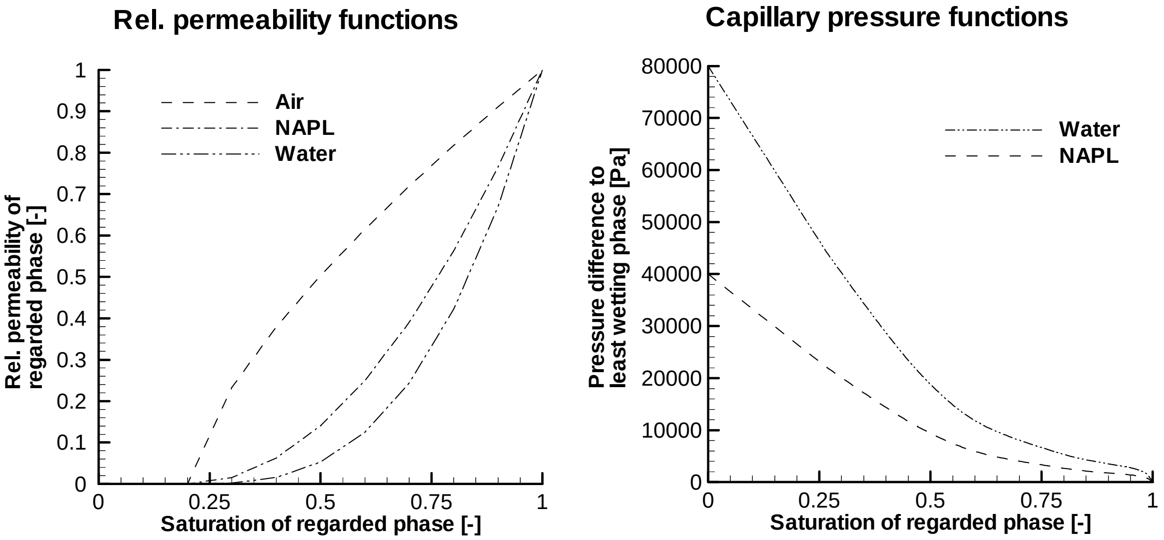


Fig. 4.31: Chosen relative permeability and capillary pressure functions for three mobile fluid phases.

An initial run with no NAPL infiltration the system is performed until steady state is reached, so that good initial conditions for a second run with the NAPL infiltration are available. Afterwards the simulation is restarted in a second run with the infiltration taking place for 10000 s. The system is discretized with an equidistant grid of 0.47 m spacing. The time stepping is adjusted dynamically, starting with a time step of 1 s. The matrix recalculation is triggered for relative changes of the mobility or of the saturation of 10^{-4}.

Table 4.14: Overview of numerical and material parameters for the three-phase flow example.

Fluid density air (pressure dependent)	$\rho_{air} = 1.225 \cdot 10^{-5}$ kg/(m^3 Pa)
Fluid density NAPL	$\rho_{NAPL} = 900$ kg/m^3
Fluid density water	$\rho_{water} = 1000$ kg/m^3
Fluid viscosity air	$\mu_{air} = 1.79 \cdot 10^{-5}$ kg/(ms)]
Fluid viscosity NAPL	$\mu_{NAPL} = 2 \cdot 10^{-3}$ kg/(ms)]
Fluid viscosity water	$\mu_{water} = 1 \cdot 10^{-3}$ kg/(ms)]
Porosity	n = 0.1
Permeability	$k = 10^{-12}$ m^2

Table 4.15: Boundary conditions for the three-phase flow example.

Pressure at the bottom	$p_{water} = 75000$ Pa
Pressure at the top for $x < 1$ m and $y < 1$ m	$p_{water} = 0$ Pa
Saturations at the bottom	$S_{water} = 0.95$, $S_{NAPL} = 0$, $S_{air} = 0.05$
Saturations at the top for $x < 1$ m and $y < 1$ m	$S_{water} = 0.05$, $S_{NAPL} = 0$, $S_{air} = 0.95$
Infiltration of NAPL	10^{-3} m^3/s for 10000 s

The results for the second run are presented in Figure 4.32. The system is cut into two parts in order to have a free sight on the interiors. The infiltration of the NAPL is shown in the pictures of the left column. It is interesting to see, that the NAPL forms the characteristic lens at the end of the infiltration. Furthermore, the NAPL plume leaves a trace of NAPL in the upper parts of the system. This trace is determined by the residual saturation of the NAPL. The water saturation is nearly uninfluenced in the beginning of the simulation. This is due to the simple coupling functions that are used in this test case. If capillary pressure and relative permeability functions are used, which regard the influence of all phases on the coupling functions, this behaviour will change. Only at the end of the regarded time span, when the NAPL reaches the water surface, the influence of the NAPL is visible. In this test case the model adaptive methods reduced the computation time by approximately 50% compared to the standard approach. The differences in the results are very small, they are in the range of the error margin for the matrix recalculations.

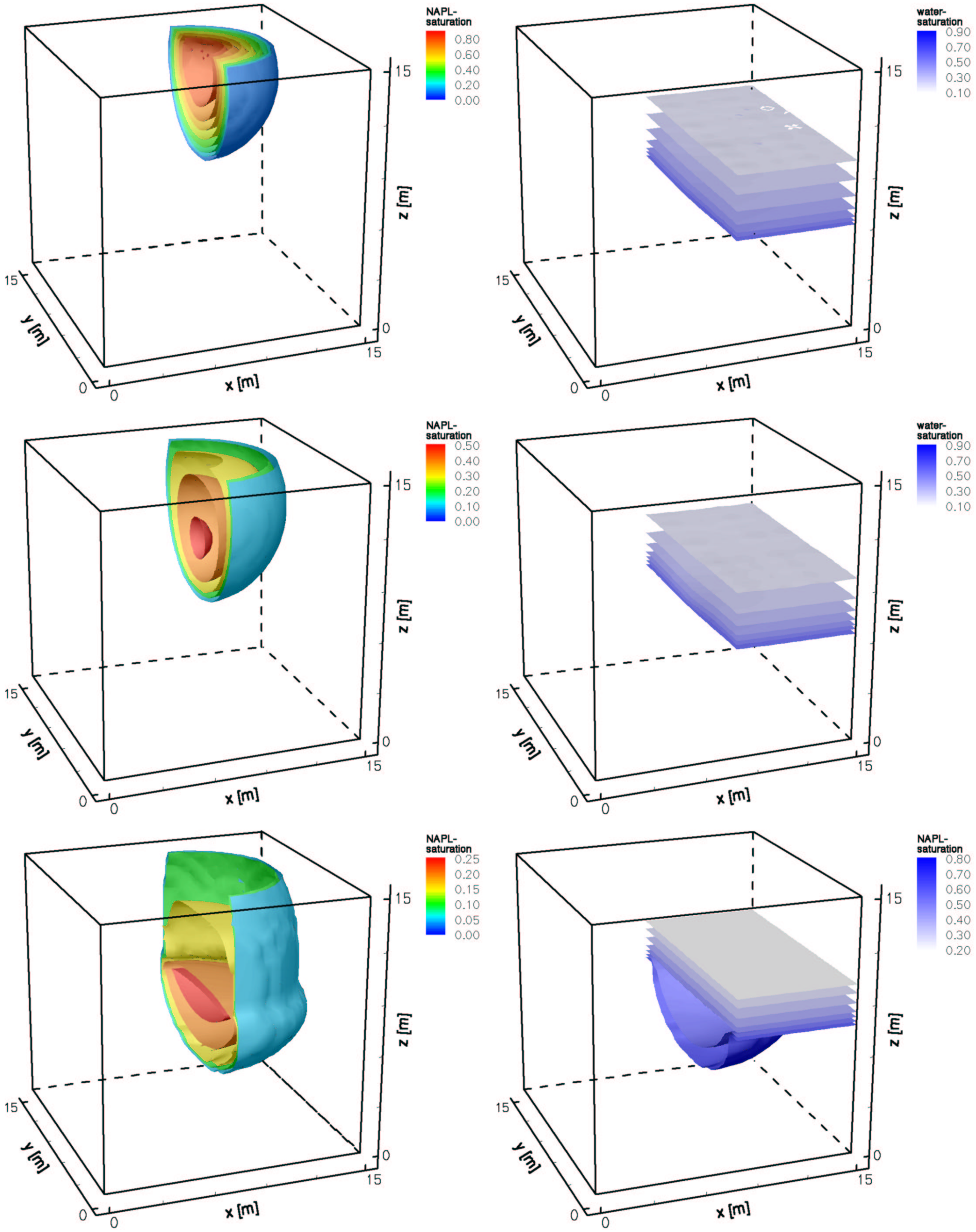


Fig. 4.32: Movement of a NAPL plume in a three-phase flow system. The pictures on the left show the NAPL saturation and the pictures on the right the water saturation. Regarded are the times 10^4 s, 10^5 s and $5 \cdot 10^6$ s (upper to lower).

5 Conclusions and outlook

5.1 Results of this work

The methods developed in this work have shown their suitability for a wide range of flow processes in the subsurface. The developed numerical model is suitable for the simulation of

- saturated single-phase flow,
- unsaturated single-phase flow,
- density driven flow and
- multiphase flow

in the subsurface. Additionally, a single-phase, single-component tracer transport model was extended to the scope of multiphase, multi-component tracer transport and was coupled to the flow model to take into account for transport processes. Furthermore, the tracer transport model was enhanced by a modified method of characteristics scheme in order to enhance the applicability for fracture networks. It was shown that the purely advective method of characteristics scheme is suitable to describe one of the main effects of tracer transport in fracture networks, the tailing effect.

Due to the multitude of regarded physical processes within the flow model, a scalable approach was derived, which can treat flow processes of different complexity by up- and downsizing of a superset model. The downsizing methods were automatized and are suitable to dynamically adapt the underlying physical model. The pros and cons of this method were presented in the regarded application examples. It became obvious that substantial improvements in computation time can be reached and that the introduced errors can be quantified. A possible drawback can be a negative impact on the efficiency of the nonlinear iteration scheme.

The developed system can serve as a construction kit for further developments, as the model is strictly parted in “special” and “common” parts. The common parts can easily be adapted or extended to other physical processes.

For the regarded physical models the set of standard boundary conditions was enhanced by special variants. These variants include special infiltration conditions for unsaturated flow, dynamically switched Dirichlet-type boundary conditions for tracer transport and free outflow (non-reflecting) conditions for multiphase flow.

Due to the strong nonlinear coupling between the regarded primary variables in the model, it was necessary to enhance the iteration and time stepping schemes. The nonlinear iteration was enhanced by a linear predictor and a dynamic relaxation, both can significantly reduce the simulation time. As the regarded processes are highly transient and require the adaptation of the time stepping to the current physical situation, the standard time stepping methods by deterministic criteria were enhanced by different heuristic adaptation schemes.

In order to validate the model, a set of test applications was regarded. Where analytical solutions were available, these were compared to the results of the model in order to quantify the accuracy. For all of the regarded test cases good agreement between analytical and numerical solution was found. As no analytical solutions are available for complex nonlinear coupled processes in arbitrary domains, the results of these test cases can only be checked for plausibility.

A laboratory scale experiment was performed to validate the numerical model for the case of density driven flow in unsaturated media. The experiments also enhance the palette of available benchmarks for testing of other numerical codes. The numerical code developed here was shown to reproduce and predict the flow behaviours observed in the experiments. Significantly, the laboratory experiments were designed according to initial simulations from the numerical model, thus underlining the predictive capacity of the model. Though the predicted behaviour of the system was quite unexpected, the experimental work proved the correctness of the prior numerical simulations. Simulations on the basis of the actual experimental data reproduced the behaviour of the experiments qualitatively and quantitatively.

The numerical simulations of the laboratory scale experiment demonstrate the need for high-resolution input data that define relationships among the different physical parameters. In particular, detailed knowledge of relative permeability - saturation, and capillary pressure - saturation, relationships is crucial to improve the match between numerical simulations and experimental observations in the partially saturated zone. Small-scale structural heterogeneities that are ubiquitous, yet difficult to quantify, of course play a fundamental role in these relationships. And yet, notwithstanding these uncertainties, the numerical model effectively captures the observed flow patterns of the experiment.

5.2 Proposals for further work

Though the simulation of flow and transport in saturated media delivers quite satisfying results, still a large amount of work is to be done in the field of density driven and multiphase flow simulations. This includes the development of both the physical and numerical models.

The applicability of the physical models used in this work on multiphase flow in fractures is doubtable, as it is not yet clear if a REV can be found for those processes. Experiments by *Persoff and Pruess* (1995) and *Thunvik and Braester* (1991) show that finding a REV is at least difficult. If the regarded processes cover only small scales of a fracture, single pathways will govern the system behaviour. Thus, models based on percolation theory may provide a better reproduction of the underlying physical behaviour. Even if a REV is found for the regarded scale, upscaling is still a problem in modelling approaches for single and multiphase flow and transport.

The numerical model offers several possibilities for enhancements; most of them are subject of the multiphase flow kernel:

- In order to enhance the robustness of the multiphase flow kernel, the type of the chosen formulation should dynamically be switched, based on the phase saturations and the material parameters. This will increase the execution speed and the robustness of the code.
- The global pressure function can be weighted between the regarded phases. Together with appropriate formulations for the saturations, this can lead to better convergence (*Ewing*, 1996)
- Phase transitions should be incorporated into the developed model via the sink and source terms. If this step is taken, it should be considered to extend the existing heat transport module (which is already coupled to the flow model described in this work) to the scope of multiphase flow.
- The saturations on material borders should be defined multiple times, according to the number of surrounding materials. Thus, a better representation of the behaviour in heterogeneous systems will be gained.
- The combination of a Quasi-Newton iteration scheme with the current Picard scheme can help to reach convergence faster.
- The parallelization of the code would give access to the world of “super computers”, in order to tackle more complex and larger scale applications.

Obviously, there is still a large amount of work to be done. This is mainly due to the complexity of multiphase flow processes and because of the strong nonlinear coupling of the underlying equations that describe the physical processes. It is important to point out that the multiple nonlinear couplings are not only difficult to treat in the numerical model, but also require an in-depth understanding of the physical processes and of numerical modelling from a modeller who tries to apply this type of numerical model. Otherwise, the modeller will not be able to assemble a coherent set of initial conditions, boundary conditions and appropriate material functions and to use the model appropriately.

6 References

Abbott, M. B., *An introduction to the method of characteristics*, American Elsevier, New York, 1966

Barlag, C., *Adaptive Methoden zur Modellierung von Stofftransport im Kluftgestein*, Dissertation, Institut für Strömungsmechanik und Elektronisches Rechnen im Bauwesen der Universität Hannover, 1997.

Barrett, R., M. Berry, T. Chan, J. Demmel, J. Donato, J. Dongarra, V. Eijkhout, R. Pozo, C. Romine and H.v.d. Vorst, *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods 2nd Edition*, SIAM, Philadelphia, 1994

Bear, J., *Dynamics of Fluids in Porous Media*, Elsevier, New York, 1972

Bear, J., *Hydraulics of Groundwater*, McGraw-Hill, New York, 1979

Bear, J. and A. Verruijt, *Modeling Groundwater Flow and Pollution*, Kluwer Academic Publishers, Dordrecht/Boston/Lancaster/Tokyo, 1987

Bear, J. and Y. Bachmat, *Introduction to Modeling of Transport Phenomena in Porous Media*, Kluwer Academic Publishers, Dordrecht/Boston/London, 1990.

Bear, J., C. F. Tsang and G. de Marsily, *Flow and Contaminant Transport in Fractured Rock*, Academic Press Inc., San Diego/New York/Berkeley/Boston/London/Sydney/Tokyo, 1993

Benard, H., Les tourbillons cellulaires dans nappe liquide transportant de la chaleur par convections en regime permanent, *Rev. Gen. Sci. Pures Appl. Bull. Assoc.*, 11, pp. 1261-1271, 1900

Berkowitz, B., J. Bear and C. Braester, Continuum models for contaminant transport in fractured porous formations, *Water Resour. Res.*, 24, pp. 1225-1236, 1988

Berkowitz, B., and I. Balberg, Percolation theory and its application to groundwater theory, *Water Resour. Res.*, 29, 4, pp. 775-794, 1993

Boufadel, M. C., M. T. Suidan, and A. D. Venosa, Numerical modeling of water-flow below dry salt lakes – Effect of capillarity and viscosity, *J. Hydrol.*, 221(1-2), pp. 55-74, 1999.

Braun, C., *Ein Upscaling-Verfahren für Mehrphasenströmungen in porösen Medien*, Ph.D. thesis, Institut für Wasserbau, Universität Stuttgart, ISBN 3-933761-06-9, 2000

Brooks, A. N. & T. J. R. Hughes, Streamline Upwind Petrov Galerkin Formulations for Convection Dominated Flows with Particular Emphasis on the Incompressible Navier-Stokes Equations, *Computer Methods in Applied Mechanics and Engineering*, 32, pp. 199-259, 1982

Brooks, R. N. and A. T. Corey, Hydraulic properties of porous media, *Hydrology Papers*, Colorado State University, Fort Collins, Colorado, 1964

Brooks, R. N. and A. T. Corey, Properties of porous media affecting fluid flow, *Journal of Irrigation Drainage*, American Society of Civil Engineers, 92 (IR2), pp. 61-88., 1966

Broyden, C. G., Quasi-Newton methods and their application to function minimization, *Math. of Computation*, Vol. 21, No. 99, pp. 368-381, 1967

Buckley, S. E. and M. C. Leverett, Mechanism of Fluid Displacements in Sands, *Transactions of the American Institute of Mining and Metallurgical Engineers (TAIME)*, Vol. 146, pp. 107-116, 1941

Busch, K.-F., L. Luckner and K. Tiemer, *Lehrbuch der Geohydraulik, Band 3, Geohydraulik*, Gebrüder Bornträger, Berlin, Stuttgart 1993

Celia, M. A., E. T. Bouloutas and R L. Zarba, A general mass-conservative numerical solution for the unsaturated flow equation, *Water Resour. Res.*, Vol. 26, No 7, pp. 1483-1496, 1990

Christie, I., D. F. Griffiths, A. R. Mitchell and J. M. Sanz-Serna, Product Approximation for Non-linear Problems in the Finite-Element Method, *IMA J. Numer. Anal.*, 1, pp 253-266, 1981

Courant, R. , K. Friedrichs and H. Lewy, On the partial difference equations of mathematical physics, *IBM-J. Res. Develop.*, 11, pp 215-234, 1967

Cuthill, E. H. and J. McKee, Reducing the bandwidth of sparse symmetric matrices, in *Proc. 24th Nat. Conf. Assoc. Comp. Mech*, ACM Publ., pp. 157-172, 1969

Diersch, H.-J. G., Finite element analysis of dispersion affected saltwater upcoming below a pumping well, *Appl. Math. Modelling*, 8, 5, pp. 305-312, 1984

Diersch, H.-J. G. and P. Perrochet, On the primary variable switching technique for simulating unsaturated-saturated flows, *Adv. in Water Res.*, 23, 1, 25-55, 1999

Duijn, C. J., J. Molenaar and M. J. de Neef, The Effect of Capillary Forces on Immiscible Two-Phase Flow in Heterogeneous Media, *Transport in Porous Media*, 21, pp. 71-93, 1995

DVWK, *Voraussetzungen und Einschränkungen bei der Modellierung der Grundwasserströmung*, Deutscher Verband für Wasserwirtschaft und Kulturbau, Heft 206, Bonn, 1985

Elder, J. W., Transient convection in a porous medium, *J. Fluid Mech.*, 27(3), 609-623, 1967.

Ewing, R., *Three-Phase Flow Formulations*, Technical Note, Institute for Scientific Computation, Texas A&M University, 1996

Forchheimer, P., *Hydraulik*, B.G. Teubner, Leipzig, 1914

Forsyth, P. A., Y. S. Wu, and K. Pruess, Robust numerical methods for saturated-unsaturated flow with dry initial conditions in heterogeneous media. *Adv. in Water Res.* 18, 1, pp. 25-38, 1995

Förster, S. and J. Bruck, *Bestimmung des Diffusions- und Permeabilitätsverhaltens von Wasserstoff in Steinsalz und kompaktiertem Salzgrus, Teil I: Permeabilitätsuntersuch-*

ungen, TU Bergakademie Freiberg, 1994

Fourar, M., S. Bories, R Lenormand and P. Persoff, Two-Phase Flow in Smooth and Rough Fractures: Measurement and Correlation by Porous-Medium and Pipe Flow Models, *Water Resour. Res.*, 29, 11, pp. 3699-3708, 1993

Gärtner, S., *Zur diskreten Approximation kontinuumsmechanischer Bilanzgleichungen*, Institut für Strömungsmechanik und Elektronisches Rechnen im Bauwesen der Universität Hannover, 1987

Geier, J., W, Dershowitz, P. Wallmann, and T. Doe, Discrete Fracture Modeling of In Situ Hydrologic and Tracer Experiments in Fractured and Jointed Rock Masses. In*: Proceedings of the Fractured and Jointed Rock Masses, Lake Tahoe, California, June 3-5, 1992,* eds. L.R. Myer, and C.F. Tsang, 511-8. A.A. Balkema, Rotterdam, 1995

Gibbs, N. E., W. G. Poole and P. K. Stockmeyer, An algorithm for reducing the bandwidth and profile of a sparse matrix, *SIAM J. Num. Anal*, 13, pp. 236-250, 1976

Gresho, P. M. and R. L. Sani, *Incompressible Flow and the Finite Element Method, Advection-Diffusion and Isothermal Laminar Flow,* John Wiley and Sons, Chichester UK, 1998

Habbar, A., *Vergleich verschiedener Krylov-Verfahren für allgemeine reguläre und sehr große lineare Gleichungssysteme,* Diploma thesis, Institut für Strömungsmechanik und Elektronisches Rechnen im Bauwesen der Universität Hannover, 1995

Habbar, A., Direct and inverse modelling of reactive transport processes in porous and fractured media, Dissertation, Institut für Strömungsmechanik und Elektronisches Rechnen im Bauwesen der Universität Hannover, 2001 (in preparation)

Hackbusch, W., *Iterative Lösung schwach besetzter Gleichungssysteme*, Teubner, Stuttgart, 1991

Haverkamp, R., M. Vauclin, J. Touma, P.J. Wierenga and G. Vachaud, A Comparison of Numerical Simulation Models One-Dimensional Infiltration, *Journal of the American Soil Science Society*, Vol. 41, pp. 285ff, 1977

Helmig, R., *Theorie und Numerik der Mehrphasenströmungen in geklüftet-porösen Medien*, Ph.D. thesis, Institut für Strömungsmechanik und Elektronisches Rechnen im Bauwesen der Universität Hannover, 34, 1993

Helmig, R. and R. Huber, Comparison of Galerkin-type discretization techniques for two-phase flow in heterogeneous porous media, *Adv. in Water Res.*, Vol. 21, No. 8, pp. 697-711, 1998a

Helmig, R., *Gekoppelte Strömungs- und Transportprozesse im Untergrund – Ein Beitrag zur Hydrosystemmodellierung -*, Habilitationsschrift, Institut für Wasserbau, Universität Stuttgart, 1998b

Henry, H. R., Salt intrusion into coastal aquifers, *Int. Assoc. of Sci. Hydrol.*, Pub., No. 52, IAHS Press, Oxfordshire, UK, 1960

Herbert, A. W., C. P. Jackson and D. A. Lever, Coupled groundwater flow and solute transport with fluid density strongly dependent upon salt concentration, *Water Resour. Res.*, Vol. 24, pp. 1781-1795, 1988

Hinkelmann, R. (1997) *Parallelisierung eines Lagrange-Euler-Verfahrens für Strömungs- und Stofftransportprozesse in Oberflächengewässern (Parallelization of a Lagrangian-Eulerian scheme for free surface flow and transport processes),* Institut für Strömungsmechanik und Elektronisches Rechnen im Bauwesen der Universität Hannover, 51, 1997

Hirsch, C., *Numerical Computation of Internal and External Flows*, John Wiley & Sons, New York, 1988

Holzbecher, E. O., *Modeling Density-Driven Flow in Porous Media*, Springer-Verlag, Berlin, 1998

Huyakorn, P. S. and G. F. Pinder, Solution of Two-Phase Flow through Porous Media, *Adv. in Water Resources*, Vol. 1, No 5, 1978

HYDROCOIN - *The International HYDROCOIN Project, Level 2: Model Validation*, OECD, Paris 1990.

Jüttner, U., *Entwicklung objektorientierter Datenstrukturen und Algorithmen für die Finite-Element-Methode*, Diploma thesis, Institut für Strömungsmechanik und Elektronisches Rechnen im Bauwesen der Universität Hannover, 1999

Kaiser, R., *Gitteradaption für die Finite-Elemente Simulation gekoppelter Prozesse in geklüftet porösen Medien*, Dissertation, Institut für Strömungsmechanik und Elektronisches Rechnen im Bauwesen der Universität Hannover, 2001 (in preparation)

Kolditz, O., *Stoff- und Wärmetransport im Kluftgestein*, Habilitation, Institut für Strömungsmechanik und Elektronisches Rechnen im Bauwesen der Universität Hannover, 47, 1996

Kolditz, O., R. Ratke, H.J. Diersch, and W. Zielke, Coupled groundwater flow and transport: 1. Verification of variable density flow and transport models, *Adv. Water Resources*, 21(1), 27-46, 1998a

Kolditz, O., A. Habbar, R. Kaiser, H.. Kasper, T. Rother, M. Schulze-Ruhfus, C. Thorenz and W. Zielke, Software concept of simulating coupled processes in subsurface hydrosystems, *Proc. Hydroinformatics 98*, pp. 613-618, Copenhagen, 24-26 August, Balkema, Rotterdam-Brookfield, ISBN 90 5410 984 X., 1998b

Kolditz, O., R. Kaiser, D. Habbar, T. Rother, and C. Thorenz, *ROCKFLOW- Theory and Users Manual, Release 3.4,* Groundwater Modeling Group, Institut für Strömungsmechanik und Elektronisches Rechnen im Bauwesen der Universität Hannover, 1999

Kröhn, K. P., *Simulation von Transportvorgängen im klüftigen Gestein mit der Methode der Finiten Elemente*, Institut für Strömungsmechanik und Elektronisches Rechnen im Bauwesen der Universität Hannover, 1991

Kueper, B. H. and D. B. McWhorter, The Behaviour of Dense, Nonaqueous Phase Liquids in Fractured Clay and Rock, *Groundwater*, 29, 5, pp. 716-728, 1991

Lapidus, L. and G. F. Pinder, *Numerical Solution of Partial Differential Equations in Science and Engineering*, John Wiley & Sons, New York, 1999

Lehmann F. and P. Ackerer, Comparison of Iterative Methods for Improved Solutions of the Fluid Flow Equation in Partially Saturated Porous Media, *Transp. Porous Media,* 31(3), pp. 275-292, 1998

Lenormand, R., Zarcone C. and A. Sarr, Mechanism of the displacement of one fluid by another in a network of capillary ducts, *J. Fluid Mechanics,* 189, 165, 1988

Leverett, M. C., Flow of Oil-water Mixtures through Unconsolidated Sands, , *Transactions of the American Institute of Mining and Metallurgical Engineers (TAIME),* Vol . 132, pp. 149-171, 1938

Leverett, M. C., Capillary Behavior in Porous Solids, *Petroleum Technology*, T.P. 1223, August, 1940

Li M.-H., H.-P. Cheng, and G. T. Yeh, Solving a 3D subsurface flow and transport with adaptive multigrid, *J. Hydrologic Eng*., January, 5, pp. 75-81, 2000

Miller, T., G. Christakos, P. Imhoff, J. McBride, J. Pedit and J. Trangenstein, Multiphase flow and transport modelling in heterogeneous porous media: challenges and approaches, *Adv. in Water Resources,* Vol. 21, No. 2, pp. 77-120, 1998.

Murphy, J. R. and N. R. Thomson, Two-Phase Flow in a Variable Aperture Fracture, *Water Resour. Res.*, 29, 10, pp. 3453-3476, 1993

Muskat, M., *Physical Principles of Oil Production*, McGraw-Hill, New York, 1949

Neumann, S. P., Saturated-unsaturated seepage by finite elements, *Proc. J. Hydr. Div.*, ASCE, 99, 12, pp. 2233-2250, 1973

Pawlowski, J., *Veränderliche Stoffgrößen in der Ähnlichkeitstheorie*, Salle und Sauerländer, Frankfurt am Main, 1991

Parker, J. C., R. J. Lenhard and T. Kuppusamy, A Parametric Model for Constitutive Properties Governing Multiphase Flow in Porous Media, *Water Resour. Res.*, 23, 4, pp. 618-624, 1987a

Parker, J. C. and R. J. Lenhard, A Model for Hysteretic Constitutive Relations Governing Multiphase Flow, *Water Resour. Res.*, 23, 12, pp. 2187-2196, 1987b

Persoff, P., K. Pruess, Two-phase flow visualization and relative permeability measurements in natural rough-walled fractures, *Water Resour. Res.,* Vol. 31, No. 5, 1995

Pruess, K. and Y. W. Tsang, On Two-Phase Relative Permeability and Capillary Pressure of Rough-Walled Rock Fractures, *Water Resour. Res*, 26, 9, pp. 1915-1926, 1990

Redeker, A., *Tracer Transport in Multiphase Flow*, Semester work, Institut für Strömungsmechanik und Elektronisches Rechnen im Bauwesen der Universität Hannover, 1999

Reid R. C., J. M. Prausnitz and B. E. Poling, *The Properties of Gases and Liquids*, Mc Graw-Hill, Singapore, 1988

Richards, L. A., Capillary conduction of liquids through porous media, *Physics*, 1, pp. 318-333, 1931.

Rinnert, B., *Hydrodynamische Dispersion in porösen Medien: Einfluß von Dichteunterschieden auf die Vertikalvermischung in horizontaler Strömung*, Mitteilungen des Institutes für Wasserbau, Heft 52, Universität Stuttgart, 1983

Ritchie, D. M., The Development of the C Language, *Second History of Programming Languages conference*, Cambridge Mass., April, 1993

Rother, T., *Improvement of user environment in geometric modeling and grid generation of hydrogeologic data*, Dissertation, Institut für Strömungsmechanik und Elektronisches Rechnen im Bauwesen der Universität Hannover, 2001 (in preparation)

Sahimi, M., *Flow and Transport in Porous Media and Fractured Rock*, VCH-Verlagsgesellschaft, Weinheim, New York, Basel, Cambridge, Tokyo, 1995

Scheidegger, A. E., General Theory of Dispersion in Porous Media, *Journal of Geophysical Research*, Vol. 66, No. 10, 1961

Schincariol, R. A. and F. W. Schwartz, An Experimental Investigation of Variable Density Flow and Mixing in Homogeneous and Heterogeneous Media, *Water Resour. Res.*, 26(10), pp. 2317-2329, 1990

Schulze-Ruhfus, M., *Adaptive Verfeinerung und Vergröberung gekoppelter 1D/2D/3D-Elemente*, Diploma thesis, Institut für Strömungsmechanik und Elektronisches Rechnen im Bauwesen der Universität Hannover, 1996

Shao, H., O. Kolditz and W. Zielke, *ROCKFLOW- Testbeispiele für Gasströmungen*, Report, Institut für Strömungsmechanik und Elektronisches Rechnen im Bauwesen der Universität Hannover, 1995

Simmons, C. T., K. A. Narayan and R. A. Wooding, Subsurface Hydrology - On a test case for density-dependent groundwater flow and solute transport models: The salt lake problem., *Water Resour. Res.*, 35(12), pp. 3607-3620, 1999

Simunek, J., M. Sejna and M. Th. v. Genuchten, *The HYDRUS-2D Software Package for Simulating the Two-Dimensional Movement of Water, Heat, and Multiple Solutes in Variably-Saturated Media*, U.S. Salinity Laboratory, USDA, ARS, Riverside, 1999

Snell, R. W., Three-Phase Relative Permeability in an Unconsolidated Sand, *Journal of the Institute of Petroleum*, Vol. 48, Num. 459, 1962

Soll, W.E. and M.A. Celia, A modified percolation approach to simulating three-fluid capillary pressure-saturation relationships, *Adv. in Water Resources*, 16(2), pp. 107-126, 1993

Spitz, K., *Dispersion in porösen Medien: Einfluss von Inhomogenitäten und Dichteunterschieden*, Mitteilungen des Institutes für Wasserbau, Heft 60, Universität Stuttgart, 1985

Swartz, B. and B. Wendroff, Generalized Finite Difference Schemes, *Math. Comp.*, 23, pp 37-49, 1969

Thorenz, C., *Problemangepasste Netzgenerierung zur numerischen Simulation von Migrationsproblemen im Grundwasser*, Diploma thesis, Institut für Strömungsmechanik und Elektronisches Rechnen im Bauwesen der Universität Hannover, 1995

Thorenz, C., O. Kolditz and W. Zielke, *Numerische Modellierung von Flüssigkeits-Gasströmungen im Festgestein.- Weiterentwicklung der Mehrphasenmodelle ROCKFLOW-MM und MUFTE*, Institut für Strömungsmechanik und Elektronisches Rechnen im Bauwesen der Universität Hannover, 1996

Thorenz, C., O. Kolditz, and W. Zielke, A "method of characteristics" concept for advective tracer transport in fracture networks, *Int. Assoc. of Sci. Hydrol.*, Pub. No. 265, IAHS Press, Oxfordshire UK, 2000

Thunvik, R. and C. Braester, Gas Migration in Discrete Fracture Networks, *Water Resour. Res.*, Vol. 2, 1991

UNESCO, Tenth Report of the Joint Panel on Oceanographic Tables and Standards, *UNESCO Technical Papers in Marine Science*, 36, pp. 24ff, 1981

van Genuchten, M. Th., A closed form equation for predicting the hydraulic conductivity of unsaturated soils, *Soil Sci. Soc. Am. J.*, 44(5), pp. 892-898, 1980

Warren, J. E., and P. J. Root, The Behavior of Naturally Fractured Reservoirs, *Society of Petroleum Engineer*s Journal, pp. 245-255, Sept. 1963

Witherspoon, P. A., J. S. Y. Wang, K. Iwai and J. E. Gale, Validity of cubic law for fluid flow in a deformable rock fracture, *Water Resour. Res.*, Vol. 16, No. 6, pp. 1016-1024, 1980

Wollrath, J., *Ein Strömungs- und Transportmodell für klüftiges Gestein und Untersuchungen zu homogenen Ersatzsystemen*, Dissertation, Institut für Strömungsmechanik und Elektronisches Rechnen im Bauwesen der Universität Hannover, 1990

Zienkiewicz, O. C. and P. N. Godbole, *Finite Elements in Fluids, Vol. 1, Viscous Flow and Hydrodynamics*, John Wiley and Sons, London, England, 1975

A Element matrices for 1D elements

In the following the evaluation of element matrices for a two node element with linear weighting and basis functions will be presented. The indices ^{a} and ^{b} will be used to denote node values of nodes a and b, respectively. To simplify the evaluation, the *Group FEM* is used for the one-dimensional elements where noted. Furthermore expressions in the denominator of an expression are replaced y the corresponding element middle value, because otherwise computationally very expensive numerical treatment of the integrals would be required. The local coordinate of each element is given by l, the length of the element by L.

Matrix type A and B: Storage and mass matrices

The matrices of type A and B are rather similar, thus only the derivation of a more generic matrix type will be presented here.

Streamline upwinded weighting functions will be used for this matrix type. The upwinding factor α must be directed streamline upwards.

$$\varphi_a = 1 - l/L + \frac{\alpha}{2} \qquad \text{(A.10)}$$

$$\varphi_b = l/L - \frac{\alpha}{2}$$

Within the following equation e denotes the integrant. It is approximated by the element middle value. The vector of unknown node values is given by **u**.

Generic storage or mass matrix : (A.11)

$$\int_V \boldsymbol{\varphi}\overline{e}\,\overline{\boldsymbol{\omega}}\,\overline{\mathbf{u}}\,dV \cong Ae\int_L \begin{bmatrix} \varphi_a\left(u^a\omega_a + u^b\omega_b\right) \\ \varphi_b\left(u^a\omega_a + u^b\omega_b\right) \end{bmatrix} dl =$$

$$Ae\int_L \begin{bmatrix} \left(1-\frac{l}{L}+\frac{\alpha}{2}\right)\left(\left(1-\frac{l}{L}\right)+\frac{l}{L}\right) \\ \left(\frac{l}{L}-\frac{\alpha}{2}\right)\left(\left(1-\frac{l}{L}\right)+\frac{l}{L}\right) \end{bmatrix} dl =$$

$$\frac{Ae}{12L}\int_L \begin{bmatrix} 4+3\alpha & 2+3\alpha \\ 2-3\alpha & 4-3\alpha \end{bmatrix} dl$$

Matrix type C: Flux matrix

The derivation is performed in two parts. In the program code it is performed in a similar way, but the matrices are added upon each other before storing.

First part of flux matrix C

$$\int_{V_E} \mathrm{grad}(\boldsymbol{\varphi}) \left(\frac{k_{r_\alpha} k}{\mu_\alpha} \mathrm{grad}(p_\alpha) \right) dV_E \cong \tag{A.12}$$

$$kA \begin{bmatrix} \int_{L_E} \mathrm{grad}(\varphi^a) \left(\left(\omega^a \left(\frac{k^a_{r_\alpha}}{\mu^a_\alpha} \right) + \omega^b \left(\frac{k^b_{r_\alpha}}{\mu^b_\alpha} \right) \right) \mathrm{grad}\left(\left(\omega^a p^a_\alpha \right) + \left(\omega^b p^b_\alpha \right) \right) \right) dl \\ \int_{L_E} \mathrm{grad}(\varphi^b) \left(\left(\omega^a \left(\frac{k^a_{r_\alpha}}{\mu^a_\alpha} \right) + \omega^b \left(\frac{k^b_{r_\alpha}}{\mu^b_\alpha} \right) \right) \mathrm{grad}\left(\left(\omega^a p^a_\alpha \right) + \left(\omega^b p^b_\alpha \right) \right) \right) dl \end{bmatrix} =$$

$$kA \begin{bmatrix} \int_{L_E} \mathrm{grad}(1 - l/L) \left(\left((1 - l/L) \left(\frac{k^a_{r_\alpha}}{\mu^a_\alpha} \right) + (l/L) \left(\frac{k^b_{r_\alpha}}{\mu^b_\alpha} \right) \right) \mathrm{grad}\left(\left((1 - l/L) p^a_\alpha \right) + \left((l/L) p^b_\alpha \right) \right) \right) dl \\ \int_{L_E} \mathrm{grad}(l/L) \left(\left((1 - l/L) \left(\frac{k^a_{r_\alpha}}{\mu^a_\alpha} \right) + (l/L) \left(\frac{k^b_{r_\alpha}}{\mu^b_\alpha} \right) \right) \mathrm{grad}\left(\left((1 - l/L) p^a_\alpha \right) + \left((l/L) p^b_\alpha \right) \right) \right) dl \end{bmatrix} =$$

$$kA \begin{bmatrix} -\frac{1}{L^2} \int_{L_E} \left((1 - l/L) \left(\frac{k^a_{r_\alpha}}{\mu^a_\alpha} \right) + (l/L) \left(\frac{k^b_{r_\alpha}}{\mu^b_\alpha} \right) \right) dl \left(-p^a_\alpha + p^b_\alpha \right) \\ \frac{1}{L^2} \int_{L_E} \left((1 - l/L) \left(\frac{k^a_{r_\alpha}}{\mu^a_\alpha} \right) + (l/L) \left(\frac{k^b_{r_\alpha}}{\mu^b_\alpha} \right) \right) dl \left(-p^a_\alpha + p^b_\alpha \right) \end{bmatrix} =$$

$$\frac{kA}{2L} \left(\frac{k^a_{r_\alpha}}{\mu^a_\alpha} + \frac{k^b_{r_\alpha}}{\mu^b_\alpha} \right) \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} p^a_\alpha \\ p^b_\alpha \end{bmatrix}$$

Second part of flux matrix C

One possibility is the evaluation with the *Group FEM:*

$$-\int_{V_E} \boldsymbol{\varphi} \left(\mathrm{grad}(\rho_\alpha) \frac{k_{r_\alpha} k}{\rho_\alpha \mu_\alpha} \mathrm{grad}(\overline{\boldsymbol{\omega}}) \right) dV_E \cong \tag{A.13}$$

$$-k \frac{\rho^b_\alpha - \rho^a_\alpha}{L} \int_{V_E} \{\varphi_1, .., \varphi_n\} \left((1 - l/L) \left(\frac{k^a_{r_\alpha}}{\rho^a_\alpha \mu^a_\alpha} \right) + (l/L) \left(\frac{k^b_{r_\alpha}}{\rho^b_\alpha \mu^b_\alpha} \right) \right) dV_E \cdot \left(\frac{p^b_\alpha - p^a_\alpha}{L} \right) =$$

$$-kA\frac{\rho_\alpha^b-\rho_\alpha^a}{L}\begin{bmatrix}\int\limits_L\left(1-\frac{l}{L}\right)\left(\left(1-\frac{l}{L}\right)\frac{k_{r_\alpha}^a}{\rho_\alpha^a\mu_\alpha^a}+\frac{l}{L}\frac{k_{r_\alpha}^b}{\rho_\alpha^b\mu_\alpha^b}\right)dl\\ \int\limits_L\left(\frac{l}{L}\right)\left(\left(1-\frac{l}{L}\right)\frac{k_{r_\alpha}^a}{\rho_\alpha^a\mu_\alpha^a}+\frac{l}{L}\frac{k_{r_\alpha}^b}{\rho_\alpha^b\mu_\alpha^b}\right)dl\end{bmatrix}\cdot\left(\frac{p_\alpha^b-p_\alpha^a}{L}\right)=$$

$$-kA\frac{\rho_\alpha^b-\rho_\alpha^a}{L}\begin{bmatrix}\int\limits_L\left(\left(1-2\frac{l}{L}+\frac{l^2}{L^2}\right)\frac{k_{r_\alpha}^a}{\rho_\alpha^a\mu_\alpha^a}+\left(\frac{l}{L}-\frac{l^2}{L^2}\right)\frac{k_{r_\alpha}^b}{\rho_\alpha^b\mu_\alpha^b}\right)dl\\ \int\limits_L\left(\left(\frac{l}{L}-\frac{l^2}{L^2}\right)\frac{k_{r_\alpha}^a}{\rho_\alpha^a\mu_\alpha^a}+\frac{l^2}{L^2}\frac{k_{r_\alpha}^b}{\rho_\alpha^b\mu_\alpha^b}\right)dl\end{bmatrix}\cdot\left(\frac{p_\alpha^b-p_\alpha^a}{L}\right)=$$

$$-kA\frac{\rho_\alpha^b-\rho_\alpha^a}{L}\begin{bmatrix}\left(\left(\frac{1}{3}L\right)\frac{k_{r_\alpha}^a}{\rho_\alpha^a\mu_\alpha^a}+\left(\frac{1}{2}L-\frac{1}{3}L\right)\frac{k_{r_\alpha}^b}{\rho_\alpha^b\mu_\alpha^b}\right)\\ \left(\left(\frac{1}{2}L-\frac{1}{3}L\right)\frac{k_{r_\alpha}^a}{\rho_\alpha^a\mu_\alpha^a}+\left(\frac{1}{3}L\right)\frac{k_{r_\alpha}^b}{\rho_\alpha^b\mu_\alpha^b}\right)\end{bmatrix}\cdot\left(\frac{p_\alpha^b-p_\alpha^a}{L}\right)=$$

$$\frac{kA}{6L}\left(\rho_\alpha^b-\rho_\alpha^a\right)\begin{bmatrix}\left(2\frac{k_{r_\alpha}^a}{\rho_\alpha^a\mu_\alpha^a}+\frac{k_{r_\alpha}^b}{\rho_\alpha^b\mu_\alpha^b}\right) & -\left(2\frac{k_{r_\alpha}^a}{\rho_\alpha^a\mu_\alpha^a}+\frac{k_{r_\alpha}^b}{\rho_\alpha^b\mu_\alpha^b}\right)\\ \left(\frac{k_{r_\alpha}^a}{\rho_\alpha^a\mu_\alpha^a}+2\frac{k_{r_\alpha}^b}{\rho_\alpha^b\mu_\alpha^b}\right) & -\left(\frac{k_{r_\alpha}^a}{\rho_\alpha^a\mu_\alpha^a}+2\frac{k_{r_\alpha}^b}{\rho_\alpha^b\mu_\alpha^b}\right)\end{bmatrix}\cdot\begin{bmatrix}p_\alpha^a\\ p_\alpha^b\end{bmatrix}$$

The equation presented above describes the resulting flux matrix if the Group FEM approach is used. Another possibility derivation for the matrix uses element middle values for the denominator:

$$-\int\limits_{V_E}\varphi\left(\operatorname{grad}(\rho_\alpha)\frac{k_{r_\alpha}k}{\rho_\alpha\mu_\alpha}\operatorname{grad}(\overline{\varpi})\right)dV_E\cong \tag{A.14}$$

$$-k\frac{\rho_\alpha^b-\rho_\alpha^a}{L\rho_\alpha\mu_\alpha}\int\limits_{V_E}\{\varphi_1,..,\varphi_n\}\left((1-l/L)k_{r_\alpha}^a+(l/L)k_{r_\alpha}^b\right)dV_E\cdot\left(\frac{p_\alpha^b-p_\alpha^a}{L}\right)=$$

$$-kA\frac{\rho_\alpha^b-\rho_\alpha^a}{L\rho_\alpha\mu_\alpha}\begin{bmatrix}\int\limits_L\left(1-\frac{l}{L}\right)\left(\left(1-\frac{l}{L}\right)k_{r_\alpha}^a+\frac{l}{L}k_{r_\alpha}^b\right)dl\\ \int\limits_L\left(\frac{l}{L}\right)\left(\left(1-\frac{l}{L}\right)k_{r_\alpha}^a+\frac{l}{L}k_{r_\alpha}^b\right)dl\end{bmatrix}\cdot\left(\frac{p_\alpha^b-p_\alpha^a}{L}\right)=$$

$$-kA\frac{\rho_\alpha^b-\rho_\alpha^a}{L\rho_\alpha\mu_\alpha}\begin{bmatrix}\int\limits_L\left(\left(1-2\frac{l}{L}+\frac{l^2}{L^2}\right)k_{r_\alpha}^a+\left(\frac{l}{L}-\frac{l^2}{L^2}\right)k_{r_\alpha}^b\right)dl\\ \int\limits_L\left(\left(\frac{l}{L}-\frac{l^2}{L^2}\right)k_{r_\alpha}^a+\frac{l^2}{L^2}k_{r_\alpha}^b\right)dl\end{bmatrix}\cdot\left(\frac{p_\alpha^b-p_\alpha^a}{L}\right)=$$

$$\frac{kA}{6L}\frac{\rho_\alpha^b-\rho_\alpha^a}{\rho_\alpha\mu_\alpha}\begin{bmatrix}\left(2k_{r_\alpha}^a+k_{r_\alpha}^b\right) & -\left(2k_{r_\alpha}^a+k_{r_\alpha}^b\right)\\ \left(k_{r_\alpha}^a+2k_{r_\alpha}^b\right) & -\left(k_{r_\alpha}^a+2k_{r_\alpha}^b\right)\end{bmatrix}\cdot\begin{bmatrix}p_\alpha^a\\ p_\alpha^b\end{bmatrix}$$

Matrix type D: Gravitation matrix

The gravitation matrix is very similar to the flux matrix, the treatment is equivalent. The first part of matrix type **D** is:

$$\int\limits_{V_E}\text{grad}(\boldsymbol{\varphi})\left(\frac{k_{r_\alpha}k}{\mu_\alpha}(\rho_\alpha\mathbf{g})\right)dV_E=\int\limits_{V_E}\text{grad}(\boldsymbol{\varphi})\left(\frac{k_{r_\alpha}k}{\mu_\alpha}\left(\rho_\alpha g\frac{\partial z}{\partial l}\right)\right)dV_E\cong \tag{A.15}$$

$$kA\begin{bmatrix}\int\limits_{L_E}\text{grad}(\varphi^a)\left(\left(\omega^a\left(\frac{k_{r_\alpha}^a\rho_\alpha^a}{\mu_\alpha^a}\right)+\omega^b\left(\frac{k_{r_\alpha}^b\rho_\alpha^b}{\mu_\alpha^b}\right)\right)\left(\rho_\alpha g\frac{\partial z}{\partial l}\right)\right)dl\\ \int\limits_{L_E}\text{grad}(\varphi^b)\left(\left(\omega^a\left(\frac{k_{r_\alpha}^a\rho_\alpha^a}{\mu_\alpha^a}\right)+\omega^b\left(\frac{k_{r_\alpha}^b\rho_\alpha^b}{\mu_\alpha^b}\right)\right)\left(\rho_\alpha g\frac{\partial z}{\partial l}\right)\right)dl\end{bmatrix}=$$

$$kAg\begin{bmatrix}\int\limits_{L_E}\text{grad}(1-l/L)\left(\left((1-l/L)\left(\frac{k_{r_\alpha}^a\rho_\alpha^a}{\mu_\alpha^a}\right)+(l/L)\left(\frac{k_{r_\alpha}^b\rho_\alpha^b}{\mu_\alpha^b}\right)\right)\left(\frac{z_b-z_a}{L}\right)\right)dl\\ \int\limits_{L_E}\text{grad}(l/L)\left(\left((1-l/L)\left(\frac{k_{r_\alpha}^a\rho_\alpha^a}{\mu_\alpha^a}\right)+(l/L)\left(\frac{k_{r_\alpha}^b\rho_\alpha^b}{\mu_\alpha^b}\right)\right)\left(\frac{z_b-z_a}{L}\right)\right)dl\end{bmatrix}=$$

$$kA\begin{bmatrix}-\frac{1}{L}\int\limits_{L_E}\left(\left((1-l/L)\left(\frac{k_{r_\alpha}^a\rho_\alpha^a}{\mu_\alpha^a}\right)+(l/L)\left(\frac{k_{r_\alpha}^b\rho_\alpha^b}{\mu_\alpha^b}\right)\right)\left(\frac{z_b-z_a}{L}\right)\right)dl\\ \frac{1}{L}\int\limits_{L_E}\left(\left((1-l/L)\left(\frac{k_{r_\alpha}^a\rho_\alpha^a}{\mu_\alpha^a}\right)+(l/L)\left(\frac{k_{r_\alpha}^b\rho_\alpha^b}{\mu_\alpha^b}\right)\right)\left(\frac{z_b-z_a}{L}\right)\right)dl\end{bmatrix}=$$

$$\frac{kA}{2L}\left(\frac{k_{r_\alpha}^a\rho_\alpha^a}{\mu_\alpha^a}+\frac{k_{r_\alpha}^b\rho_\alpha^b}{\mu_\alpha^b}\right)(z_b-z_a)\begin{bmatrix}1\\ -1\end{bmatrix}$$

Second part of gravitation matrix D

$$-\int_{V_E}\boldsymbol{\varphi}\left(\mathrm{grad}(\rho_\alpha)\frac{k_{r_\alpha}k}{\rho_\alpha\mu_\alpha}\rho_\alpha\mathbf{g}\right)dV_E=-\int_{V_E}\boldsymbol{\varphi}\left(\mathrm{grad}(\rho_\alpha)\frac{k_{r_\alpha}k}{\rho_\alpha\mu_\alpha}\rho_\alpha g\frac{\partial z}{\partial l}\right)dV_E\cong \tag{A.16}$$

$$-k\frac{\rho_\alpha^b-\rho_\alpha^a}{L}\int_{V_E}\{\varphi_1,..,\varphi_n\}\left((1-l/L)\left(\frac{k_{r_\alpha}^a}{\mu_\alpha^a}\right)+(l/L)\left(\frac{k_{r_\alpha}^b}{\mu_\alpha^b}\right)\right)dV_E\cdot\left(\frac{z_b-z_a}{L}\right)=$$

$$-kA\frac{\rho_\alpha^b-\rho_\alpha^a}{L}\begin{bmatrix}\displaystyle\int_L\left(1-\frac{l}{L}\right)\left(\left(1-\frac{l}{L}\right)\frac{k_{r_\alpha}^a}{\mu_\alpha^a}+\frac{l}{L}\frac{k_{r_\alpha}^b}{\mu_\alpha^b}\right)dl\\ \displaystyle\int_L\left(\frac{l}{L}\right)\left(\left(1-\frac{l}{L}\right)\frac{k_{r_\alpha}^a}{\mu_\alpha^a}+\frac{l}{L}\frac{k_{r_\alpha}^b}{\mu_\alpha^b}\right)dl\end{bmatrix}\cdot\left(\frac{z_b-z_a}{L}\right)=$$

$$-kA\frac{\rho_\alpha^b-\rho_\alpha^a}{L}\begin{bmatrix}\left(\left(\frac{1}{3}L\right)\frac{k_{r_\alpha}^a}{\mu_\alpha^a}+\left(\frac{1}{6}L\right)\frac{k_{r_\alpha}^b}{\mu_\alpha^b}\right)\\ \left(\left(\frac{1}{6}L\right)\frac{k_{r_\alpha}^a}{\mu_\alpha^a}+\left(\frac{1}{3}L\right)\frac{k_{r_\alpha}^b}{\mu_\alpha^b}\right)\end{bmatrix}\cdot\left(\frac{z_b-z_a}{L}\right)=$$

$$\frac{kA}{6L}\left(\rho_\alpha^b-\rho_\alpha^a\right)\left(z_a-z_b\right)\begin{bmatrix}\left(2\frac{k_{r_\alpha}^a}{\mu_\alpha^a}+1\frac{k_{r_\alpha}^b}{\mu_\alpha^b}\right)\\ \left(1\frac{k_{r_\alpha}^a}{\mu_\alpha^a}+2\frac{k_{r_\alpha}^b}{\mu_\alpha^b}\right)\end{bmatrix}$$

Liste der bereits erschienenen Institutsberichte

01/1970 **Holz, K.-P.** Ergänzung des Verfahrens finiter Elemente durch Ecksingularitäten zur verbesserten Berechnung schiefwinkliger Platten. Dissertation, Techn. Univ. Hannover, 1970

02/1971 **Ehlers, K.-D.** Berechnung instationärer Grund- und Sickerwasserströmungen mit freier Oberfläche nach der Methode finiter Elemente. Dissertation, Techn. Univ. Hannover, 1971

03/1971 **Meissner, U.** Berechnung von Schalen unter großen Verschiebungen und Verdrehungen bei kleinen Verzerrungen mit Hilfe finiter Dreieckselemente. Dissertation, Techn. Univ. Hannover, 1971

04/1972 **Grotkop, G.** Die Berechnung von Flachwasserwellen nach der Methode der finiten Elemente. Dissertation, Techn. Univ. Hannover, Sonderdruck aus dem Jahresbericht 1971 d. SFB 79, H. 2, 1972

05/1973 **Schulze, K.-W.** Eine problemorientierte Sprache für die Dynamik offener Gerinne. Dissertation, Techn. Univ. Hannover, Mitteil. d. SFB 79, Heft 1, 1973

06/1977 **Beyer, A.** Die Berechnung großräumiger Grundwasserströmungen mit Vertikalstruktur mit Hilfe der Finite-Element-Methode. Dissertation, Fortschrittberichte der VDI-Zeitschriften, Reihe 4, Nr. 34, 1977

07/1977 **Ebeling, H.** Berechnung der Vertikalstruktur wind- und gezeitenerzeugter Strömungen nach der Methode der finiten Elemente. Dissertation, Fortschrittberichte der VDI-Zeitschriften, Reihe 4, Nr. 32, 1977

08/1977 **Gärtner, S.** Zur Berechnung von Flachwasserwellen und instationären Transportprozessen mit der Methode der finiten Elemente. Dissertation, Fortschrittberichte der VDI-Zeitschriften, Reihe 4, Nr. 30, 1977

09/1977 **Herrling, B.** Eine hybride Formulierung in Wasserständen zur Berechnung von Flachwasserwellen mit der Methode finiter Elemente. Dissertation, Fortschrittberichte der VDI-Zeitschriften, Reihe 4, Nr. 37, 1977

10/1979 **Hennlich, H.-H.** Aeroelastische Stabilitätsuntersuchung von Linientragwerken. Dissertation, Fortschrittberichte der VDI-Zeitschriften, Reihe 4, Nr. 49, 1979

11/1979 **Kalocay, E.** Zur numerischen Behandlung der Konvektions-Diffusions-Gleichung im Hinblick auf das inverse Problem. Dissertation, Univ. Hannover, 1979

12/1980 **Januszewski, U**. Automatische Eichung für ein- und zweidimensionale, hydrodynamisch-numerische Flachwassermodelle. Dissertation, Univ. Hannover, Fortschrittberichte der VDI-Zeitschriften, Reihe 4, Nr. 58, 1980

13/1982 **Carbonel Huaman, C.A.A.** Numerisches Modell der Zirkulation in Auftriebsgebieten mit Anwendung auf die nordperuanische Küste. Dissertation, Univ. Hannover, 1982

14/1985 **Tuchs, M.** Messungen und Modellierung am Deep Shaft. Dissertation, Univ. Hannover, 1984

15/1985 Theu**nert, F.** Zum lokalen Windstau in ästuarien bei Sturmfluten - Numerische Untersuchungen am Beispiel der Unterelbe. Dissertation, Univ. Hannover, 1984

16/1985 **Perko, H.-D.** Gasausscheidung in instationärer Rohrströmung. Dissertation, Univ. Hannover, 1984

17/1985 **Crotogino, A.** Ein Beitrag zur numerischen Modellierung des Sedimenttransports in Verbindung mit vertikal integrierten Strömungsmodellen. Dissertation, Univ. Hannover, 1984

18/1985 **Rottmann-Söde, W.** Ein halbanalytisches FE-Modell für harmonische Wellen zur Berechnung von Wellenunruhen in Häfen und im Küstenvorfeld. Dissertation, Univ. Hannover, 1985

19/1985 **Nitsche, G.** Explizite Finite-Element-Modelle und ihre Naturanwendungen auf Strömungsprobleme in Tidegebieten. Dissertation, Univ. Hannover, 1985

20/1985 **Vera Muthre, C.** Untersuchungen zur Salzausbreitung in Ästuarien mit Taylorschen Dispersionsmodellen. Dissertation, Univ. Hannover, 1985

21/1985 **Schaper, H.** Ein Beitrag zur numerischen Berechnung von nichtlinearen kurzen Flachwasserwellen mit verbesserten Differenzenverfahren. Dissertation, Univ. Hannover, 1985

22/1986 **Urban, C.** Ein Finite-Element-Verfahren mit linearen Ansätzen für stationäre zweidimensionale Strömungen. Dissertation, Univ. Hannover, 1986

23/1987 **Heyer, H.** Die Beeinflussung der Tidedynamik in ästuarien durch Steuerung - Ein Beitrag zur Anwendung von Optimierungsverfahren in der Wasserwirtschaft. Dissertation, Univ. Hannover, 1987

24/1987 **Gärtner, S.** Zur diskreten Approximation kontinuumsmechanischer Bilanzgleichungen. Institutsbericht, davon 4 Abschnitte als Habilitationsschrift angenommen, Univ. Hannover, 1987

25/1988 **Rogalla, B.U.** Zur statischen und dynamischen Berechnung geometrisch nichtlinearer Linientragwerke unter Strömungs- und Wellenlasten. Dissertation, Univ. Hannover, 1988

26/1990 **Lang, G.** Zur Schwebstoffdynamik von Trübungszonen in Ästuarien. Dissertation, Univ. Hannover, 1990

27/1990 **Stittgen, M.** Zur Fluid-Struktur-Wechselwirkung in flexiblen Offshore-Schlauchleitungen. Dissertation, Univ. Hannover, 1990

28/1990 **Wollrath, J.** Ein Strömungs- und Transportmodell für klüftiges Gestein und Untersuchungen zu homogenen Ersatzsystemen. Dissertation, Univ. Hannover, 1990

29/1991 **Kröhn, K.-P.** Simulation von Transportvorgängen im klüftigen Gestein mit der Methode der Finiten Elemente. Dissertation, Univ. Hannover, 1991

30/1991 **Lehfeldt, R.** Ein algebraisches Turbulenzmodell für Ästuare. Dissertation, Univ. Hannover, 1991

31/1991 **Prüser, H.-H.** Zur mathematischen Modellierung der Interaktion von Seegang und Strömung im flachen Wasser. Dissertation, Univ. Hannover, 1991

dito **Schröter, A.** Das numerische Seegangsmodell BOWAM2 1990 - Grundlagen und Verifikationen - . Univ. Hannover, 1991

32/1992 **Leister, K.** Anwendung numerischer Flachwassermodelle zur Bestimmung von Wasserlinien. Dissertation, Univ. Hannover, 1992

33/1993 **Ramthun, B.** Zur Druckstoßsicherung von Fernwärmenetzen und zur Dynamik von Abnehmeranlagen. Dissertation, Univ. Hannover, 1993

34/1993 **Helmig, R.** Theorie und Numerik der Mehrphasenströmungen in geklüftet-porösen Medien. Dissertation, Univ. Hannover, 1993

35/1994 **Plüß, A.** Netzbearbeitung und Verfahrensverbesserungen für Tidemodelle nach der Finiten Element Methode. Dissertation, Univ. Hannover, 1994

36/1994 **Nöthel, H.** Statistisch-numerische Beschreibung des Wellen- und Strömungsgeschehens in einem Buhnenfeld. Dissertation, Univ. Hannover, 1994

37/1994 **Shao, H.** Simulation von Strömungs- und Transportvorgängen in geklüfteten porösen Medien mit gekoppelten Finite-Element- und Rand-Element-Methoden. Dissertation, Univ. Hannover, 1994

38/1994 **Stengel, T.** Änderungen der Tidedynamik in der Deutschen Bucht und Auswirkungen eines Meeresspiegelanstiegs. Dissertation, Univ. Hannover, 1994

39/1994 **Schubert, R.** Ein Softwaresystem zur parallelen interaktiven Strömungssimulation und -visualisierung. Dissertation, Univ. Hannover, 1994

40/1994 **Alm, W.** Zur Gestaltung eines Informationssystems im Küsteningenieurwesen. Dissertation, Univ. Hannover, 1994

41/1994 **Benali, H.** Zur Kopplung von FEM- und CAD-Programmen im Bauwesen über neutrale Datenschnittstellen. Dissertation, Univ. Hannover, 1994

42/1995 **Schröter, A.** Nichtlineare zeitdiskrete Seegangssimulation im flachen und tieferen Wasser. Dissertation, Univ. Hannover, 1995

43/1995 **Blase, Th.** Ein systemtechnischer Ansatz zur Modellierung von Hydraulik, Stofftransport und reaktionskinetischen Prozessen in Kläranlagen. Dissertation, Univ. Hannover, 1995

44/1995 **Malcherek, A.** Mathematische Modellierung von Strömungen und Stofftransportprozessen in Ästuaren. Dissertation, Univ. Hannover, 1995

45/1995 **Lege, T.** Modellierung des Kluftgesteins als geologische Barriere für Deponien. Dissertation, Univ. Hannover, 1995

46/1996 **Arnold, H.** Simulation dammbruchinduzierter Flutwellen. Dissertation, Univ. Hannover, 1996

47/1996 **Kolditz, O.** Stoff- und Wärmetransport im Kluftgestein. Habilitation, Univ. Hannover, 1996

48/1996 **Hunze, M.** Numerische Modellierung reaktiver Strömungen in oberflächenbelüfteten Belebungsbecken. Dissertation, Univ. Hannover, 1996

49/1996 **Wollschläger, A.** Ein Random-Walk-Modell für Schwermetallpartikel in natürlichen Gewässern. Dissertation, Univ. Hannover, 1996

50/1997 **Feist, M.** Entwurf eines Modellierungssystems zur Simulation von Oberflächengewässern. Dissertation, Univ. Hannover, 1997

51/1997 **Hinkelmann, R.** Parallelisierung eines Lagrange-Euler-Verfahrens für Strömungs- und Stofftransportprozesse in Oberflächengewässern. Dissertation, Univ. Hannover, 1997

52/1997 **Barlag, C.** Adaptive Methoden zur Modellierung von Strofftransport im Kluftgestein. Dissertation, Univ. Hannover, 1997

53/1997 **Saberi-Haghighi, K.** Zur Ermittlung der verformungsabhängigen Windbelastung bei Hängedächern. Dissertation, Univ. Hannover, 1997

54/1998 **Krüger, A.** Physikalische Prozesse im Nachklärbecken - Modellbildung und Simulation. Dissertation, Univ. Hannover, 1998

55/1998 **Wolters, A. H.** Zur Modellierung des instationären thermohydraulischen Betriebsverhaltens von Fernwärmeanlagen. Dissertation, Univ. Hannover, 1998

56/1999 **Jankowski, J. A.** A non-hydrostatic model for free surface flows. Dissertation, Univ. Hannover, 1999

57/1999 **Kopmann, R.** Mehrdimensionale Effekte in dimensionsreduzierten Gewässergütemodellen. Dissertation, Univ. Hannover, 1999

58/1999 **Kahlfeld, A.** Numerische Seegangsmodellierung als Bestandteil einer funktionellen Hafenplanung. Dissertation, Univ. Hannover, 1999

59/1999 Festschrift zum 60. Geburtstag von Prof. Dr.-Ing. Werner Zielke. Univ. Hannover, 1999

60/2000 **Kolditz, O., Zielke, W., Wriggers, P., Dürbaum, H.-J., Wallner, M.** 3. Workshop Kluft-Aquifere - Gekoppelte Prozesse in Geosystemen. Univ. Hannover, 2000

61/2001 **Malcherek, A.** Hydromechanik der Fließgewässer. Habilitation, Univ. Hannover, 2001

62/2001 **Thorenz, C.** Model Adaptive Simulation of Multiphase and Density Driven Flow in Fractured and Porous Media. Dissertation, Univ. Hannover, 2001

63/2001 **Kaiser, R.** Gitteradaption für die Finite-Elemente-Modellierung gekoppelter Prozesse in geklüftet-porösen Medien. Dissertation, Univ. Hannover, 2001

64/2001 **Rother, T.** Geometric Modelling of Geo-Systems. Dissertation, Univ. Hannover, 2001

65/2001 **Habbar, A.** Direkte und inverse Modellierung reaktiver Transportprozesse in klüftig-porösen Medien. Dissertation, Univ. Hannover, 2001