Simulation of Two-Phase Flow Processes in Heterogeneous Porous Media with Adaptive Methods

Von der Fakultät Bau- und Umweltingenieurwissenschaften der Universität Stuttgart
zur Erlangung der Würde eines Doktors der
Ingenieurwissenschaften (Dr.-Ing.) eingereichte Abhandlung

Vorgelegt von

Maren Paul

aus Göttingen

Hauptberichter:	Prof. Dr.-Ing. Rainer Helmig
Mitberichter:	Prof. Dr. rer. nat. Peter Bastian
Tag der mündlichen Prüfung:	28. Februar 2003

Institut für Wasserbau der Universität Stuttgart

2003

D93 Simulation of Two-Phase Flow Processes in Heterogeneous Porous Media with Adaptive Methods

CIP Titelaufnahme der Deutschen Bibliothek

Paul, Maren:
Simulation of Two-Phase Flow Processes in Heterogeneous Porous Media with Adaptive Methods / von Maren Paul. Institut für Wasserbau, Universität Stuttgart. - Stuttgart: Inst. für Wasserbau, 2003

(Mitteilungen / Institut für Wasserbau, Universität Stuttgart ; H. 120)
Zugl.: Stuttgart, Univ., Diss., 2003
ISBN 3-9337 61-23-9

Herausgegeben 2003 vom Eigenverlag des Instituts für Wasserbau
Druck: Sprint-Druck, Stuttgart

Vorwort und Danksagung

Die vorliegende Dissertationsschrift entstand während meiner Zeit als wissenschaftliche Mitarbeiterin am Institut für Computeranwendungen im Bauingenieurwesen der Technischen Universität Carolo-Wilhelmina zu Braunschweig und am Institut für Wasserbau der Universität Stuttgart. Die Arbeit wurde im Rahmen des von der *Deutschen Forschungsgemeinschaft* geförderten Graduiertenkollegs *Wechselwirkung von Struktur und Fluid* durchgeführt.

Ganz besonders bedanken möchte ich mich bei Herrn Prof. Dr. Rainer Helmig für die Übernahme des Hauptberichters. Auch in der anstrengenden Zeit des Institutsumzugs und des Neuaufbaus in Stuttgart hat er es durch seinen Elan und seine Motivationsfähigkeit geschafft, die Arbeitsgruppe zusammenzuhalten. Herrn Prof. Dr. rer. nat. Peter Bastian danke ich für die Übernahme des Mitberichts. Auch für die Beantwortung der UG-spezifischen Fragen möchte ich mich herzlich bedanken.

Ein ganz großes Dankeschön möchte ich zudem an dieser Stelle an meine Kolleginnen und Kollegen vom IWS, Lehrstuhl für Hydromechanik und Hydrosystemmodellierung, in Stuttgart richten. Ich bin froh, dass sich auch im "Ländle" die herzliche und intensive Atmosphäre halten konnte.

Ein besonderer Dank gilt Ulrich Ölmann, der mir geduldig geholfen hat, in mathematischen Tiefen vorzudringen, Lina Neunhäuserer, mit der ich nächtelang den irc geteilt habe und Prudence Lawday, die es tapfer auf sich genommen hat, das Englisch meiner Arbeit gegenzulesen. Der größte Dank gebührt allerdings Hartmut und Debelt Jakobs, die mich auf dem nicht immer geraden Weg zur Promotion begleitet haben und ab und zu den richtigen Kick in die richtige Richtung liefern konnten.

Schließlich möchte ich mich noch bei meiner Familie bedanken, die mich in meinem Weg unterstützt hat und mir auch jederzeit eine Zuflucht gewährte, wenn die Arbeit überhand nahm. Und ganz besonders möchte ich Thomas danken, der mit viel Geduld verhindert hat, dass ich während der anstrengenden Phasen verhungert bin.

"Captain, es gibt eine grundlegende und sehr wertvolle Aussage in der Wissenschaft.
Sie ist ein Zeichen von Weisheit und sie lautet: 'Ich weiss es nicht'."

"Captain, the most elementary and valuable statement in science,
the beginning of wisdom is: 'I do not know'."

Lt. Cmdr. Data, Sternzeit 42193.6
The Next Generation [TNG]: Illusion oder Wirklichkeit
(Where Silence Has Lease)

Contents

List of Figures

List of Tables

Notation

The following table shows the significant symbols used in this work. Local notations are explained in the text.

Symbol	Definition	Dimension
Greek Letters:		
α_{up}	upwinding parameter	[-]
γ_{ij}^{BOX}	transmissivity integral	[-]
Γ	boundary between two subdomains	[-]
$\Gamma_{\alpha,D}$	part of a domain comprising Dirichlet boundary conditions	[-]
$\Gamma_{\alpha,N}$	part of a domain comprising Neumann boundary conditions	[-]
Γ_{B_i}	integration path along the boundary of box B_i	[-]
$\delta_{\alpha w}$	*Kronecker*-delta (with $\alpha \in \{w,n\}$)	[-]
Δt	time step	[s]
Δx	element size	[m]
ε	residuum	[-]
η_i	set of all neighboring nodes of node i	[-]
η_j	error indicator or estimator value associated with element j	[-]
λ	*Brooks-Corey*-parameter (grain size distribution)	[-]
λ_α	mobility of phase α	[(m s)/kg]
$\overline{\lambda}$	mean mobility	[(m s)/kg]
μ	dynamic fluid viscosity	[kg/(m s)]
μ_α	dynamic fluid viscosity of phase α	[kg/(m s)]
ρ_α	fluid density of phase α	[kg/m^3]
ϕ	porosity	[-]
ϕ_{eff}	effecitve porosity	[-]
$\phi_{\text{eff},\alpha}$	effective porosity for phase α	[-]
$\psi_{\alpha i}$	total potential of phase α at node i	[-]

Latin Letters:

B_i	control volume or box for node K_i	[-]
∂B_i	boundary of box B_i	[-]
cor^{max}	value controlling the coarsening bound for the 'max value' method	[-]
cor^{mean}	value controlling the coarsening bound for the 'mean value' method	[-]
$\mathbf{D}$	dispersion tensor	[m^2/s]
e	scalar entity (for the $p_w - S_n$-formulation: $e_\alpha = S_n \phi \rho_\alpha$)	[kg / m^3]
E_i	set of elements connected to node K_i	[-]
$\mathbf{F}$	convective flow and diffusion (for the $p_w - S_n$-formulation: $\mathbf{F}_\alpha = \rho_\alpha \mathbf{v}_\alpha$)	[kg / (m^2 s)]
f_α	fractional flow function for phase α	[-]
$\mathbf{g}$	vector of gravitational acceleration $(0, 0, -\mathrm{g})^T$	[m/s^2]
g	(scalar) gravitational acceleration	[m/s^2]
G	solution domain	[-]
∂G	boundary of solution domain	[-]
h	piezometric head	[m]
$k_{r\alpha}$	relative permeability for phase α	[-]
$\mathbf{K}$	tensor of (intrinsic) permeability	[m^2]
$\mathbf{K}_\alpha$	tensor of permeability for phase α	[m^2]
$\mathbf{K}_f$	tensor of hydraulic conductivity	[m/s]
K_i	node of the finite element mesh	[-]
$m_{\alpha i}$	flow of phase α over the Neumann boundary $\partial B_i \cap \Gamma_{\alpha,N}$	[kg / (m^2 s)]
$M_{ij}^{(lump)}$	(lumped) mass matrix	[-]
ML	maximum number of mesh refinement level	[-]
$\mathbf{n}$	outer normal vector of ∂G	[-]
N_i	basis functions for node i	[-]
N	number of elements with an error of η	[-]
p_c	capillary pressure	[Pa]
$p_c\mathrm{max}$	maximum capillary pressure for a linear $p_c(S_w)$-relationship	[Pa]
$p^i_{c,min}$	minimal entry pressure	[Pa]
p_d	entry pressure ($p_c(S_w)$-relationship after *Brooks-Corey*)	[Pa]
p_d^Δ	difference of the entry pressures for two subdomains having different soil properties	[Pa]
Pe	*Peclet* number	[-]
p_n	pressure of the non-wetting phase	[Pa]
p_w	pressure of the wetting phase	[Pa]
Q_α	source or sink of phase α in the domain	[1/s]

r	sources and sinks (for the $p_w - S_n$-formulation: $r_\alpha = \rho_\alpha Q_\alpha$)	[kg / (m^3 s)]
ref^{max}	value controlling the refinement bound for the 'max value' method	[-]
ref^{mean}	value controlling the refinement bound for the 'mean value' method	[-]
S_α	saturation of phase α	[-]
$S_{\alpha r}$	residual saturation of phase α	[-]
S_w^d	downstream water saturation	[-]
S_e	effective saturation	[-]
S_w^u	upstream water saturation	[-]
S_w^*	threshold saturation	[-]
S_{w_i}	water saturation at the inflection point of the fractional flow function	[-]
S_{w_t}	water saturation at the tangent point of the fractional flow function	[-]
t	time	[s]
tol$_{coarsen}$	tolerance for coarsening	[-]
tol$_{refine}$	tolerance for refinement	[-]
u	unknown quantity	[-]
$\mathbf{v}$	*Darcy* velocity	[m/s]
$\mathbf{v}_\alpha$	phase velocity of phase α	[m/s]
$\mathbf{v}_{a,\alpha}$	average phase velocity of phase α	[m/s]
$\mathbf{v}_a$	average velocity	[m/s]
$\mathbf{v}_t$	total velocity	[m/s]
V_α	volume fraction of phase α	[m^3]
V_i	volume for the box associated with node *i*	[m^3]
W_i	weighting function for node *i*	[-]
z	elevation head	[m]
z_i	geodetic height of node *i*	[m]

Subscripts:

α	phase, either wetting (*w*) or non-wetting (*n*)
d	downstream
n	non-wetting phase (e.g. oil, gas)
u	upstream
w	wetting phase (e.g. water)

Superscripts:

˜	approximation
ˆ	discrete value at a node
cent	central weighting method
n	time step
up	(fully) upwind weighting method

Abstract

In recent years the demand for numerical simulations which help to support the work of engineers has constantly gained weight. In the same way the computational power advances, the complexity of the problems and demands posed to the numerical simulation tools increases, too.

This thesis expands the simulation tool MUFTE-UG for two-phase flow processes in porous media with adaptive methods The approaches presented here can be divided into two parts:

On the one hand, a space adaptive method is introduced. Here, the element-size resolution throughout the discretization mesh is changed in certain areas from time step to time step. A marking of the relevant elements which need to be refined or coarsened is realized by an empirically derived error indicator. While at first three different indicators are compared, the final investigations of the various test cases are performed with an indicator locating a steep gradient of the saturation distribution in the system. As it shows that the applied h-adaptive strategy is not mass conservative due to mesh manipulations, an algorithm is developed which ensures the mass preservation for coarsening as well as for refinement.

On the other hand, the discretization method is adaptively adjusted inside the domain. Since advection-dominated processes require an other numerical treatment than diffusion-dominated processes (in the here presented case realized by the application of a 'fully upwinding' or a 'centrally weighted' scheme), at first a suitable indicator needs to be found which accounts for these processes. For this, the two-phase (element) *Peclet* number is derived which describes the ratio between advection and diffusion.

The developed methods are applied to homogeneous and heterogeneous test cases. For the choice of the homogeneous test cases it is considered that the here introduced methods need to be capable of handling purely advection-dominant problems (e.g. *Buckley-Leverett* problem), purely diffusion-dominant problems (e.g. *McWhorter* problem) and problems, where both effect appear in the domain at the same time (e.g. Sandbox problem). The heterogeneous test case resembles the Sandbox problem with a lense.

Overall it can be said that the deployed space adaptive methods work very well for all the investigated test cases. The results obtained by the adaptive choice of discretization method are only mildly satisfactory. Here it shows, especially for the heterogeneous case, that the switch to the centrally weighted scheme needs a very careful adjusting.

Kurzfassung

In den vergangenen Jahren hat die Nachfrage nach numerischen Simulationsprogrammen, die eine Unterstützung der täglichen Arbeit von Ingenieuren bieten, ständig an Gewicht zugenommen. Im gleichen Maße in dem die Rechnerleistung zunimmt, steigen auch die Komplexität der Fragestellungen und die Anforderungen an die Simulationsprogramme.

Diese Arbeit erweitert das Simulationsprogramm MUFTE-UG für Zwei-Phasen Strömungsprozesse in porösen Medien um adaptive Methoden. Die hier vorgestellten Vorgehensweisen können in zwei Bereiche unterteilt werden:

Auf der einen Seite wird eine orts-adaptive Methode vorgestellt. Dabei wird die Elementgröße in bestimmten Gebieten des Diskretisierungsnetzes von Zeitschritt zu Zeitschritt angepasst. Das Markieren der relevanten Elemente, die verfeinert oder vergröbert werden sollen, wird mittels eines empirischen Fehlerindikators realisiert. Zunächst werden drei verschiedene Indikatoren verglichen. Die letztendlichen Untersuchungen werden mit einem Indikator durchgeführt, der einen steilen Gradienten der Sättigung im Gebiet lokalisiert. Da sich zeigt, dass das verwendete h-adaptive Verfahren aufgrund von Netz-Manipulationen nicht massenkonservativ ist, wird ein Algorithmus entwickelt der die Erhaltung der Masse für das Vergröbern und das Verfeinern sicherstellt.

Auf der anderen Seite findet ein adaptives Anpassen der Diskretisierungsmethode innerhalb des Gebietes statt. Da advektions-dominante Prozesse eine andere numerische Behandlung als diffusions-dominante Prozesse benötigen (was in dieser Arbeit durch Anwendung der ’fully upwinding’ oder ’zentral gewichteten’ Methode geschieht), muss zunächst ein geeigneter Indikator gefunden werden, der diese Prozesse identifiziert. Hierfür wird die Zwei-Phasen (element-basierte) *Peclet*-Zahl hergeleitet, die das Verhältnis zwischen Advektion und Diffusion beschreibt.

Die entwickelten Methoden werden für homogene und heterogene Testfälle eingesetzt. Bei der Wahl der homogenen Fälle wurde berücksichtigt, dass die vorgestellten Methoden in der Lage sein sollen sowohl rein advektions-dominante Probleme (z.B. *Buckley-Leverett* Problem) und rein diffusions-dominante Problem (z.B. *McWhorter* Problem) als auch Probleme behandeln können soll, bei denen beide Effekte gleichzeitig im Gebiet auftreten (z.B. Sandbox Problem). Als heterogenes Test wird das Sandbox Problem mit einer Linse gewählt.

Zusammenfassend kann gesagt werden, dass die Ergebnisse der verwendeten orts-adaptiven Methoden für alle Testfälle sehr gut sind. Die Ergebnisse, die mit der adaptiven Anpassung der Diskretisierungsmethode erreicht wurden sind nur mäßig zufriedenstellend. Hier zeigt sich, dass, insbesondere für den heterogenen Fall, der Übergang zur zentral gewichteten Methode sehr sorgfältig gewählt werden muss.

1 Introduction

1.1 Motivation

Flow processes in porous media play an important part in the human society, although this may not always be evident. A theme of topical interest, for example, is the issue of dike stability (Figure 1.1). As the increase of natural disasters related to water and especially the floods of the river Elbe in the eastern parts of Germany this year have shown, a correct prognosis of the stability of protective structures is very important. For these predictions, an understanding of the relevant flow processes of water and air inside the dikes is necessary. Numerical simulations of two-phase flow processes in porous media represent a powerful tool for providing local experts with knowledge about these flow behaviors.

Another important issue is the simulation of water and NAPL flows in the soil. NAPL stands for non-aqueous phase liquid. NAPLs are usually liquid organic pollutants which are practically immiscible with water. These liquids can be divided into LNAPLs (liquids lighter than water) and DNAPLs (liquids denser than water). In most cases, the interest in simulating water - NAPL flows lies in tracking contaminants in the soil and in surveying and improving soil recovery strategies (e.g. *Class* [17]). If the infiltration of NAPLs into the soil takes place in the vadose or unsaturated zone, it may also be important to track the movement of the gas in the soil and therefore consider three-phase flow.

These examples represent only a small part of the wide field of multiphase flow processes. As numerous as the applications are the special cases one has to deal with. Certain aspects, however, can be found in a large number of problems. In the following, two of these aspects are explained further.

The first one is a so-called ”sharp front”. Here, diffusive effects play only a negligible role and the transition between a region completely saturated with one fluid (e.g. water) and a region completely saturated with another fluid (e.g. oil) is (nearly) abrupt. Sharp fronts pose a difficult problem for a numerical scheme. The difficulties lie not only in the choice of a sufficiently small element-size resolution, but also in the choice of a suitable discretization technique, as the numerical solutions for problems including sharp fronts tend to show oscillations if the correct measures are not taken.

The second aspect concerns heterogeneities, i.e. regions with different material properties, such as, for example, porosity and permeability. They have a great influence on the flow processes, as they affect flow paths and the intrusion into soil layers, for example, and therefore

Figure 1.1: Water flows through a broken dike

the mass distribution of phases in a system. Especially in civil engineering applications, the number of heterogeneous problems is greater than that of homogeneous ones.

The aspects mentioned above increase the - already large - complexity of two-phase flow problems even more. Especially for real-life problems (as opposed to academic problems), it is not only important to get correct results as a basis for predictions, but also to get results fast. The prediction of the stability for a sea-dike in a storm surge with rising water level may serve as an example: the computation should be faster than realtime in order to undertake the right measures (bring in sandbags or evacuate the residents) in time.

The scenario depicted in Figure 1.2 illustrates some of the problems involved in the numerical simulation of two-phase flow applied to engineering problems. It shows a highly heterogeneous dam, which is infiltrated from the left by water (indicated by the shaded area). In the upper left part of the domain, a small lake is situated, which serves as a constant inflow condition to the dam.

Firstly, the domain is large when compared to academic problems. In order to keep the computational time reasonable, one would choose relatively large elements. However, in order to represent the front of the water well enough, small elements are needed in this region. An increase in the number of elements inevitably leads to an increase in computational storage requirements and execution time. Here, it is important to keep in mind that the problem should still remain computable.

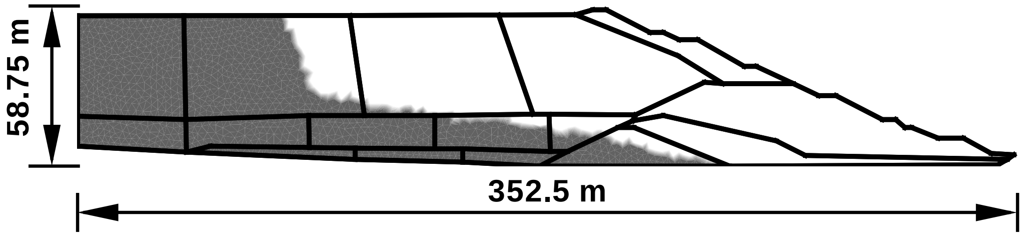


Figure 1.2: System of a real-life dam. The different areas represent parts with different soil parameters.

Secondly, the domain is composed of a large number of subdomains with different soil properties. The interfaces between, for example, different permeabilities constitute a challenge to the numerical scheme. In the case considered here, multiple interfaces have to be considered in one time step. Also, the flow is not always perpendicular or parallel to the interfaces. And, last but not least, a large variety of permeabilities leads to a non-well conditioned system matrix and thus to a difficult solving behavior of the problem.

Thirdly, in some regions of the domain, advective processes dominate whereas, in other regions, diffusion-dominated processes are present at the same time. This makes a further demand on the numerical scheme, as each of these processes requires special treatment.

All together, these aspects result in the need for adaptive methods which, on the one hand, improve the mesh-size resolution in critical areas (see Figure 1.3) and, on the other hand, can be employed to steer the behavior of the numerical scheme. In this work, adaptivity means that, according to an indicator which identifies regions of higher importance, steps are taken to improve the solution in these regions whereas regions of lower importance are treated with lower computational endeavor. This applies to the mesh size as well as the numerical scheme.

1.2 Goal and structure

In this work, the adaptive capabilities of the numerical simulator MUFTE-UG are expanded. The program combines the physical context and the discretization methods of MUFTE (Multiphase Flow, Transport, and Energy Model, Institut für Wasserbau, Stuttgart, e.g. *Helmig et al.* [29], *Class et al.* [19]) with the solution methods and multigrid techniques of the software toolbox UG (Unstructured Grids, Interdisziplinäres Zentrum für Wissenschaftliches Rechnen, Heidelberg, e.g. *Bastian et al.* [5]).

A powerful adaptive mesh refinement, even for parallel and three-dimensional simulations, has already been implemented by *Lang* [44]. The aim of this work is to apply these techniques to the strongly coupled hyperbolic-parabolic two-phase flow equations with particular reference to heterogeneities. Additionally, the goal is to provide an error indicator

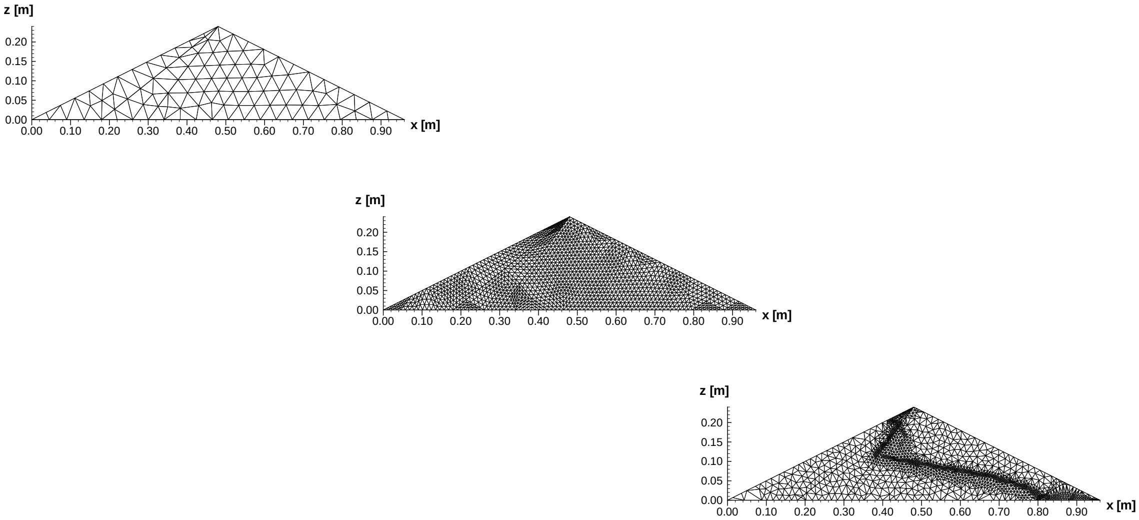


Figure 1.3: Different meshes for the simulation of a dike. Left: coarse mesh; middle: fine mesh; right: adaptive mesh.

suitable for a large set of applications. It should not only be applicable to the incompressible, isothermal two-phase flow investigated here, but also to even more complex problems such as, for example, non-isothermal multiphase multicomponent flow. The emphasis lies on its practical applicability to engineering tasks: the indicator should be easy to implement and operate.

The adaptive mesh-refinement techniques were deployed to a variety of test cases ranging from diffusion-dominated flows in a homogeneous medium to diffusive-advective flows in a medium with a heterogeneity. Next to an adaptive mesh refinement and coarsening, an adaptive choice of the numerical scheme (choice of the upwinding weighting) is also realized to meet the requirements of regional flow behaviors. In order to arrive at a suitable criterion for this choice, the two-phase *Peclet* number is derived. For a third adaptive variant, the *Peclet* number is used additionally to control the mesh-refinement and -coarsening criteria. This leads to the following three cases: adaptive choice of mesh via indicator – adaptive choice of numerical scheme via *Peclet* number – adaptive choice of mesh via indicator and *Peclet* number.

■□□■
□■■□

The different states of adaptivity employed in this work are brought together in the following graphic. In order to illustrate the current combination of options, a small icon will indicate, which conditions are active. The icon in the margin, for example, represents the state 'homogeneous domain, adaptive simulation, *Peclet* upwinding, standard adaptive refinement'. For consistency's sake, the uniform simulations will also be marked by an icon.

homogeneous	**uniform**	**standard upwinding**	**standard adaptive**
heterogeneous	**adaptive**	**Peclet upwinding**	**Peclet adaptive**

This thesis is structured as follows:

At first, the fundamentals of the physical and mathematical modeling of two-phase flow are presented in Chapter 2. Here, the governing partial differential equations with their complementary constitutive relationships are introduced. Also, the discretization techniques applied in this work are explained. At this point, the focus is the special treatment of heterogeneities.

Next, the test cases for flow in a homogeneous medium are presented in Chapter 3. Three representative cases covering the whole bandwidth of complexity of two-phase flow equations are chosen. For each of these cases, the simulation with a uniformly refined mesh is carried out. They serve as reference solutions for the following adaptive simulations.

The basic principles of adaptive methods are then presented in Chapter 4. At first, an overview of different adaptive methods is given. Then, the utilized error indicators are discussed. Although the discretization method used in this work is considered as mass conservative, the adaptive method applied is not. Due to mesh manipulations, the mass in the system can increase or decrease, depending on the problem. The background for this performance and the algorithm ensuring mass conservation, which was developed in this work for the simulation tool MUFTE-UG used here, are described in Chapter 5.

These chapters lead the way to the application of the chosen adaptive methods for the homogeneous test cases introduced earlier. Based on the results obtained by these simulations (Chapter 6), a proposition for a criterion which is suitable for indicating the need for a change in the numerical treatment is formulated. This criterion is based on the two-phase *Peclet* number (*Pe*), which is introduced in Section 7.1. The investigations of the homogeneous test cases are completed in Chapter 7 by the application of the *Pe*-criterion on the one hand to the choice of the numerical scheme, on the other hand to the refinement and coarsening strategies.

In Chapter 8, the heterogeneous test case is introduced. Here, the uniform as well as adaptive simulation results with and without the developed *Pe*-criterion are presented.

Finally, the efficiency of the adaptive algorithm is underlined by comparing numerical simulation results with experimental data and applying it to a real-life dam in Chapter 9.

Chapter 10 ends this thesis with a summary and outlook.

2 Physical and mathematical background

2.1 Basic definitions and concepts

Averaging process

A wide range of materials can be considered as porous media. Among them are, for example, synthetic foams (used as an impact absorber in cars) and bone materials. The porous material that will be considered in this thesis, however, is the subsurface. The subsurface consists in general of soil material (e.g. sand, clay, loose or solid rock matrices) and is connected to one or more of the following: the atmosphere, surface water and the sea (*Dyck and Peschke* [22]). It may be fully or partially saturated by liquids, such as for example water.

Whether the porous media are regarded as compressible or not, they all have this characteristic in common: they consist of a solid material (such as soil grains) and void spaces of different sizes in between (either connected or not). These spaces are also called **pores**.

Especially for natural materials, it is impossible to describe the geometry of the pores and grains completely. Therefore, the porous media flow models are often based on a continuum approach. Here, the properties on the microscale[1] are averaged over a **representative elementary volume (REV)**, which represents the macroscale ($\geq$ mm). The discontinuities that are present on the microscale (for example microcracks, fluid interfaces) are now 'smeared' and no longer distinguishable (see Figure 2.1). This averaging process creates a new set of parameters, which are only available on the macroscale. These are, for example, the saturation and the porosity. They will be explained further in the following.

The difficulties associated with the continuum approach lie in the choice of the size of a REV: If too large a REV is chosen, process-relevant discontinuities (for example fractures or lenses with a different permeability) may also be 'smeared out' and the simulation results may therefore give a wrong picture of the actual circumstances.

For the examples shown in this thesis, it is always assumed that a valid REV is chosen.

Wetting – non-wetting fluid

When two-phase flow in porous media are considered, the fluids cannot only be distinguished by their specific fluid properities such as density or viscosity. They also differ in their **contact angle** θ towards a solid surface, for example the grain surface. By definition,

[1] On the microscale ($<$ mm), the grain geometry is described precisely and the different fluids are separated from each other by a distinct interface in the pores.

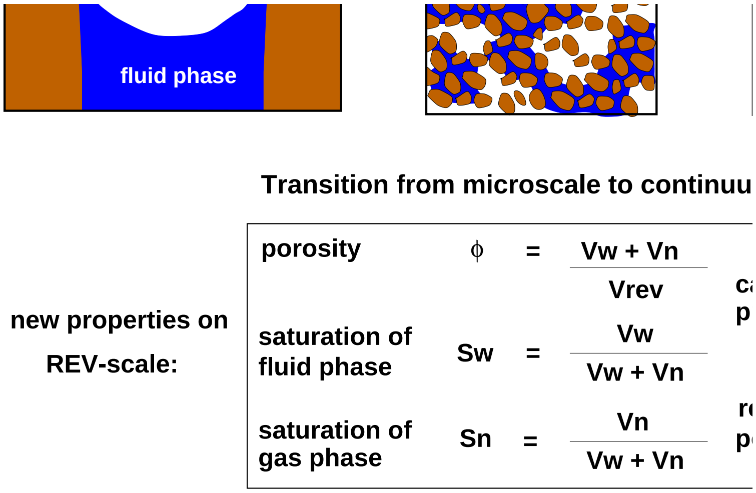

fluid phase
Transition from microscale to continuu
new properties on
REV-scale:
porosity
$\phi = \frac{Vw + Vn}{Vrev}$
saturation of
fluid phase
$Sw = \frac{Vw}{Vw + Vn}$
saturation of
gas phase
$Sn = \frac{Vn}{Vw + Vn}$
ca
p
re
p

Figure 2.1: Transition from the microscale to a continuum approach: the interfaces between the different phases are no longer clearly distinguishable and new properties are introduced. (V_α = volume fraction of phase α)

the contact angle lies between 0° and 180°. If the angle is smaller than 90°, the fluid is called the **wetting phase** (see Figure 2.2). If the contact angle is larger than 90°, the fluid is called the **non-wetting phase**.

In the case of the fluids considered in this thesis, water represents the wetting fluid and will in the following be marked by the index w. The non-wetting phase is represented by oil or gas and is marked by the index n.

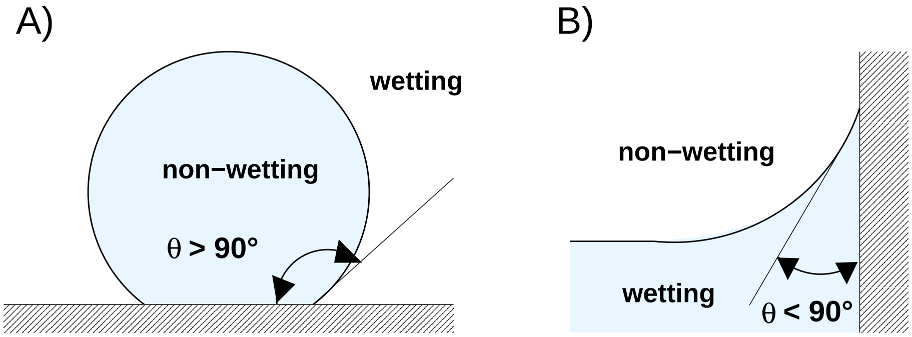


Figure 2.2: Contact angles between wetting and non-wetting fluids

Saturation

Of these two (immiscible) fluids, each one fills its share of the pore space. Although the exact location of the fluid particles is (due to the averaging process) no longer known, how much of the pore space in a REV is filled with a certain phase it is still of interest. This is described by the **saturation of phase α (S_α)**:

$$S_\alpha = \frac{\text{volume of fluid } \alpha \text{ within the REV}}{\text{volume of the pore space within the REV}} \tag{2.1}$$

It is assumed that all fluid phases fill the pore space completely, so the different saturations must add up to one:

$$\sum_\alpha S_\alpha = S_n + S_w = 1 \tag{2.2}$$

Porosity

As the name implies, a porous medium consists of a solid matrix and pores. The amount of pore space in a porous medium is referred to as **porosity** ϕ:

$$\text{porosity } \phi = \frac{\text{volume of the pore space}}{\text{volume of the REV}}$$

The pores in a medium usually consist of connected flow paths and so-called dead-end pores. A fluid flowing through a porous medium can only move along the flow paths. This is accounted for by the **effective porosity**:

$$\text{effective porosity } \phi_{\text{eff}} = \frac{\text{volume of the flow paths}}{\text{volume of the REV}}$$

In the case of a two-phase flow, the presence of one fluid affects the flow paths of the other fluid. So fluid *w* can only flow where fluid *n* is not. In order to describe this phenomenon in terms of volumes, the **effective porosity for phase** α is used:

$$\text{effective porosity for phase}\,\alpha\ \phi_{\text{eff},\alpha} = \frac{\text{volume of the flow paths for phase}\,\alpha}{\text{volume of the REV}}$$

$$\begin{aligned} \phi_{\text{eff},w} &= (1-(S_{wr}+S_n))\cdot\phi \\ \phi_{\text{eff},n} &= (1-(S_w+S_{nr}))\cdot\phi \end{aligned} \tag{2.3}$$

Here, S_{wr} and S_{nr} represent the residual saturations of the wetting and non-wetting phase respectively. They will be explained further in the next section.

These different definitions for the porosity now have the following consequences for the concepts of velocities:

Velocity

For single-phase flow, the flow velocity of a fluid in a porous medium is usually given by the *Darcy* velocity $\mathbf{v}$:

$$\mathbf{v} \;=\; -\mathbf{K}_f \;\mathbf{grad}\, h \tag{2.4}$$

Here, $\mathbf{K}_f$ is the tensor of hydraulic conductivity and h is the piezometric head. This *Darcy* velocity is derived from an experimental point of view: it describes the amount of fluid (Q, in [m^3/s]) that flows over the whole cross-section of the domain (A_{domain}, in [m^2]). So this velocity assumes that the fluid flow is not restricted to the pores of a porous media alone, but uses the complete area.

Of course, in reality the flow only takes place in the pores, so the same amount of fluid has to flow through a much smaller area with a much higher velocity. The resulting velocity is called **average velocity** $\mathbf{v}_a$:

$$\mathbf{v}_a \;=\; \frac{\mathbf{v}}{\phi_{\text{eff}}} \tag{2.5}$$

For an extension to multiphase flow, we first rewrite the *Darcy* law (2.4):

$$\begin{aligned} \mathbf{v} \;&=\; -\mathbf{K}_f \;\mathbf{grad}\, h \\ &=\; -\mathbf{K}\,\frac{\rho\, g}{\mu}\;\mathbf{grad}\,(\frac{p}{\rho\, g} \;+\; z) \\ &=\; -\mathbf{K}\,\frac{1}{\mu}\,(\mathbf{grad}\, p + \rho\, \mathbf{g}) \end{aligned} \tag{2.6}$$

Here, the following relationships are included:

1. $\mathbf{K}_f \;=\; \mathbf{K}\,\frac{\rho g}{\mu}$

   with the properties:

$\mathbf{K}_f$	:	tensor of hydraulic conductivity	[m/s]
$\mathbf{K}$	:	tensor of (intrinsic) permeability (depends solely on the soil properties)	[m^2]
ρ	:	fluid density	[kg/m^3]
g	:	(scalar) gravitational acceleration	[m/s^2]
μ	:	dynamic fluid viscosity	[kg/(m s)]

2. $h = \frac{p}{\rho g} \;+\; z$

   This states that the hydraulic head h is the sum of the pressure head $\frac{p}{\rho g}$ and the elevation head z.

3. $\rho\;\mathbf{grad}\,(g z) \;=\; \rho\,\mathbf{g}$

   The multiplication of the scalar gravitational acceleration with the gradient of the elevation head results in the vector of gravitational acceleration $\mathbf{g}$ with the components (0, 0, -g)^T.

Next, the *Darcy* law is expanded to multiphase flow (see *Scheidegger* [57], *Helmig* [31]):

$$\begin{aligned}\mathbf{v}_\alpha &= -\frac{1}{\mu_\alpha}\mathbf{K}_\alpha\left(\mathbf{grad}\, p_\alpha - \rho_\alpha\,\mathbf{g}\right)\\ &= -\frac{k_{r\alpha}}{\mu_\alpha}\mathbf{K}\left(\mathbf{grad}\, p_\alpha - \rho_\alpha\,\mathbf{g}\right)\\ &= -\lambda_\alpha\mathbf{K}\left(\mathbf{grad}\, p_\alpha - \rho_\alpha\,\mathbf{g}\right)\end{aligned} \tag{2.7}$$

$\mathbf{v}_\alpha$ is called the **phase velocity**, where α represents the subscript for the different phases w and n. $\mathbf{K}_\alpha$ and $k_{r\alpha}$ denote the permeability and the relative permeability for each phase respectively. The relative permeability will be discussed in the next section. The ratio $\frac{k_{r\alpha}}{\mu_\alpha}$ is also called **mobility** λ_α.

Now it is possible to define an **average phase velocity** $\mathbf{v}_{a,\alpha}$:

$$\mathbf{v}_{a,\alpha} = \frac{\mathbf{v}_\alpha}{\phi_{\mathrm{eff},\alpha}} \tag{2.8}$$

2.2 Constitutive relationships

Capillary pressure

As shown in Figure 2.2, the interface between a wetting and a non-wetting phase is always curved. Due to equilibrium constraints, the pressure of the non-wetting phase has to be larger at the interface than the pressure of the wetting phase. The difference between those two pressures is called **capillary pressure** p_c:

$$p_c = p_n - p_w \tag{2.9}$$

On the microscale, the capillary pressure depends on the interfacial tension and the pore radius (see, for example, *Helmig* [31]). The smaller the pore radius, the larger the capillary pressure. This implies that, in the case of a drainage of the wetting phase, the larger pores are drained first and the fluid remains in the smaller pores. The resulting amount of wetting phase that is left in the porous medium is called the **residual saturation** S_{wr} (of the wetting phase).

In analogy to the residual water saturation, a residual gas saturation S_{nr} can also exist. This means that, if the porous medium is filled to the maximum with water, small entrapped gas or oil bubbles still exist that will not vanish in a simple infiltration process of water.

The existence of the residual saturations makes it clear that the concept of the capillary pressure only makes sense for saturation states between these two extrema. In order to describe this in-between state and to scale the behavior for different residual saturations, the **effective saturation** S_e is introduced:

$$S_e = \frac{S_w - S_{wr}}{1 - S_{wr} - S_{nr}} \qquad 0 \leq S_e \leq 1 \tag{2.10}$$

The relationship between the decrease of the saturation of the wetting phase and the increase of the capillary pressure is expressed in the capillary pressure - saturation function. By convention, this function is mostly expressed in S_w rather than S_e:

$$p_c = p_c(S_w) \tag{2.11}$$

Due to the irregular (and unknown) pore geometry, this relationship cannot be determined analytically. Next to an experimental identification of the $p_c(S_w)$-function, which is complicated and costly, various empirical approaches exist.

The simplest one is to assume a linear dependence (see Figure 2.3 (left)):

$$p_c(S_w) = p_c\mathrm{max} \left(1 - \frac{S_w - S_{wr}}{1 - S_{wr} - S_{nr}}\right) \tag{2.12}$$

Here, $p_c\mathrm{max}$ represents the maximum capillary pressure for $S_{w(r)}$ = 0.

A more sophisticated approach is the one after *Brooks and Corey* [14]:

$$p_c(S_w) = p_d \left(\frac{S_w - S_{wr}}{1 - S_{wr} - S_{nr}}\right)^{-\frac{1}{\lambda}} \tag{2.13}$$

p_d represents the **entry pressure** (see Figure 2.3 (right)). This is the minimum pressure that has to be overcome in order for the non-wetting phase to displace the wetting phase in the system. This pressure is also called **displacement pressure**. The λ-parameter describes the grain-size distribution of the soil. A small λ describes a highly non-uniform material while a large λ-value stands for a material, consisting of a single grain size.

Relative permeability

Beside the capillary pressure, the **relative permeability** k_r represents another important quantity for multiphase flow.

Each phase α has a $k_{r\alpha}$, which can be seen as a scaling parameter. It describes to what extent the presence of one fluid disturbs the flow behavior of another fluid. This disturbance is due to interaction forces as well as to a change of possible flow paths.

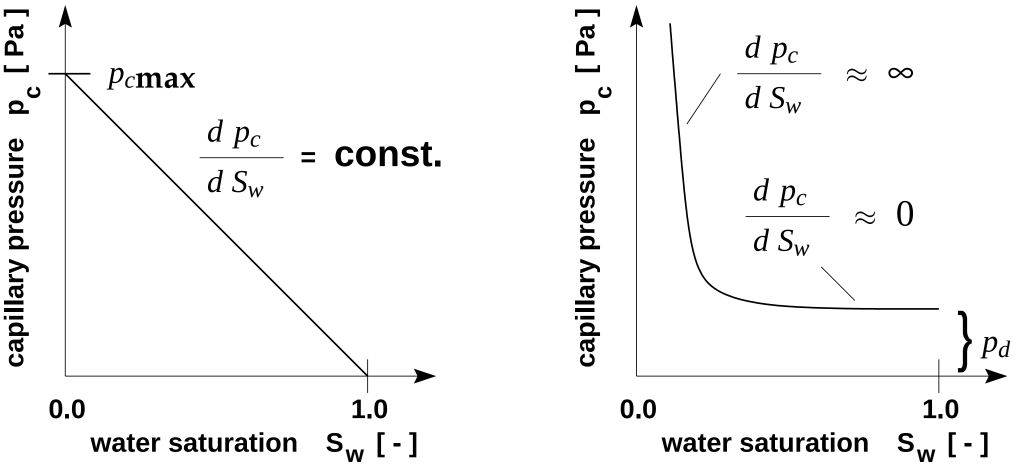


Figure 2.3: Capillary pressure functions. Left: linear; right: after *Brooks and Corey*

For example, if the saturation of the non-wetting phase is increased, it will at first fill the larger pores and will therefore restrict the wetting phase to the smaller pores and thus cause longer flow paths.

So if one phase fills out the available pore space completely (not regarding the residual saturation of the other phase), the relative permeability for this phase is 1. On the other hand, if a phase is only present in residual saturation and therefore immobile, the relative permeability is 0.

As it is impossible to describe the complex pore geometry precisely, the relationship between the relative permeability and the saturation can only be described quantitatively.

Again, the simplest approach is to assume a linear relationship (see Figure 2.4 (left)):

$$k_{rw}(S_w) = S_e \tag{2.14}$$

$$k_{rn}(S_w) = 1 - S_e \tag{2.15}$$

Here, again an approach after *Brooks and Corey* [14] is introduced (see Figure 2.4 (right)):

$$k_{rw}(S_w) = S_e^{\frac{2+3\lambda}{\lambda}} \tag{2.16}$$

$$k_{rn}(S_w) = (1 - S_e)^2 \left(1 - S_e^{\frac{2+\lambda}{\lambda}}\right) \tag{2.17}$$

As can be seen in Figure 2.4 (right), the two relative permeabilities do not add up to one for this case. This means that both phases influence and slow each other down.

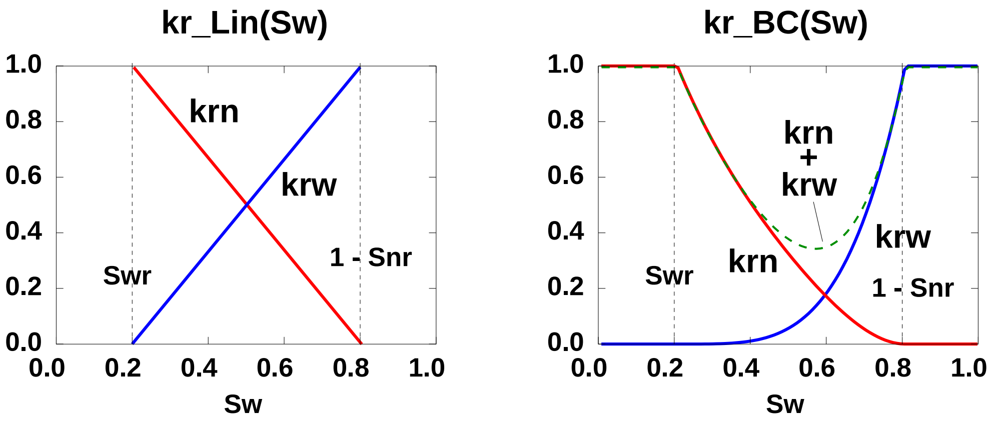


Figure 2.4: Relative permeability functions. Left: linear; right: after *Brooks and Corey*

2.3 Two-phase flow equations

The balance equations for the flow of two immiscible fluid phases in porous media are given by the conservation of mass (Eq. (2.18)) and the conservation of momentum in terms of the extended *Darcy* law (Eq. (2.19), see also Section 2.1):

$$\frac{\partial(S_\alpha \phi \rho_\alpha)}{\partial t} + \text{div}\,(\rho_\alpha \mathbf{v}_\alpha) - \rho_\alpha Q_\alpha = 0 \tag{2.18}$$

$$\mathbf{v}_\alpha = -\frac{k_{r\alpha}}{\mu_\alpha}\mathbf{K}\,(\mathbf{grad}\,p_\alpha - \rho_\alpha \mathbf{g}) \qquad \alpha \in \{w,n\} \tag{2.19}$$

For the two-phase flow of water and oil or gas, water represents the wetting (w) and oil or gas the non-wetting (n) phase (see p. 7).

In Eqs. (2.18) and (2.19), the variables denote the following quantities (in order of appearance): S_α unknown saturation, ϕ porosity, ρ_α density, t time, $\mathbf{v}_\alpha$ velocity, Q_α a source or sink in the domain, $k_{r\alpha}$ relative permeability, $\mathbf{K}$ intrinsic permeability tensor, μ_α dynamic viscosity, p_α unknown pressure, and $\mathbf{g}$ vector of gravitational acceleration. Inserting Eq. (2.19) into Eq. (2.18) yields the two-phase flow equation:

$$\begin{aligned} L_\alpha(p_\alpha, S_\alpha) \quad := \quad & \frac{\partial(S_\alpha \phi \rho_\alpha)}{\partial t} - \text{div}\,(\rho_\alpha \lambda_\alpha \mathbf{K}\,(\mathbf{grad}\,p_\alpha - \rho_\alpha \mathbf{g})) \\ & - \rho_\alpha Q_\alpha = 0 \qquad \alpha \in \{w,n\} \end{aligned} \tag{2.20}$$

with the mobility $\lambda_\alpha = k_{r\alpha}/\mu_\alpha$.

This equation is completed by two supplementary equations:

1. The void space in the porous medium is completely filled by the different fluid phases:

$$S_w + S_n = 1 \qquad \Rightarrow \qquad \frac{\partial S_w}{\partial t} = \frac{\partial(1-S_n)}{\partial t} = -\frac{\partial S_n}{\partial t} \tag{2.21}$$

2. The capillary pressure p_c represents the difference between the two fluid pressures at every point:

$$p_c \;=\; p_n - p_w \qquad \Rightarrow \qquad \mathbf{grad}\, p_c \;=\; \mathbf{grad}\, p_n - \mathbf{grad}\, p_w \tag{2.22}$$

With the use of these relations, two of the four unknowns in Eq. (2.20) (p_w, p_n, S_w, S_n) can be eliminated. This results either in a pressure - pressure, saturation - saturation or pressure - saturation formulation (*Helmig* [31]).

At first, we take a look at the **saturation formulation** (e.g. *Aziz and Settari* [1]). Equation (2.18) can be formulated for each phase as follows:

wetting phase (e.g. water) w

$$L(S_w, \mathbf{v_w}) \;:=\; \phi \frac{\partial(\rho_w S_w)}{\partial t} + \operatorname{div}\{\rho_w \mathbf{v}_w\} - \rho_w Q_w \;=\; 0 \tag{2.23}$$

non-wetting phase (e.g. oil) n

$$L(S_n, \mathbf{v_n}) \;:=\; \phi \frac{\partial(\rho_n S_n)}{\partial t} + \operatorname{div}\{\rho_n \mathbf{v}_n\} - \rho_n Q_n \;=\; 0 \tag{2.24}$$

If we assume that both phases are incompressible (division by ρ_α is possible), apply Eq. (2.21) and add Eqs. (2.23) and (2.24), we obtain

$$\phi \left[\frac{\partial S_w}{\partial t} + \frac{\partial(1 - S_w)}{\partial t} \right] + \operatorname{div} \mathbf{v}_w + \operatorname{div} \mathbf{v}_n - Q_w - Q_n \;=\; 0 \tag{2.25}$$

After introducing the **total velocity** $\mathbf{v}_t = \mathbf{v}_w + \mathbf{v}_n$ and simplifying the time-derivative term, we get the following relationship:

$$\operatorname{div} \mathbf{v}_t \;=\; \operatorname{div}(\mathbf{v}_w + \mathbf{v}_n) \;=\; Q_w + Q_n \;=\; Q_t \tag{2.26}$$

Next, we write *Darcy's* law for each phase:

$$\begin{aligned} \mathbf{v}_w &= -\lambda_w \mathbf{K} \left(\mathbf{grad}\, p_w - \rho_w \mathbf{g} \right) \\ \mathbf{v}_n &= -\lambda_n \mathbf{K} \left(\mathbf{grad}\, p_w + \mathbf{grad}\, p_c - \rho_n \mathbf{g} \right) \end{aligned} \tag{2.27}$$

If the equation for $\mathbf{v}_n$ is multiplied by $\frac{-\lambda_w}{\lambda_n}$ and then subtracted from the equation for $\mathbf{v}_w$, we get after some rearranging

$$\mathbf{v}_w = \frac{\lambda_w}{\lambda_n} \mathbf{v}_n + \lambda_w \mathbf{K} \left(\mathbf{grad}\, p_c + \rho_w \mathbf{g} - \rho_n \mathbf{g} \right) \tag{2.28}$$

and with $\mathbf{v}_n = \mathbf{v}_t - \mathbf{v}_w$

$$\mathbf{v}_n = \frac{1}{1+\frac{\lambda_w}{\lambda_n}} \left[\mathbf{v}_t - \lambda_w \mathbf{K} \left(\mathbf{grad}\, p_c + \rho_w \mathbf{g} - \rho_n \mathbf{g}\right)\right] \tag{2.29}$$

Now we introduce the **fractional flow functions**

$$f_n = \frac{\lambda_n}{\lambda_w + \lambda_n} = \frac{\frac{k_{rn}(S_n)}{\mu_n}}{\frac{k_{rw}(S_w)}{\mu_w} + \frac{k_{rn}(S_n)}{\mu_n}} \quad ; \quad f_w = \frac{\lambda_w}{\lambda_w + \lambda_n} = \frac{\frac{k_{rw}(S_w)}{\mu_w}}{\frac{k_{rw}(S_w)}{\mu_w} + \frac{k_{rn}(S_n)}{\mu_n}} \tag{2.30}$$

and the mean mobility

$$\overline{\lambda} = \frac{\lambda_w \lambda_n}{\lambda_w + \lambda_n}$$

so the resulting equation is

$$\mathbf{v}_n \;=\; f_n \mathbf{v}_t - \overline{\lambda}\mathbf{K}(\mathbf{grad}\, p_c + \rho_w \mathbf{g} - \rho_n \mathbf{g}) \tag{2.31}$$

If we now insert Eq. (2.31) into Eq. (2.24), we get

$$-\phi \frac{\partial S_w}{\partial t} + \operatorname{div}\left[f_n \mathbf{v}_t - \overline{\lambda}\mathbf{K}(\mathbf{grad}\, p_c + \rho_w \mathbf{g} - \rho_n \mathbf{g})\right] - Q_n \;=\; 0 \tag{2.32}$$

Next, the following relationships are introduced:

$$\begin{aligned}
\mathbf{grad}\, p_c &= \frac{d\,p_c}{d\,S_w}\,\mathbf{grad}\, S_w \\
\operatorname{div}(f_n \mathbf{v}_t) &= \mathbf{v}_t\,\mathbf{grad}\, f_n + f_n \operatorname{div} \mathbf{v}_t \\
&= \mathbf{v}_t \frac{d\,f_n}{d\,S_w}\,\mathbf{grad}\, S_w + f_n\, Q_t \\
\operatorname{div}\left[\overline{\lambda}\mathbf{K}(\rho_w \mathbf{g} - \rho_n \mathbf{g})\right] &= \mathbf{K}(\rho_w \mathbf{g} - \rho_n \mathbf{g}) \cdot \mathbf{grad}\, \overline{\lambda} \\
&= \mathbf{K}(\rho_w \mathbf{g} - \rho_n \mathbf{g}) \cdot \frac{d\overline{\lambda}}{d\,S_w}\,\mathbf{grad}\, S_w \\
f_n + f_w &= 1 \\
\frac{d\,f_n}{d\,S_w} &= -\frac{d\,f_w}{d\,S_w} \\
-Q_n + f_n\, Q_t &= Q_w - f_w\, Q_t
\end{aligned} \tag{2.33}$$

We have to remember that the fluids were assumed to be incompressible. Also, it is assumed that $\mathbf{g}$ is not a function of space.

If we now insert these relationships in Eq. (2.32), we finally obtain the **saturation formulation** for two-phase flow:

$$
\begin{aligned}
L(S_w) \quad := \quad & -\underbrace{\phi\frac{\partial S_w}{\partial t}}_{\text{term 1}} \\
& -\underbrace{\left[\mathbf{v}_t\frac{d\,f_w}{d\,S_w}+\mathbf{K}\left(\rho_w\mathbf{g}-\rho_n\mathbf{g}\right)\frac{d\bar{\lambda}}{d\,S_w}\right]\cdot\mathbf{grad}\,S_w}_{\text{term 2}} \\
& -\underbrace{\operatorname{div}\left[\bar{\lambda}\mathbf{K}\frac{d\,p_c}{d\,S_w}\mathbf{grad}\,S_w\right]}_{\text{term 3}} \\
& +\underbrace{(Q_w-f_w\,Q_t)}_{\text{term 4}}=0
\end{aligned}
\qquad (2.34)
$$

In order to solve this equation, $\mathbf{v}_t$ has to be calculated beforehand, which is by no means trivial for 2D.

As the capillary pressure and the relative permeability show a highly non-linear dependence on the saturation, Eq. (2.34) represents a strongly coupled non-linear partial differential equation, which is of a mixed parabolic-hyperbolic type.

With the help of the four terms identified in Eq. (2.34) - an accumulation or storage part (term 1), an advective or convective part (term 2), a diffusive part (term 3), and a source and sink part (term 4) - the mixed character of the two-phase flow will be further explained:

For low water saturations ($S_w \approx S_{wr}$), the absolute value of the derivative of the capillary pressure $\frac{d\,p_c}{d\,S_w}$ becomes significantly large (see Figure 2.5) and therefore term 3 dominates Eq. (2.34). In this case, the differential equation is formally parabolic.

If, however, the water saturation is very large ($S_w \approx 1 - S_{nr}$), $\frac{d\,p_c}{d\,S_w}$ tends to zero and term 2 dominates the equation. In this case, the character of Eq. (2.34) becomes hyperbolic.

Since in a problem domain the water saturation is usually not constant but ranges between the aforementioned extremes, both parabolic and hyperbolic properties are present at the same time and have to be accounted for by the numerical scheme.

The approximation of the advection (hyperbolic) term is the main problem for discretizing the two-phase flow equations. In order to explain the problems that arise here, we have a look at a special case of the two-phase flow: the *Buckley-Leverett* problem, which is also described in Section 3.1.1. Here, capillary effects are neglected (term 3 = 0) and sources and sinks are omitted. For a quasi-1D system, this leads to the following equation (*Buckley and Leverett* [15]):

$$
\phi\frac{\partial S_w}{\partial t}+\mathbf{v}_t\underbrace{\frac{d\,f_w}{d\,S_w}}_{f_w{}'}\underbrace{\frac{\partial S_w}{\partial x}}_{\mathbf{grad}\,S_w}=0 \qquad (2.35)
$$

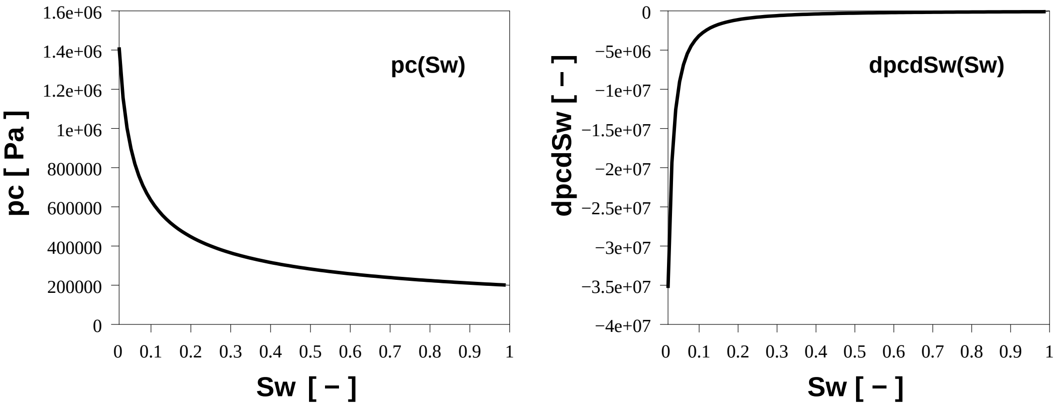


Figure 2.5: Example of a capillary pressure curve after *Brooks-Corey* and its derivative with respect to S_w ($S_{wr} = 0$)

In this case, the water saturation will move at a rate proportional to the derivative of the fractional flow function ($f_w{}'$) and the shape of the saturation profile becomes similar to that of $f_w{}'$ (*Dullien* [21]).

As can be seen in Figure 2.6, the shape of $f_w{}'$ is non-convex for a non-linear relative permeability function. Therefore, more than one value of the saturation can be associated with a certain value of f_w. As a front shape like this would be physically meaningless, the correct saturation profile can be found by satisfying two conditions (*Lake* [43], *Helmig* [31]):

1. *Rankine-Hugenoit* **jump condition:**
   With the help of the law of conservation of mass, a sharp front or shock is constructed, which results in a discontinuity of the saturation (*Dullien* [21], *LeVeque* [47]).

   Graphically, the front is constructed by drawing a tangent to the fractional flow curve from the point of the initial saturation in the domain (S_{wr}) (see Figure 2.7). At the point of $S_{w\_}t$ (saturation at the tangent point), the area of the fields A and B in the left picture

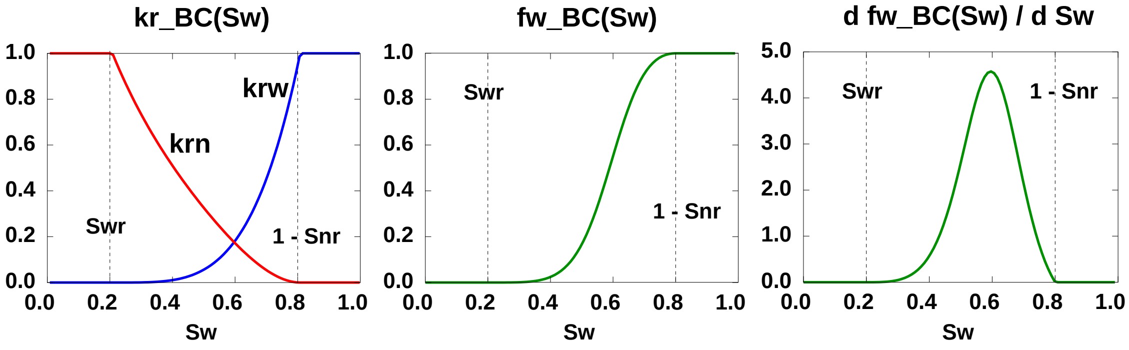


Figure 2.6: $k_{rw}(S_w)$, $k_{rn}(S_w)$, $f_w(S_w)$, and $f_w{}'(S_w)$ for the *Brooks-Corey* relationships

are the same: mass is distributed equally. In this figure, also $S_{w\_}i$ is pictured, which represents the saturation at the inflection point.

The front velocity has to fulfil the following condition (*Helmig* [31]):

$$\tilde{\mathbf{v}}_{\Delta S_w} = \frac{f_w(S_w^u) - f_w(S_w^d)}{S_w^u - S_w^d} = \frac{\Delta f_w}{\Delta S_w} \tag{2.36}$$

Here, S_w^u represents the upstream and S_w^d the downstream water saturation of the shock. This means that the velocity at which the front travels is proportional to the slope of the tangent. Saturations which are larger than $S_{w\_}t$ (see Figure 2.7) travel at a velocity that is proportional to the tangent at these saturations.

Equation (2.36) states that the specific front velocity has to correspond to the physical process (jump in the saturations, relative permeability - saturation relationship) in order to satisfy the continuity condition at the jump. This condition, however, does not guarantee that the *form* of the solution corresponds to the physical solution. This is checked by the next condition:

2. **Entropy condition:**
   The saturation jump has to satisfy (*LeVeque* [47]):

$$\frac{f_w(S_w) - f_w(S_w^u)}{S_w - S_w^u} \geq \mathbf{v}^{shock} \geq \frac{f_w(S_w) - f_w(S_w^d)}{S_w - S_w^d} \tag{2.37}$$

   This is valid for all $S_w(x,t)$ between S_w^u and S_w^d. Interpreted physically, the upstream velocity $\mathbf{v}_u$ of the shock always has to be larger than the downstream velocity $\mathbf{v}_d$. Or in other words, the formulation for the shock front must be self-sharpening.

   If capillary pressure effects dominate, the entropy condition is fulfilled (*Helmig* [30]). If, however, term 3 in Eq. (2.34) tends to zero, the discritization method has to introduce enough numerical diffusion in order to continue to satisfy the entropy condition.

The two above-mentioned conditions are explained using the example of a special case; they are, however, also valid for general two-phase flow.

For most practical applications, the saturation formulation is not feasible due to the a priori calculation of the total velocity and the constriction to incompressible fluids alone. Therefore, as it is also the case in this work, the **pressure-saturation formulation** is applied.

For the unknowns water pressure p_w and oil saturation S_n, the resulting equations look as follows :

wetting phase (e.g. water) w

$$\begin{aligned} L_w(p_w, S_n) \;:=\; -\phi\,\frac{\partial(S_n\,\rho_w)}{\partial t} \quad &- \quad \operatorname{div}\left(\rho_w\,\lambda_w\,\mathbf{K}\left(\mathbf{grad}\,p_w \;-\; \rho_w\,\mathbf{g}\right)\right) \\ &- \quad \rho_w\,Q_w \;=\; 0 \end{aligned} \tag{2.38}$$

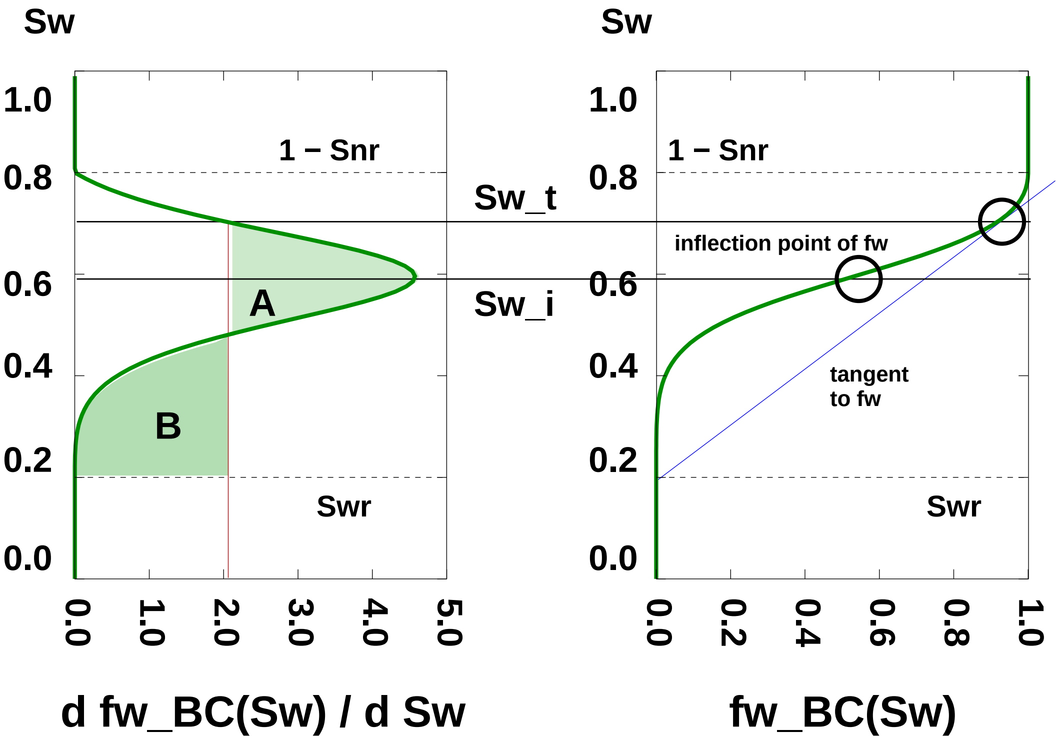


Figure 2.7: Construction of the sharp front

non-wetting phase (e.g. oil) n

$$L_n(p_w, S_n) := \phi \frac{\partial(S_n \rho_n)}{\partial t} - \operatorname{div}\left(\rho_n \lambda_n \mathbf{K}\left(\mathbf{grad}\, p_c + \mathbf{grad}\, p_w - \rho_n \mathbf{g}\right)\right) - \rho_n Q_n = 0 \tag{2.39}$$

or in a more general way:

$$L_\alpha(p_w, S_n) := (-1)^{\delta_{\alpha w}} \phi \frac{\partial(S_n \rho_\alpha)}{\partial t} - \operatorname{div}\left(\rho_\alpha \lambda_\alpha \mathbf{K}\left(\mathbf{grad}\, p_w + \delta_{\alpha n}\mathbf{grad}\, p_c - \rho_\alpha \mathbf{g}\right)\right) - \rho_\alpha Q_\alpha = 0 \qquad \alpha \in \{w, n\} \tag{2.40}$$

The porosity ϕ is assumed to be constant over time.

In this formulation, the advective (hyperbolic) and diffusive (parabolic) properties of the two-phase system cannot be explicitly distinguished. This makes the numerical treatment of these equations even more challenging, as both effects need a special treatment.

In order to identify regions where one of these properties dominates and therefore to improve the numerical scheme, a criterion can be applied which describes the interaction between advection and diffusion in the system: the *Peclet* number. This will be further explained in Section 7.1.

2.4 Discretization

For the partial differential equations describing two-phase flow in porous media which were introduced in the previous section, analytical solutions only exist in a few special cases. Therefore, numerical methods are applied, which, by means of a discretization scheme, approximate the partial differential equations by algebraic equations. As each of these approximations covers only a small domain in space and/or time, the numerical solution is then obtained at discrete points in space and time (*Ferziger and Perić* [24]).

In order to represent a suitable discretization scheme, the chosen method must meet certain requirements. Generally, it has to be consistent (for a refinement of the discretization length (either space or time), the difference between the numerical solution and the analytically exact solution of the differential equation tends to zero), stable (small errors in the solution which occur during the numerical simulation, are not amplified), and convergent (for a decrease in the grid spacing, the numerical solution tends towards the exact solution of the differential equation). Furthermore, the discretization method should embrace the physical character of the differential equations: since the two-phase flow equations introduced here are based on conservation laws, the numerical scheme should reflect these laws by being e.g. mass conservative.

For the discretization of flow equations, different approaches exist. The basic techniques are the finite difference method, the finite element method, and the finite volume method. An overview of different discretization schemes and their properties can be found, for example, in *Huyakorn and Pinder* [36], *Forsyth* [25], *Helmig* [31], and *Hinkelmann* [33].

In the following sections, the numerical methods utilized in this work are introduced. Since transient problems are considered, a discretization for space as well as time has to be provided. In Section 2.4.1, the applied time discretization method will be described. Afterwards, in Section 2.4.2, the chosen space discretization schemes will be explained. In this thesis, a node-centered finite volume method which is based on a Galerkin finite element method, the so-called Box Method, will be applied. It has the advantage that it can be employed for unstructured grids. Furthermore, the method is locally (for each finite volume) mass conservative. In this section, a discretization technique, which pays special attention to the handling of interfaces between differently permeable soils, is also explained. The last section of this chapter is devoted to the solution process of the obtained algebraic equations.

The discretization techniques described here will be applied in the further course of this work to homogeneous as well as heterogeneous test problems. As for two-phase flow problems, special attention has to be paid on the one hand to the treatment of the strong nonlinear advective terms, on the other hand diffusion dominated processes also have to be accounted for correctly; the Box Method described here will be utilized for both extremes.

The convergence behavior of this method (improvement of the numerical solution for decreasing element sizes) serves later on as a basis for the adaptive methods applied in this work. Also, it will be shown that the different extremes of the two-phase flow equations each require a special treatment concerning the discretization. Therefore, a special procedure for the Box Method explained here will be introduced in Chapter 7.

2.4.1 Discretization in time

When unsteady flows are computed, as it is the case for the two-phase flow equations introduced in the previous section, a time discretization needs to be applied.

The particular distinctiveness of time lies in the direction of influence: interferences imposed on the flow at a given instant will affect the flow only in the future - there is no backward influence. This implies that (apart from boundary conditions) it is not possible to apply conditions to the solution at any time after the start of the calculation.

Generally, two different methods of time discretization can be distinguished:

On the one hand, time can be considered as 'just another dimension'. This approach leads to, for example, space-time finite element methods and space-time Discontinuous Galerkin methods.

On the other hand, a semi-discrete method can be applied. Here, the time-derivative term in the equations is approximated by either a one-step or a multi-step scheme. While the first one only consideres values of the solution for t^n for the computation of the current point in time t^{n+1}, the latter additionally embraces solutions at preceding time levels t^{n-k} $(k \geq 0)$. For multi-step schemes the data as well as the mesh needs to be stored for all time levels considered. This leads to an increase in transfer operations which, especially when space-adaptive methods are applied, is not unproblematic.

For the time discretization in this work, the implicit *Euler* scheme (finite difference method of the first order), a one-step scheme, is applied. Here, the time-dependent partial derivative of an unknown u is approximated by:

$$\frac{\partial u}{\partial t} = \frac{u^{n+1} - u^n}{\Delta t} = f(u^{n+1}) \qquad \text{with} \qquad \Delta t = t^{n+1} - t^n \tag{2.41}$$

The indices $n+1$ and n denote the point in time at which u is evaluated. As the right-hand side is formulated for the new time level, a system of equations with all degrees of freedom has to be solved for each time step. For the time dependent terms in Eqs. (2.38) and (2.39), the discretization with the implicit *Euler* scheme yields:

$$\underline{\text{wetting phase } (w)} \qquad\qquad \underline{\text{non-wetting phase } (n)}$$

$$-\{[S_n\,\rho_w]_j^{n+1} - [S_n\,\rho_w]_j^n\}\frac{\phi}{\Delta t} \qquad\qquad \{[S_n\,\rho_n]_j^{n+1} - [S_n\,\rho_n]_j^n\}\frac{\phi}{\Delta t} \tag{2.42}$$

The implicit *Euler* scheme is unconditionally stable for arbitrary time steps and very suitable also for complex problems (e.g. *Helmig* [31], *Helmig and Bastian* [27]). Yet it has to be remarked that an investigation of the *Taylor* series expansion shows that due to the time discretization numerical diffusion is introduced in the system (*Helmig* [30]). However, efforts to reduce this effect are made by the application of an automatic time step control, which is implemented in MUFTE-UG (see Section 2.4.3).

2.4.2 Discretization in space

2.4.2.1 Subdomain collocation finite volume method (Box Method)

For the spacial discretization, the Box Method (subdomain collocation finite volume method) is used (*Bastian* [7], *Helmig* [27]). First, the model domain G is discretized by a finite element (FE) mesh consisting of a set of nodes K_i and a set of adjoining elements e_i (see Figure 2.8). The boundary ∂G of the domain is composed of parts which have a Dirichlet boundary condition ($\Gamma_{\alpha,D}$) and parts holding a Neumann boundary condition ($\Gamma_{\alpha,N}$), so that: $\partial G = \Gamma_{\alpha,D} \cup \Gamma_{\alpha,N}$ with $\alpha \in \{w, n\}$.

Then, a secondary finite volume (FV) mesh is constructed in the following way: for each node K_i, a control volume or box B_i is constructed by connecting the midpoints of the adjoining element sides and the barycenters of the neighboring elements. Each control volume B_i consists of k subcontrol volumes b_i^k, which describe the intersection of B_i with the elements e_k connected to node K_i (see Figure 2.8). The boundary of the box is described by ∂B_i, the integration path along this boundary is given by Γ_{B_i}. The right part of Figure 2.8 shows the representation for an element. Here, the different subcontrol volumes (scv's) inside an element are pictured. In addition, the subcontrol volume faces (scvf's), dividing the individual subcontrol volumes and connecting the edge midpoints with the element barycenter, are shown. Furthermore, the integration point (IP) $x_{i\,j}^k$ located in the middle of a subcontrol volume face between the subcontrol volumes b_i^k and b_j^k is explained. And last but not least, the outer normal vector $n_{i\,j}^k$ for a subcontrol volume face is indicated.

Another thing that can be observed in Figure 2.8 is that two subdomains of G, which have different properties, meet along the boundary of control volumes as indicated by Γ. As a consequence, different properties can exist in one element, but a control volume always has one specific property.

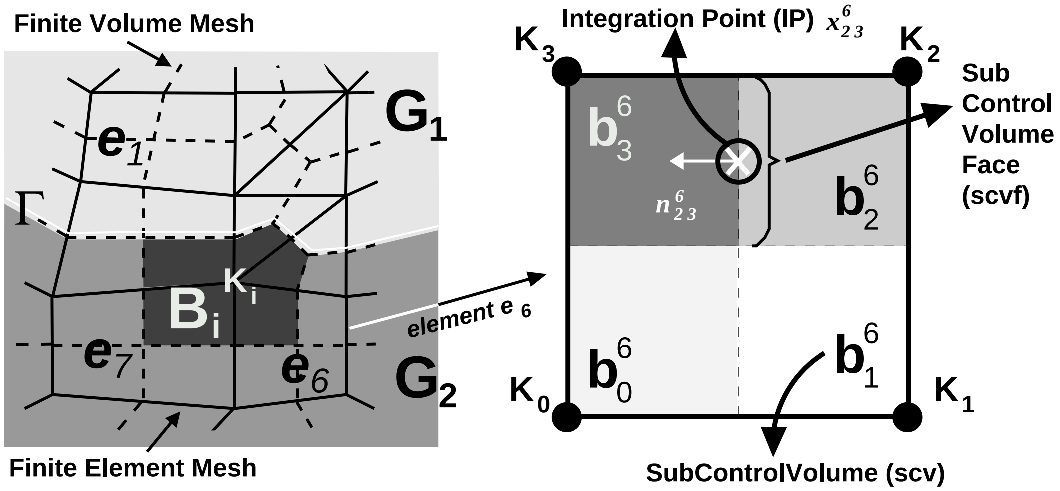


Figure 2.8: Overlap of the FE and FV mesh (left), zoom into element e_6 with corresponding subcontrol volumes (right) (after *Jakobs et al.* [37])

Just like the finite element method, the Box Method is based on the principle of weighted residuals:

The function of an unknown $f(u)$ in the whole domain is approximated ($f(\tilde{u})$) by the discrete values at the nodes of the discretization mesh ($\hat{u}_j$) and linear basis functions (N_j). For the pressure - saturation formulation, this yields:

$$\begin{aligned} \tilde{S}_\alpha &= \sum_j \hat{S}_{\alpha j} \cdot N_j & & \\ \tilde{p}_\alpha &= \sum_j \hat{p}_{\alpha j} \cdot N_j & \mathbf{grad}\,\tilde{p}_\alpha &= \sum_{j \in \eta_i} (\hat{p}_{\alpha j} - \hat{p}_{\alpha i})\,\mathbf{grad}\,N_j \\ \tilde{\mathbf{g}} &= -\mathrm{g} \sum_j \hat{z}_j \cdot N_j & \mathbf{grad}\,z &= \sum_{j \in \eta_i} (\hat{z}_j - \hat{z}_i)\,\mathbf{grad}\,N_j \end{aligned} \tag{2.43}$$

where g is the gravitational acceleration, $\hat{z}_j$ is the z-coordinate for node j and η_i is the set of all neighboring nodes of node i. The basis functions N_j are common C^0-*Lagrange* polynomials with the value 1 for node j and 0 for all other nodes.

If the approximations of Eq. (2.43) are inserted into the partial differential equations (2.40), they are no longer exact fulfilled, but lead to a residuum ε. Application of the principle of orthogonality leads to the residuum being weighted with weighting functions W_i and vanishing in the whole domain G:

$$\int_G W_i \cdot \varepsilon \, dG \overset{!}{=} 0 \qquad \text{with} \qquad \sum_i W_i = 1 \qquad \text{in } G \tag{2.44}$$

Generally speaking, each of the weighted two-phase flow equations can be written as:

$$\int_G \underbrace{W_i \frac{\partial e}{\partial t}}_{\mathbf{I}} dG + \int_G \underbrace{W_i \operatorname{div} \mathbf{F}}_{\mathbf{II}} dG - \int_G \underbrace{W_i \, r}_{\mathbf{III}} dG \overset{!}{=} 0 \tag{2.45}$$

where

			for the $p_w - S_n$-formulation:
e	:	scalar entity	$e_\alpha = S_n \phi \rho_\alpha$
$\mathbf{F}$	:	advective flow and diffusion	$\mathbf{F}_\alpha = \rho_\alpha \mathbf{v}_\alpha$
r	:	sources and sinks	$r_\alpha = \rho_\alpha Q_\alpha$

For the time-dependent **term I** of the pressure - saturation equations for the wetting (Eq. (2.38)) and non-wetting (Eq. (2.39)) phases and taking into account Eqs. (2.42) and (2.43), the following expression is obtained:

$$
\begin{aligned}
&(-1)^{\delta_{\alpha w}} \int\limits_G W_i\,\phi \frac{\partial(\tilde{S}_n\,\rho_\alpha)}{\partial t}\,dG \\
&= (-1)^{\delta_{\alpha w}} \int\limits_G W_i\,\phi\{[\tilde{S}_n\,\rho_\alpha]_j^{n+1} - [\tilde{S}_n\,\rho_\alpha]_j^n\}\frac{1}{\Delta t}\,dG \\
&= (-1)^{\delta_{\alpha w}} \int\limits_G W_i\,\phi\,\frac{1}{\Delta t} N_j\,\{[\hat{S}_n\,\rho_\alpha]_j^{n+1} - [\hat{S}_n\,\rho_\alpha]_j^n\}\,dG \\
&= (-1)^{\delta_{\alpha w}} \sum_{j\in\eta_i} M_{ij}\,\phi\,\frac{1}{\Delta t}\,\{[\hat{S}_n\,\rho_\alpha]_j^{n+1} - [\hat{S}_n\,\rho_\alpha]_j^n\} \qquad \alpha\in\{w,n\} \qquad (2.46)
\end{aligned}
$$

Here, η_i is the set of all neighboring nodes of node i and $\delta_{\alpha w}$ represents the *Kronecker* delta. For the mass matrix $M_{ij} = \int\limits_G W_i N_j\,dG$, a mass lumping is applied (*Helmig* [31], *Huber* [35]). The physical interpretation is a reduction of the storing capacities of the mesh on the nodes only. Mathematically, this is expressed by:

$$
M_{ij}^{lump} = \begin{cases} \int\limits_G W_i\,dG = \int\limits_G N_i\,dG = V_i & \text{for } i = j \\ 0 & \text{for } i \neq j \end{cases} \qquad (2.47)
$$

where V_i represents the volume for the box associated with node i. All coefficients of a row of the mass matrix are concentrated in the diagonal entry.

Analogously, the source-sink **term III** yields:

$$
\int\limits_G W_i\,(\rho_\alpha\,Q_\alpha)\,dG = V_i\,\left(\rho_{\alpha i}^{n+1}\,Q_{\alpha i}^{n+1}\right) \qquad (2.48)
$$

Q_α denotes the flow rate of a source or sink in the domain and is given in the units of [(m^3 / s) · (1 / m^3)] = [1/s] (flow rate per volume of the subcontrol volume).

For **term II**, at first the chain rule and then the *Green-Gauss*ian integral theorem is applied:

$$
\begin{aligned}
\int\limits_G W_i\,\mathrm{div}\,\mathbf{F}\,dG &= \int\limits_G \mathbf{grad}\,(W_i\cdot\mathbf{F})\,dG - \int\limits_G (\mathbf{grad}\,W_i)\cdot\mathbf{F}\,dG \\
&= \oint\limits_{\partial G} (W_i\cdot\mathbf{F})\cdot\mathbf{n}\,d\partial G - \int\limits_G (\mathbf{grad}\,W_i)\cdot\mathbf{F}\,dG \qquad (2.49)
\end{aligned}
$$

Here, ∂G represents the boundary of the domain G and $\mathbf{n}$ denotes the outer normal vector of ∂G.

For the Box Method, the weighting function is chosen as follows: W_i is chosen as piecewise constant over the control volume B_i with the following constraints:

$$W_i(x) = \begin{cases} 1 & \text{if}\, x \in B_i \\ 0 & \text{if}\, x \notin B_i \end{cases} \tag{2.50}$$

With this definition, $\mathbf{grad}\, W_i$ in Eq. (2.49) results in 0 and only the surface integral remains (the value of W_i on the border of B_i is 1).

If this is applied to Eq. (2.20) and the definitions of Eq. (2.43) are inserted, the following expression is obtained:

$$\begin{aligned}
& \int\limits_G W_i \operatorname{div} \mathbf{F}\, dG \\
= \; & -\int\limits_G W_i \operatorname{div} \left(\rho_\alpha \lambda_\alpha \mathbf{K} \left(\mathbf{grad}\, \tilde{p}_w + \delta_{\alpha n} \mathbf{grad}\, \tilde{p}_c - \rho_\alpha \tilde{\mathbf{g}}\right)\right) dG \\
= \; & -\int\limits_G W_i \operatorname{div} \Bigg(\rho_\alpha \lambda_\alpha \mathbf{K} \Bigg(\sum_{j \in \eta_i} (\hat{p}_{wj} - \hat{p}_{wi}) + \delta_{\alpha n} \sum_{j \in \eta_i} (\hat{p}_{cj} - \hat{p}_{ci}) \\
& \qquad - \rho_\alpha g \sum_{j \in \eta_i} (\hat{z}_j - \hat{z}_i) \Bigg) \mathbf{grad}\, N_j \Bigg) dG \\
= \; & -\int\limits_G W_i \operatorname{div} \left(\rho_\alpha \lambda_\alpha \mathbf{K} \sum_{j \in \eta_i} (\psi_{\alpha j} - \psi_{\alpha i}) \mathbf{grad}\, N_j \right) dG \\
= \; & -\oint\limits_{\partial B_i} W_i \left(\rho_{\alpha(ij)} \lambda^A_{\alpha(ij)} \mathbf{K} \sum_{j \in \eta_i} (\psi_{\alpha j} - \psi_{\alpha i}) \mathbf{grad}\, N_j \right) \cdot \mathbf{n}\, d\Gamma_{B_i} \\
= \; & -\sum_{l \in E_i} \sum_{j \in \eta_i} \rho^{n+1}_{\alpha(ij)} \lambda^{A,l}_{\alpha(ij)} \gamma^{BOX,l}_{ij} (\psi^{n+1}_{\alpha j} - \psi^{n+1}_{\alpha i}) - m_{\alpha i}
\end{aligned} \tag{2.51}$$

Here, E_i is the set of elements which are connected to node i.

$\psi_{\alpha i}$ represents the **total potential** of phase α at node i and is defined as:

$$\psi_{\alpha i} = \hat{p}_{wi} + \delta_{\alpha n} \hat{p}_{ci} - \rho_{\alpha i}\, g\, \hat{z}_i \tag{2.52}$$

where $\hat{z}_i$ is the geodetic height of node i. The direction of the (discrete) flow of phase α is therefore represented by $(\psi_{\alpha j} - \psi_{\alpha i})$.

γ^{BOX}_{ij} is the **transmissivity integral** with:

$$\gamma^{BOX}_{ij} = \oint\limits_{\partial B_i \setminus \Gamma_{\alpha,N}} \mathbf{K}\, \mathbf{grad}\, N_j \mathbf{n}\, d\Gamma_{B_i} \tag{2.53}$$

Note that the integral comprises the boundaries of all boxes B_i, but excludes the Neumann boundaries $\Gamma_{\alpha,N}$.

The flow over these boundaries ($\partial B_i \cap \Gamma_{\alpha,N}$) is considered by $m_{\alpha i}$ [kg / (m^2 s)].

The subscript (ij) indicates that this value is evaluated at a location halfway between the nodes i and j, where the box boundary intersects with the element edge. For the density, this results in an arithmetic weighting:

$$\rho_{\alpha,(ij)/2} = \frac{1}{2}\left(\rho_{\alpha i} + \rho_{\alpha j}\right) \tag{2.54}$$

For the mobility in Eq. (2.51), the superscript A states that λ can be included in different ways:

For the **central weighting method** (A = *cent*), the mobility is also weighted arithmetically along the element edge l_{ij}:

$$\lambda_{\alpha(ij)}^{cent} = \frac{1}{2}\left(\lambda_{\alpha i} + \lambda_{\alpha j}\right) \tag{2.55}$$

As already introduced in Section 2.3, the entropy condition always has to be fulfilled. If the capillary effects are not dominant enough, numerical oscillations and erroneous front approximations will occur and impair the solution. The standard procedure for avoiding this behavior is to introduce an optimal dose of artificial diffusion. This can be achieved with a wide range of schemes (for an overview, see *Helmig* [30], *Kröner* [41]). The method that was pursued in this work is the **fully upwind method**, where the mobility λ is considered constant in an element and holds the value of the mobility at the upstream node (see Figure 2.9).

This is evaluated by the following expression (A = *up*):

$$\lambda_{\alpha(ij)}^{up} = \begin{cases} i & \text{for} \quad (\psi_{\alpha_j} - \psi_{\alpha_i}) \leq 0 \\ j & \text{for} \quad (\psi_{\alpha_j} - \psi_{\alpha_i}) > 0 \end{cases} \tag{2.56}$$

The artificial diffusion results in a stabilization of the solution (*Lantz* [45], *Leonard* [46]). A side effect, however, is an anisotropic dispersion and a smearing of the saturation front (*Helmig* [30]). As the non-linear two-phase flow equations are self-sharpening, however, this smearing has a less dramatic effect on the solution than in the case of linear problems (*Aziz and Settari* [1]).

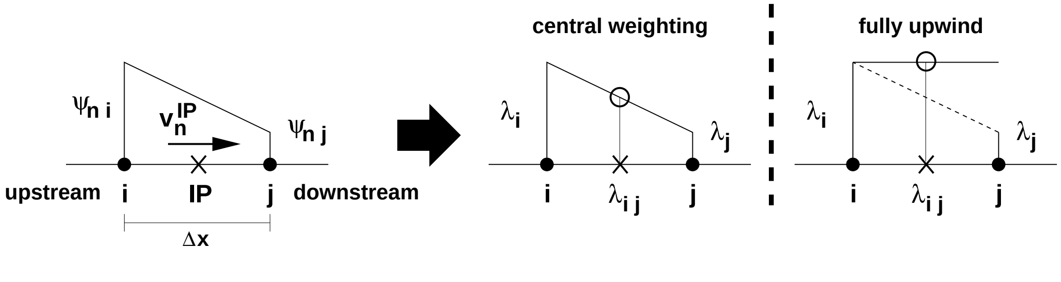


Figure 2.9: Different weighting of the mobility

In order to choose the degree of upwinding in the numerical code, the upwind parameter α_{up} is introduced as follows:

$$\lambda_{IP} \;=\; \alpha_{up} \cdot \lambda^{up} + (1-\alpha_{up}) \cdot \lambda^{cent} \qquad\qquad \text{with } 0.0 \leq \alpha_{up} \leq 1.0 \tag{2.57}$$

A value of $\alpha_{up} = 0.0$ therefore results in a centrally weighted scheme, whereas $\alpha_{up} = 1.0$ leads to the fully upwinding method.

Combining Eqs. (2.46), (2.51), and (2.48) yields the expression for the box-discretized $p_w - S_n$ two-phase flow equations:

$$\begin{aligned} g_{\alpha i}(p_{wi}^{n+1}, p_{wj}^{n+1}, S_{ni}^{n+1}, S_{ni}^{n}) \;=\;& (-1)^{\delta_{\alpha w}} \sum_{j \in \eta_i} M_{ij}\, \phi\, \frac{1}{\Delta t} \left\{ [\hat{S}_n \rho_\alpha]_j^{n+1} - [\hat{S}_n \rho_\alpha]_j^{n} \right\} \\ & + - \sum_{l \in E_i} \sum_{j \in \eta_i} \rho_{\alpha(ij)}^{n+1}\, \lambda_{\alpha(ij)}^{A,l}\, \gamma_{ij}^{BOX,l} (\psi_{\alpha j}^{n+1} - \psi_{\alpha i}^{n+1}) \\ & - V_i \left(\rho_{\alpha i}^{n+1}\, Q_{\alpha i}^{n+1} \right) \\ & - m_{\alpha i} \end{aligned} \tag{2.58}$$

In the code of MUFTE-UG (*Bastian et al.* [5], *Bastian* [6], *Helmig et al.* [29]), the Box Method - which is a combination of a finite element and finite volume method - is realized in such a way that the entries in the global stiffness matrix are computed according to the finite element formulation. This means that, inside a loop over all elements, all line integrals within a single element are computed. These line integrals are made up of segments of control volume boundaries, which are approximated by the midpoint rule: the value at the integration point $x_{(ij)}$, which is the midpoint of the subcontrol volume face between the nodes i and j (see Figure 2.8), is multiplied by the length of the corresponding subcontrol volume face.

2.4.2.2 Approximation of heterogeneities

In order to satisfy the momentum equation at the interface between two subdomains with different material properties, the capillary pressure has to be continuous over the interface (*Bear* [10]). This condition is called the **capillary equilibrium condition** (*van Dujin et al.* [20]). It implies that the saturation at the interface experiences a jump, as explained in Figure 2.10: For the same value of $p_{c\,\text{interface}}$, the saturation of water is smaller in the coarse domain (S_w^1) than in the fine-sand lense (S_w^2).

For a fully water-saturated medium, the capillary equilibrium condition is violated when a *Brooks-Corey* capillary pressure function is applied. In this case, the capillary pressure takes the value of the entry pressure p_d, which leads to $p_c|G_1(S_w = 1) = p_d|G_1 < p_d|G_2 = p_c|G_2(S_w = 1)$ (see Figure 2.10). In this figure, also S_w^* is shown. It represents the threshold saturation: here, the capillary pressure of the coarse material is equal to the entry pressure of the fine material.

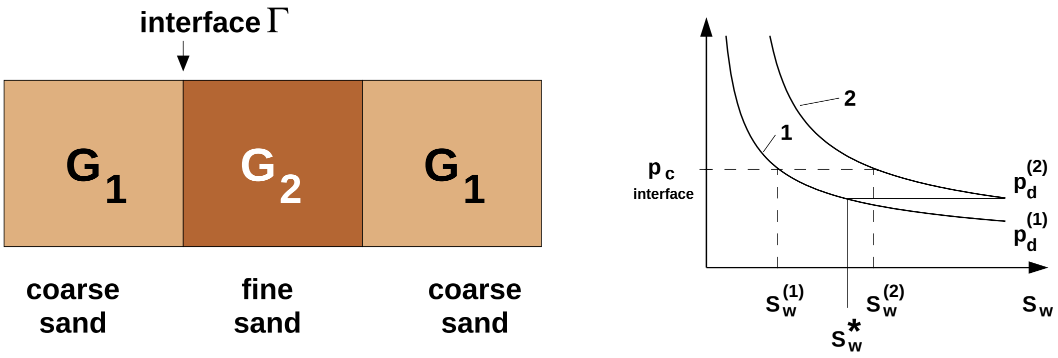


Figure 2.10: Jump of the saturation at the interface of two different media due to the continuity in the capillary pressure

If the central weighted box scheme is used for heterogeneities, one runs into the problem described in Figure 2.11 at $t = t_i$. Due to the higher capillary pressure in the fine-sand domain (comprising the nodes K_3 and K_4 in this example), a pressure gradient for the non-wetting phase points in the opposite direction of the flow. As, however, S_n and therefore the mobility λ_n are zero, no flow is possible. When the non-wetting phase approaches the interface at $t = t_i + 1$, $S_n|K_2$ and with it $\lambda_n|K_2$ become larger than 0. As the mobility is interpolated linearly between the two nodes on each side of the interface (here: nodes K_2 and K_3), λ_n is larger than 0 at the integration point on the interface. This means that a flow of the non-wetting phase is possible. As the pressure gradient points from the fine-sand lense to the coarse domain (here: from right to left), the non-wetting phase flows out of the lense, which leads to negative saturations. This is clearly a highly non-physical effect.

The fully upwinding scheme ensures a correct reproduction of the physical flow situation (*Helmig and Huber* [28]). In the case of our example, node K_3 is the upstream node for the integration point at the interface. The corresponding mobility is zero - again, no flow is possible. However, as the non-wetting phase enters the lense, the convergence of the numerical solver deteriorates noticeably. This is due to the fact that the upwind vertex oscillates between the nodes K_2 and K_3, depending on the current pressure state.

Another problem one runs into when applying the Box Method is depicted in Figure 2.12. Here, the distribution of p_w and p_n across an interface of two subdomains with different permeabilities is shown (left: coarse material, high permeability, low entry pressure; right: fine material, low permeability, high entry pressure). The state is shown for a fully water-saturated domain.

In order for the non-wetting phase to infiltrate into the fine sand domain, the corresponding entry pressure, or rather the difference between the entry pressures of the two domains p_d^{Δ}, has to be overcome. The analytical case is indicated by the number 1. Here, the interface is evaluated exactly at the borderline of the two domains. The corresponding saturation of the non-wetting phase which has to be reached represents $1 - S_w^*$, as explained in Figure 2.10.

Numbers 2 to 4 in Figure 2.12 represent different discretization lengths, where the symbols indicate the respective node location. Note that the interface is always located between

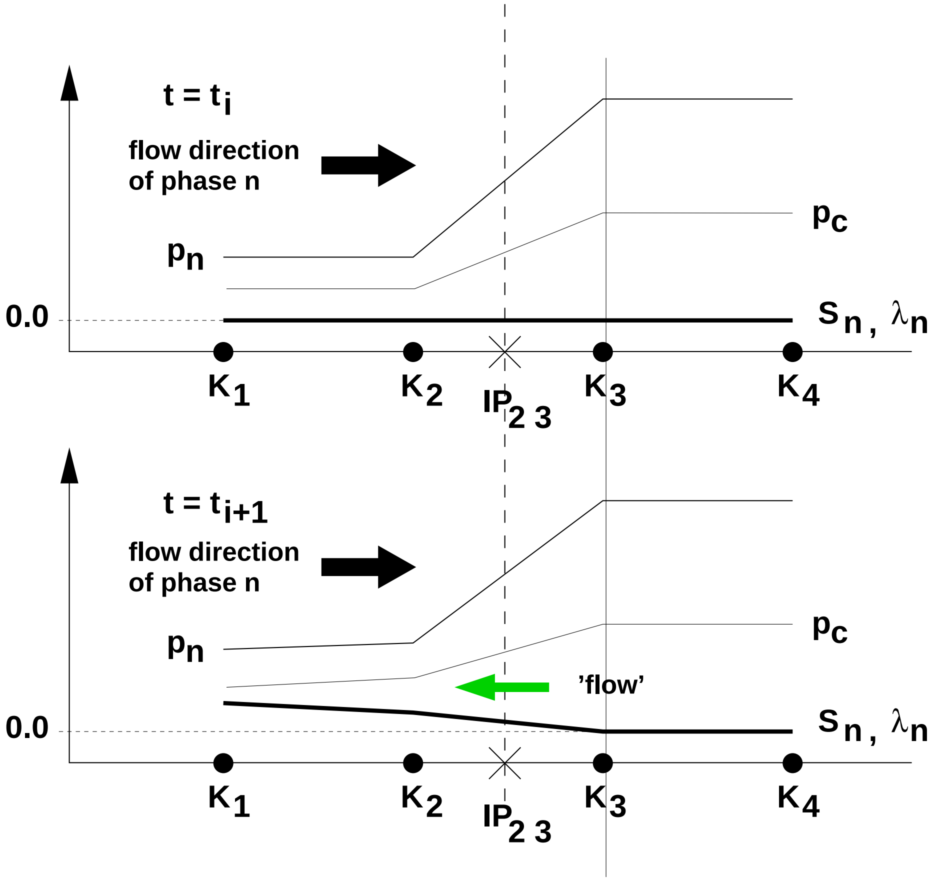


Figure 2.11: S_n, p_n, p_c, and λ_n for a centrally weighted method at an interface between different materials

two nodes, as explained in Figure 2.8. Due to the discretization, the absolute value of the first pressure of the non-wetting phase encountered in the fine-sand domain decreases for increasing element lengths. For the same reason, the absolute value of the pressure for the last node in front of the interface increases.

As can be seen by comparing the lengths of the lines representing the pressure differences for cases 1 through 4, the pressure which has to be overcome for infiltration decreases. In the case of the discretization represented by number 4, the pressure in the coarse domain is already larger than the pressure in the fine domain: Here, the non-wetting phase can enter immediately.

In order to avoid this dependence on the discretization length and to ensure that the non-wetting phase can only infiltrate when the entry pressure and S_w^* has been reached, another discretization scheme can be applied, which is described in the next section.

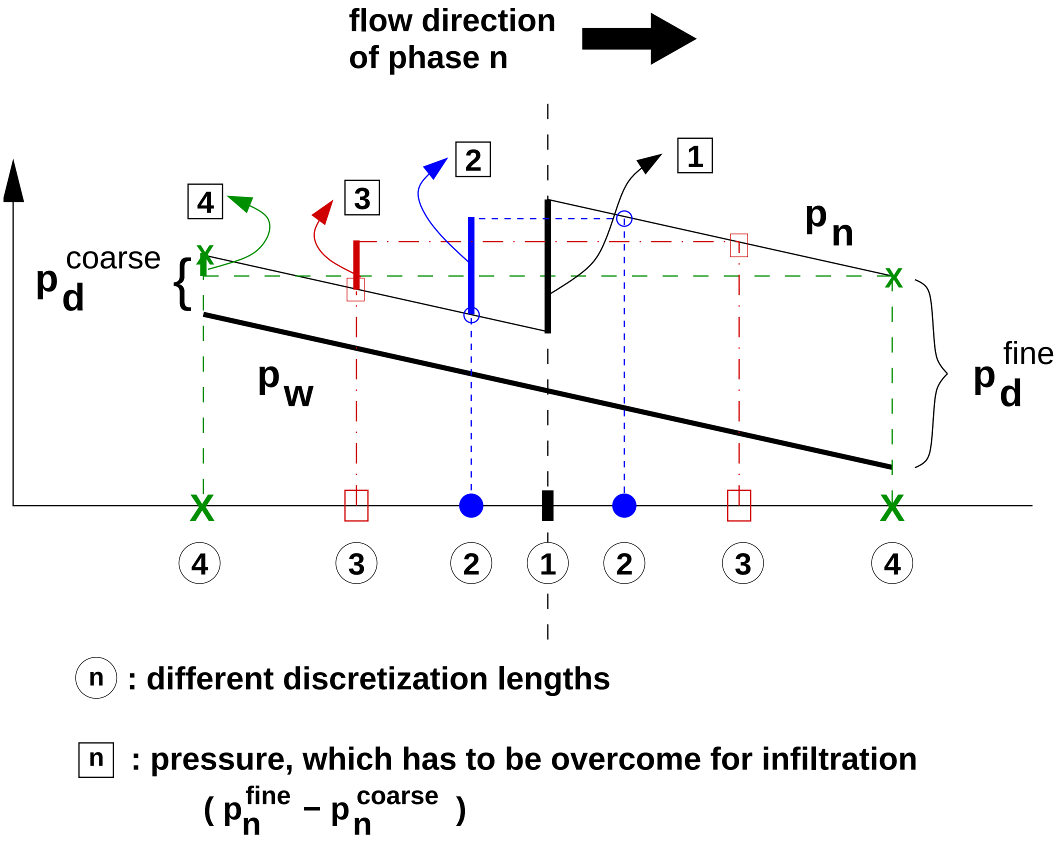

flow direction
of phase n
1
2
3
4
p_n
p_d^{coarse}
p_w
p_d^{fine}
4
3
2
1
2
3
4
n : different discretization lengths
n : pressure, which has to be overcome for infiltration
$(p_n^{fine} - p_n^{coarse})$

Figure 2.12: Explanation of different entry pressures for different discretization lengths for the Box Method

2.4.2.3 The PPSIC Method (Box-Trans Method)

The scheme presented in this section is called **phase pressure saturation interface condition** (PPSIC) or **transition condition** (trans-condition) (*Helmig et al.* [32], *Bastian et al.* [8]). Here, the transition of the saturations at the interface is computed by regarding the capillary pressure p_c at the interface as a quasi-primary variable.

At first, the **threshold saturation** S_w^* is defined according to Figure 2.10. It represents the value of the saturation, where the capillary pressure of the coarse material is equal to the entry pressure of the fine material: $p_c^{G_1}(S_w^*) = p_d^{G_2}$. Then, the saturation at the interface is calculated using the **extended capillary pressure condition** (*van Duijn et al.* [20]):

$$S_w|G_2^\Gamma = \begin{cases} 1.0 & \text{if } S_w|G_1^\Gamma \geq S_w^* \\ \mathrm{Inv}_{pc}^{G_2} & \text{if } S_w|G_1^\Gamma < S_w^* \end{cases} \tag{2.59}$$

with $\mathrm{Inv}_{p_c}^{G_2}$ as the inverse of the capillary pressure saturation function of subdomain G_2.

In order to integrate the interface condition into the Box Method, the evaluation of the parameters has to be changed from patch-wise to element-wise. In an element-oriented approach, the interface between two subdomains runs along the element edges and not along the subcontrol volume faces (see Figure 2.13, compare Figure 2.8).

The node K_i exists only once and only one saturation ($S_{n,i}$) is stored for it. The node can have, however, two (or more) saturation values associated with it, depending on the location of the element currently evaluated (here: $S_n|K_i^{G_1}$ and $S_n|K_i^{G_2}$).

The calculation of the saturation values is done in the following way: At first, the **minimal entry pressure** $p_{c,min}^i$ is calculated.

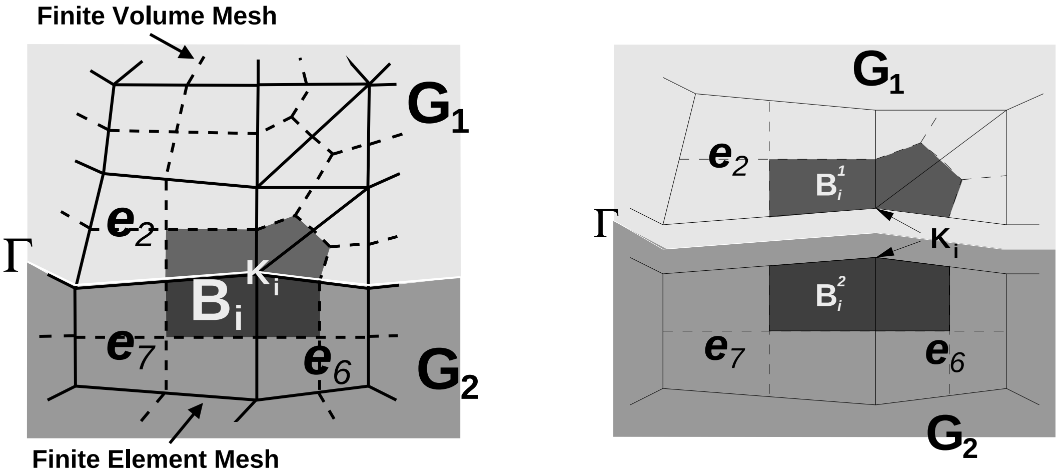


Figure 2.13: Element-oriented Box Method. The interface between two subdomains lies along the element edges (after *Jakobs et al.* [37])

It represents the smallest entry pressure for all elements connected to node K_i:

$$p^i_{c,min} = \min_{k \in E(i)} p_c(\mathbf{x}^k, (1 - S_{n,i})) \tag{2.60}$$

Here, $E(i)$ represents the set of indices of those elements connected to node K_i and $\mathbf{x}^k$ the barycenter of element e_k.

$p^i_{c,min}$ is then used to calculate the saturations $S_{n,i,k}$ belonging to the representation of node K_i in element e_k. This leads to the following relationship:

$$S_{n,i,k} = \begin{cases} 0 & p^i_{c,min} < p_c(\mathbf{x}^k, 1.0) \\ 1 - S_w & \text{where } S_w \text{ solves } \mathrm{Inv}^{e_k}_{p_c}(\mathbf{x}^k, p^i_{c,min}) \end{cases} \tag{2.61}$$

This definition makes it possible for an arbitrary number of subdomains to meet at node K_i.

The element-wise evaluation is only pursued for the domain-dependent variables. All other variables are calculated using the approach described in Section 2.4.2.1. For a further discussion of the PPSIC Method, refer to *de Neef and Molenaar* [49] and *de Neef* [50]. A more detailed description of the implementation of the PPSIC Method can be found in *Bastian* [7].

Remark:

For the stabilization of the advective flux term, a classical first order fully upwinding technique is used in this work for the Box- and Box-Trans Method. Another possible measure would be to apply so-called higher resolution schemes. To these count, for example, the total variation diminishing technique (TVD), flux limiters, and slope limiters. These methods are very efficient on structured grids, for real-life problems, however, which are usually simulated on highly unstructured grids, these methods present some difficulties (see e.g. *Huber* [34]) and are not pursued in this work.

2.4.3 Solving the algebraic equations

Applying the discretizations explained in the previous section to the two-phase flow equations results in a possibly large, strongly coupled non-linear system of algebraic equations.

These equations are solved by the software-toolbox MUFTE-UG. This toolbox is a compound of two parts: whereas the MUFTE part includes the discretization schemes and problem statements, the UG part provides the underlying data structures and fast solvers.

In this work, the two-phase flow equations are solved using a damped inexact Newton-Raphson algorithm for the non-linearities. Within each Newton step, a linear system of equations has to be solved. Here, the Bi-CGSTAB (BiConjugate Gradient Stabilized) solver is applied with a preconditioner based on a multigrid method (*Hackbusch* [26], *Bastian and Wittum* [9]). Within the multigrid method, the smoothing iterations are carried out with the help of ILU (incomplete LU-decomposition). So in order to solve the discretized problem for one time step, a certain number of non-linear steps, which each include its own number of linear steps, have to be solved. This will be returned to in Chapter 6.

Based on the convergence behavior of the Newton method, an automatic time-step adaption is included: If no solution can be found within a certain number of line-searches (here: 6), the time-step Δt is automatically halved. If, however, a solution can be found within the first line-search, Δt is doubled. The lowest and highest bounds for the time-step were set individually for each of the following problems.

A detailed description of the methods and solvers incorporated in MUFTE-UG is given in, for example, *Bastian* [7], *Bastian et al.* [5], *Helmig and Bastian* [27] and *Lang* [44].

As the implemented data-structure of MUFTE-UG supports the multigrid method, the meshes for all numerical examples in this work are constructed in a hierarchical way: At first, a coarse base mesh is generated. This is then consecutively refined (uniformly or adaptively) up to the desired level, and each new level is stored in a separate grid. The computation is then carried out for all leaf elements. Leaf elements denote the 'topmost' elements, when the meshes are put in a hierarchy, where the mesh with the largest elements is at the bottom (BOTTOMLEVEL) and the mesh with the smallest elements resides at the top (TOPLEVEL) (see also Chapter 5). In the following, the maximum number of refinements will be indicated by MAXLEVEL (ML). This number does not represent the absolute value of the element size, as it depends on the structure of the bottom mesh. It gives, however, an indication when the solutions of two different mesh sizes are compared: if ML increases by one, the length of the element edges is (usually) halved. For a more detailed description of the multigrid structured included in MUFTE-UG refer to e.g. *Lang* [44].

3 Two-phase flow in a homogeneous medium

As already pointed out in the introduction, two-phase flow phenomena in porous media have a vast range of applications. Due to the properties of the partial differential equations and the non-linear coupled balance equations (see Chapter 2), the simulation of two-phase flow is by no means trivial.

With the help of the numerical simulation program MUFTE-UG, it is possible, to obtain solutions to a large number of two-phase flow problems. An overview of the modules of MUFTE currently implemented is given in *Class* [17]. In this chapter, the focus is flow in a homogeneous medium. Here, the aim is to point out the difficulties of simulating two-phase flow and to introduce examples which will later be simulated with the help of adaptive methods.

As general two-phase flow combines hyperbolic (advective) as well as parabolic (diffusive) properties, three different test cases are introduced: first, a purely advection-dominated problem (*Buckley-Leverett* problem), then a purely diffusion-dominated problem (*McWhorter* problem) and the last case represents a typical two-phase flow situation where advective as well as diffusive effects are present.

These cases were chosen in order to show that MUFTE-UG and the adaptive methods are not only capable of handling complex two-phase flow, but also the extremes 'advection-dominant' and 'diffusion-dominant'.

3.1 Test case 1: an advection-dominated problem (*Buckley-Leverett* problem)

3.1.1 Problem description

As a fist step for testing the capabilities of MUFTE-UG, the *Buckley-Leverett* (BL) problem is considered. It represents a special case of the two-phase flow equations where advective effects are dominant (see also Section 2.3).

It describes the case of a transient flow in a quasi-one-dimensional, horizontal domain, where the wetting fluid (water) displaces the non-wetting fluid (oil) (see Figure 3.1).

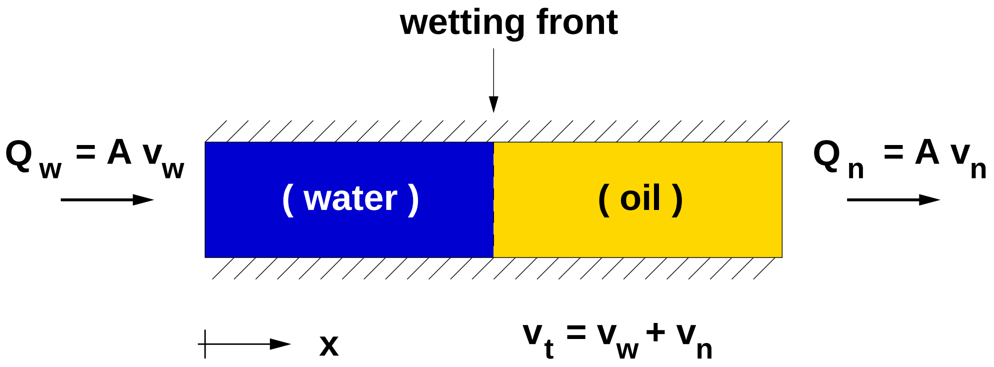


Figure 3.1: Set-up of the *Buckley-Leverett* problem

On the assumptions that we have

- two immiscible and incompressible fluids
- no capillary effects
- no gravitational effects
- no sources / sinks

the saturation formulation of the two-phase flow equations (Eqs. (2.34)) yields (compare Eq. (2.35)):

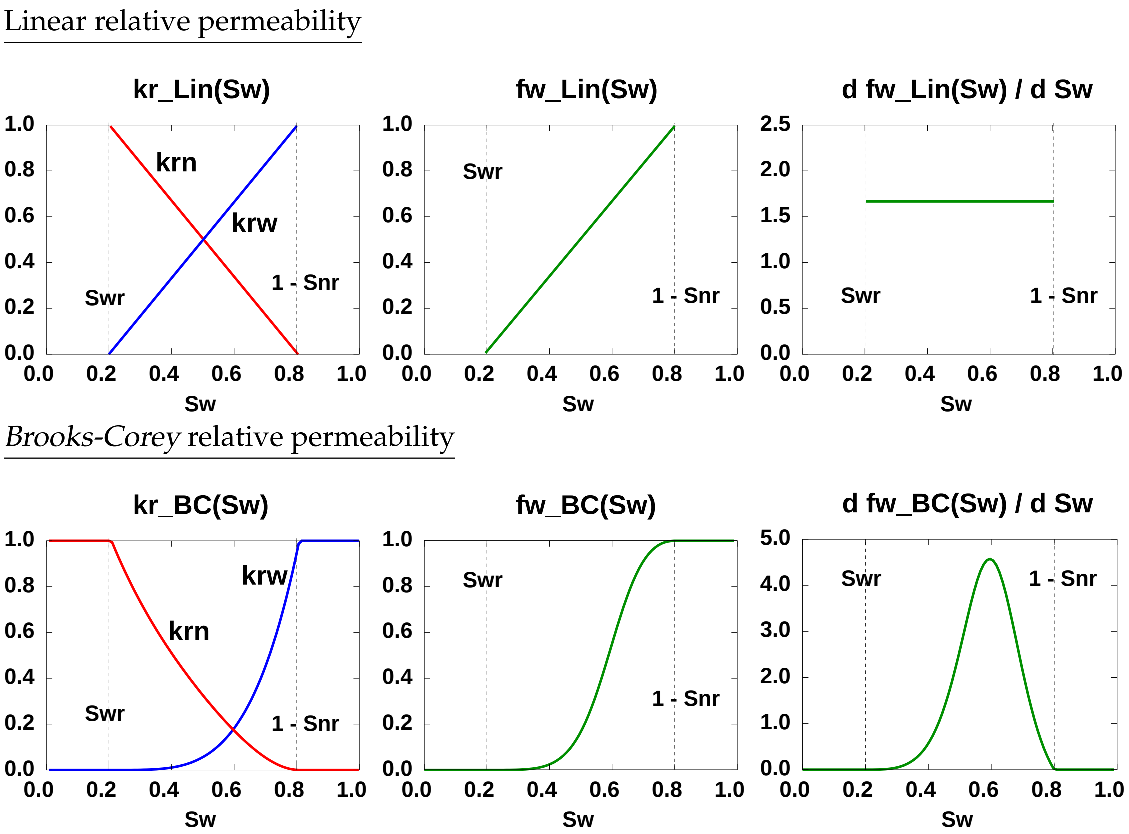
$$\phi \frac{\partial S_w}{\partial t} + \mathbf{v}_t \underbrace{\frac{d\, f_w}{d\, S_w}}_{f_w{}'} \underbrace{\frac{\partial S_w}{\partial x}}_{\mathbf{grad}\, S_w} = 0 \qquad (3.1)$$

This equation, however, is not implemented explicitly, but the above-mentioned assumptions are inserted into the equations for the $p_w - S_n$ formulation (Eq. (2.40)).

The *Buckley-Leverett* problem is chosen for two reasons: On the one hand, some of the effects that have a deep impact on two-phase flows are excluded, so that only a restricted number of processes remains which are thus better to overlook. This makes it easier to associate the results with the changes in parameters. On the other hand, Eq. (3.1) represents a quasi-linear hyperbolic differential equation (no diffusive, but only advective effects). Generally, the main discretization problem of two-phase flow equations lies in the correct description of the advective term. Therefore, as an analytical solution to the *Buckley-Leverett* problem exists (*Buckley and Leverett* [15]), it portrays a convenient method of verifying the numerical simulations.

If only advective effects are present, the interface between the wetting and the non-wetting phases represents a sharp front, where the saturations of the phases experience a jump. Mathematically, this represents a discontinuity in the primary variable S. The front moves with a constant velocity, which depends on the derivative of the fractional flow function. For the linear and the relative permeability function after *Brooks and Corey* the course of f_w and its derivative can be seen in Figure 3.2.

In the case of a **linear** relative permeability, the derivative of the fractional flow function is constant. Consequently, the (analytical) saturation profile resembles a 'step'.



Linear relative permeability

Brooks-Corey relative permeability

Figure 3.2: Relative permeability - saturation functions (*kr*, fractional flow function (f_w) and derivative of the fractional flow function (d f_w) for linear and *Brooks-Corey* relationships

In the case of a **non-linear** relative permeability, the shape of the derivative of the fractional flow function is non-convex. Therefore, more than one value of the saturation can be associated with a certain value of f_w (see Figure 3.2 bottom). As already explained in Section 2.3, the *Rankine-Hugenoit* condition as well as the entropy condition have to be satisfied, which results in an (analytical) front profile comprising a rarefaction wave and a shock front (*LeVeque* [47]).

For the numerical simulation of the *Buckley-Leverett* problem, the system and the boundary conditions depicted in Figure 3.3 are chosen. The assigned fluid and soil properties are given in Table 3.1. The time discretization is realized with an implicit *Euler* scheme with a step size of $\Delta t = 5$ days.

In order to comply with the above-mentioned constraints of the *Rankine-Hugenoit* condition and the entropy condition, the fully upwinding technique ($\alpha_{up} = 1.0$, see Section 2.4) is applied (*Aziz and Settari* [1]).

It has to be remarked that, although the fully upwinding technique is used, still numerical diffusion is introduced in the system due to the first order implicit time discretization (see e.g. *Helmig* [30]).

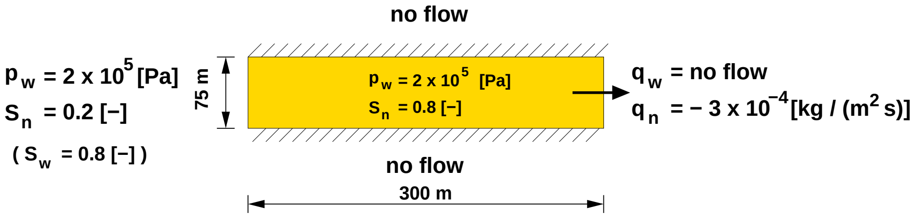


Figure 3.3: Boundary and initial conditions for the *Buckley-Leverett* problem

Table 3.1: Fluid and soil properties (*Buckley-Leverett* problem)

Fluid properties:					Soil properties:				
water density	ρ_w	=	1000	[kg/m^3]	abs. permeability	$\mathbf{K}$	=	10^{-7}	[m^2]
oil density	ρ_n	=	1000	[kg/m^3]	porosity	ϕ	=	0.20	[-]
dyn. viscosity water	μ_w	=	0.001	[kg/(m s)]	res. saturation water	S_{wr}	=	0.20	[-]
dyn. viscosity oil	μ_n	=	0.001	[kg/(m s)]	res. saturation oil	S_{nr}	=	0.20	[-]

Table 3.2: Simulation parameters (*Buckley-Leverett problem)*

Initial conditions:					Discretization:				
water pressure	p_w	=	2 x 10^5	[Pa]	step size (space)	Δx	=	9.375	[m]
oil saturation	S_n	=	0.8	[-]	step size (time)	Δt	=	5	[d]

3.1.2 Investigations with a uniform mesh

Influence of the discretization length

As a first step, simulations are carried out for different element sizes (Figure 3.4). Two of the meshes used are shown exemplarily in Figure 3.5. The elements are divided sequentially, so that the aspect ratio of 1:1 is kept at all times. It can be observed that, for decreasing element size, the solutions converge towards the analytical solution.

The following table shows the evolution of the number of elements and the computational time needed exemplarily for the case of the constitutive relations after *Brooks-Corey* (BC) for different refinement levels:

	Level 2	Level 3		Level 4	
execution time [s]	12	46	(3.9)	226	(19)
elements	64	256	(4)	1024	(16)
nodes	85	297	(3.5)	1105	(13)

The number in brackets represent the ratio between each individual level and level 2. Here, it can be noted that the needed time increases approximately linearly with the number of elements. Using h-adaptive methods reduces the node number and therefore the CPU-time while the size of the smallest element remains the same (see Chapter 6).

For the following investigations concerning the introduction of artificial diffusion (artificial capillary pressure), a mesh with 256 elements (Δx = 9.375 m (ML 3)) is used, if not otherwise stated.

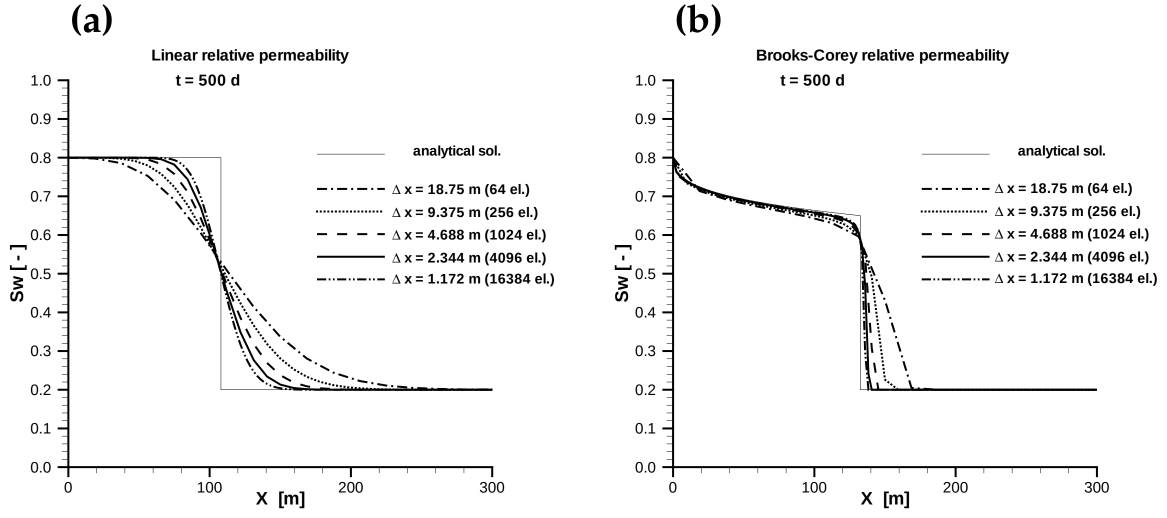


Figure 3.4: Comparison of the numerical solution for different element sizes (Left: linear; right: BC)

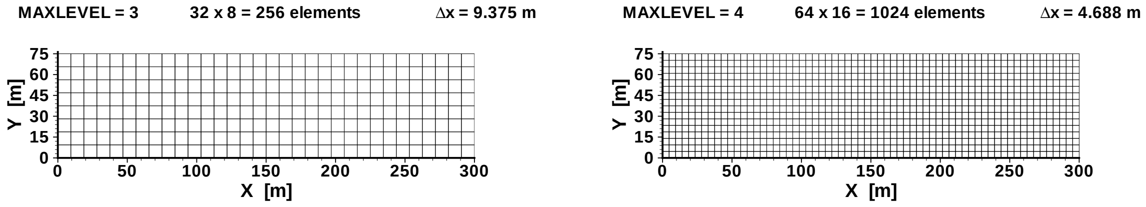


Figure 3.5: Two different meshes for the *Buckley-Leverett* problem

Influence of artificial diffusion

As described in Section 2.4, a hyperbolic partial differential equation can be handled by using a fully upwinding scheme. Another possibility is to introduce an artificial diffusion term by selecting an artificial capillary pressure gradient.

In order to test this influence, a linear $p_c(S_w)$-relationship is used and upwinding is omitted. To realize different capillary pressure gradients, $p_c\mathrm{max}$ is varied. The chosen values are

$p_c\mathrm{max}$ [Pa]	3.0	0.6	0.06	0
$\frac{d\,p_c}{d\,S_w}$	5	1	0.1	0

The results for the linear $kr(S)$-relationship and the one after *Brooks-Corey* can be seen in Figure 3.6. The results obtained are compared to a simulation using fully upwinding.

All simulations are carried out on a uniform mesh with a refinement level of ML 3. It is shown that the selection of a suitable diffusion term is problem-dependent and demanding (*Helmig* [31]): If the capillary-pressure gradients are too small, oscillations can appear and the front shape may be wrong. To reduce these oscillations, the diffusive effects and therefore the capillary-pressure gradients can be increased. Yet if the capillary pressure gradient is chosen too large, the front velocities increase significantly and the numerical simulation produces wrong results - which may not always be obvious if no analytical solution exists.

Concluding, one can say that for an advection-dominated problem a fully upwinding scheme should be preferred to an introduction of artificial diffusion - which is a well known fact and discussed in many papers in the recent years (e.g. *Kröner* [41]).

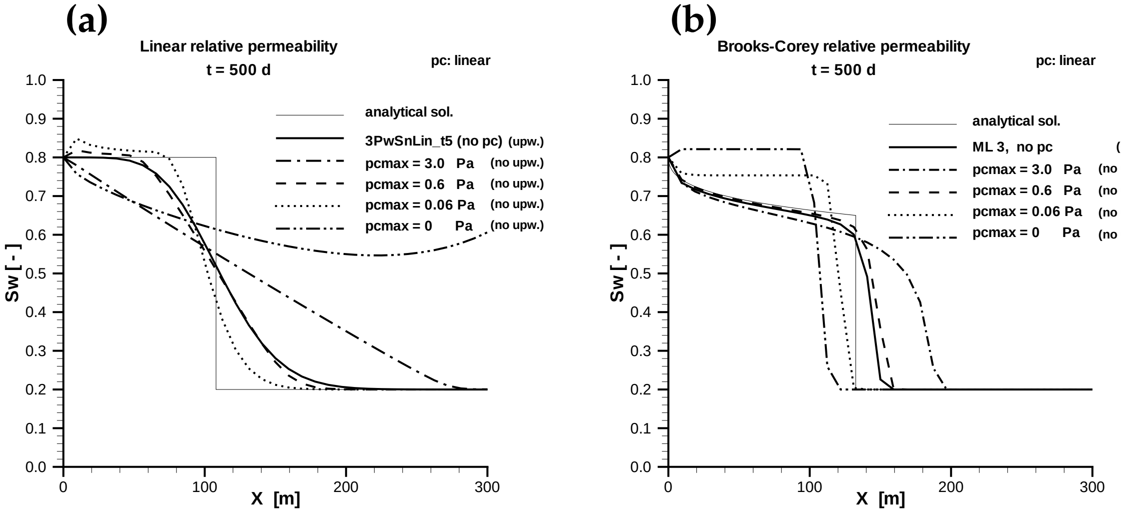


Figure 3.6: Comparison of different $p_{c\text{max}}$ (Left: linear; right: BC)

3.2 Test case 2: a diffusion-dominated problem (*McWhorter* problem)

The *McWhorter* problem describes an instationary displacement process of oil by water (see Figure 3.7), which is driven only by capillary forces and hence resembles a parabolic system.

On the assumption that we have

- two immiscible and incompressible fluids
- no gravitational effects
- no sources / sinks

the saturation formulation of the two-phase flow equations (Eq. (2.34)) yields:

$$\phi \frac{\partial S_w}{\partial t} + \mathbf{v}_t \frac{d f_w}{dS_w} \frac{\partial S_w}{\partial x} + \frac{\partial}{\partial x}\left(\mathbf{D}\frac{\partial S_w}{\partial x}\right) = 0 \tag{3.2}$$

where $\mathbf{v}_t$ represents the total flow velocity, f_w the fractional flow function and $\mathbf{D}$ the so-called *dispersion tensor*

$$\mathbf{D}(S_w) = \frac{k_{rn}}{\mu_n} f_w \mathbf{K} \frac{dp_c}{dS_w} \tag{3.3}$$

Again, this is not the equation that is implemented, but the assumptions mentioned above are inserted into the general two-phase flow equations. For the *McWhorter* problem, a $p_n - S_w$ formulation is used (see Section 2.3). As the *McWhorter* problem is a purely diffusion-dominated problem, a central weighted mobility (α_{up} = 0.0, see Section 2.4) is used.

The system and boundary conditions for the numerical investigation are presented in Figure 3.8, the applied fluid and soil parameters are shown in Table 3.3, and the simulation parameters are given in Table 3.4.

For the case of a uniform refinement, the simulation results of the water saturation distribution for different element sizes are shown in Figure 3.9. As this problem represents a quasi-one-dimensional problem, a blue refinement rule (anisotropic refinement, here: only in x-direction) for the consecutive uniform refinement is chosen (see Section 4.2). *McWhorter and Sunada* [48] have developed a quasi-analytical solution for this problem, which makes it possible to verify the numerical simulations. As can be seen, the error becomes smaller for smaller element sizes - a behavior which is expected, due to the convergence properties of the Box Method (see Section 2.4.2).

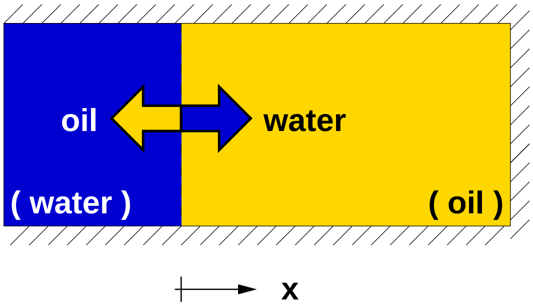


Figure 3.7: Set-up of the *McWhorter* problem

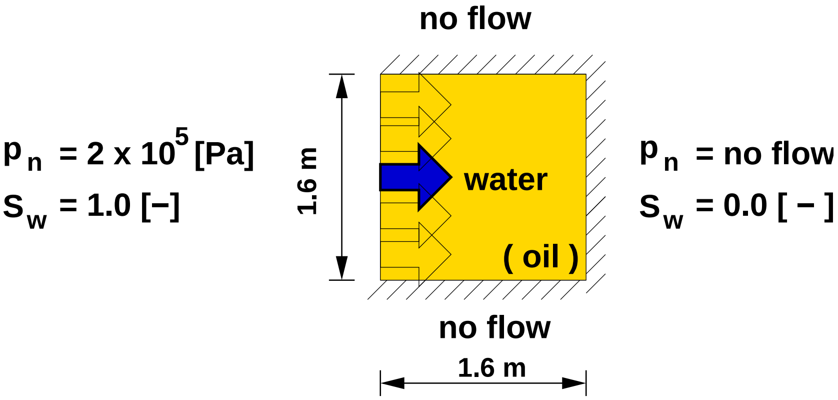


Figure 3.8: Boundary conditions of the *McWhorter* problem

Table 3.3: Fluid and soil properties (*McWhorter* problem)

Fluid properties:					Soil properties:				
water density	ρ_w	=	1000	[kg/m^3]	abs. permeability	**K**	=	10^{-10}	[m^2]
oil density	ρ_n	=	1000	[kg/m^3]	porosity	ϕ	=	0.3	[-]
dyn. viscosity water	μ_w	=	0.001	[kg/(m s)]	res. saturation water	S_{wr}	=	0.0	[-]
dyn. viscosity oil	μ_n	=	0.001	[kg/(m s)]	res. saturation oil	S_{nr}	=	0.0	[-]
					entry pressure	p_d	=	5000.0	[Pa]
					pore size distr. index	λ	=	2.0	[-]

Table 3.4: Simulation parameters (*McWhorter* problem)

Initial conditions:					Discretization:				
oil pressure	p_n	=	2 x 10^5	[Pa]	step size (space)	Δx	=	0.05	[m]
water saturation	S_w	=	0.0	[-]	step size (time)	Δt	=	100	[s]

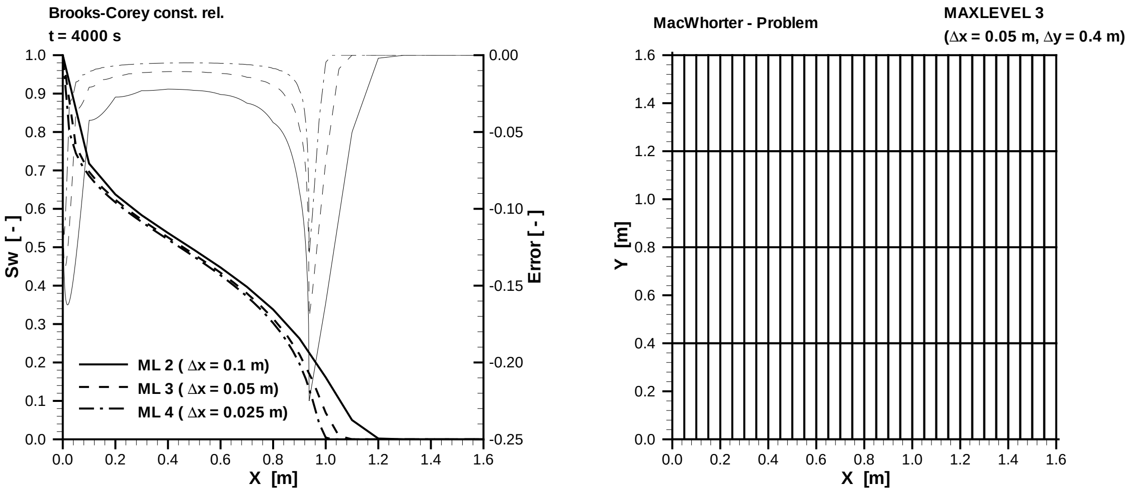


Figure 3.9: *McWhorter* problem: Water saturation S_w (thick lines) and error distribution (thin lines) for three different discretization lengths (left) and representative mesh (right)

3.3 Homogeneous model problem (Sandbox)

A numerical simulation tool has to be capable of dealing not only with the extreme sides of the two-phase flow equations, but also with the case that both behaviors - advection-dominant and diffusion-dominant - occur in one system in different places. In order to represent this problem, the model set-up for a homogeneous medium as depicted in Figure 3.10 is chosen.

It represents a common problem in environmental engineering: over a small area, a contaminant (e.g. oil) infiltrates into the (fully water-saturated) subsurface and the question arises how and over which time period will the contaminant spread throughout the system. Although this example represents a relatively simple case of two-phase flow, it also combines many of the properties a two-phase system reacts sensitively to: difference in densities and viscosities, different capillary pressure - saturation relationships and time dependence.

Also, different zones can be distinguished: along the oil front moving into the system, advection dominates, whereas in the area behind the front, diffusion is more dominant.

The infiltration is simulated over a time period of 24 h, with the fluid and soil properties of the system given in Table 3.5 and the simulation parameters in Table 3.6. In order to ensure stability along the infiltration front, the upwinding value α_{up} is set to 1.0. The result is shown in Figure 3.11. As expected, the oil distribution is symmetric. The reason for the isolines not being perfectly rounded is the discretization scheme used: As the Box Method only regards the flows over the side of a box, it resembles a finite difference 5-point stencil. Therefore, flow vectors which run diagonally across an element can be reproduced only moderately well. A better result would be obtained by applying the CVFE-method, for example (see e.g. *Helmig* [31], *Class* [17]).This is, however, not realized in this work.

As explained in Section 2.4.2, the accuracy of a numerical method improves with decreasing element size. Therefore, a comparative solution is simulated with a finer mesh (twice refined). The results are compared in Figure 3.12. It can be seen that there is a difference in the propagation of the oil saturation: the result for the larger mesh size is much more smeared.

When the element number is increased, the computational time increases as well. It is obvious, however, that the smaller elements are not needed throughout the whole domain, as there exist regions where no relevant flow takes place. So again the need for adaptive methods is shown: they combine the advantages of small element sizes and low computational time. As the elements are only refined where required, the overall element number and therefore the computational effort stays low.

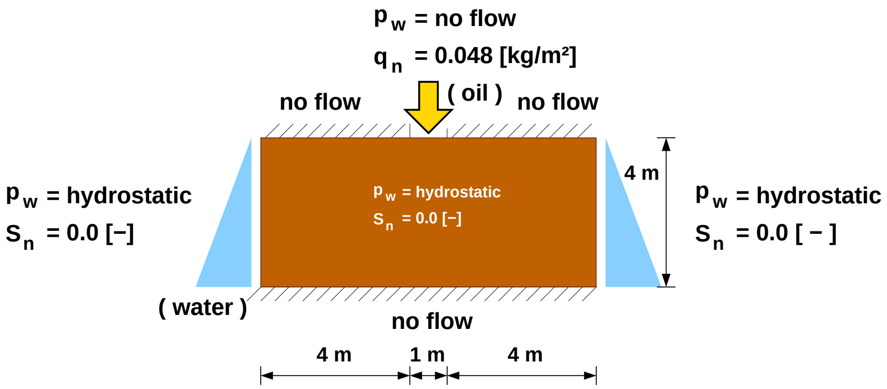


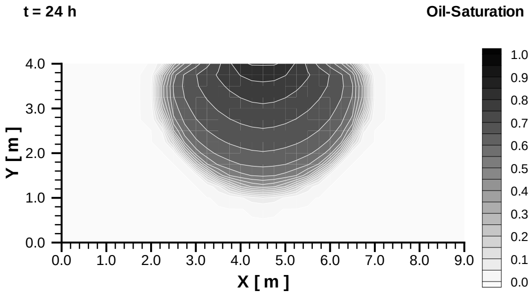
Figure 3.10: System, boundary and initial conditions of the homogeneous system

Table 3.5: Fluid and soil properties (homogeneous model problem)

Fluid properties:					Soil properties:				
water density	ρ_w	=	1000	[kg/m^3]	abs. permeability	$\mathbf{K}$	=	8 x 10^{-12}	[m^2]
oil density	ρ_n	=	1460	[kg/m^3]	porosity	ϕ	=	0.4	[-]
dyn. viscosity water	μ_w	=	0.001	[kg/(m s)]	res. saturation water	S_{wr}	=	0.05	[-]
dyn. viscosity oil	μ_n	=	0.0057	[kg/(m s)]	res. saturation oil	S_{nr}	=	0.0	[-]
					entry pressure	p_d	=	700.0	[Pa]
					pore size distr. index	λ	=	2.0	[-]

Table 3.6: Simulation parameters (homogeneous model problem)

Initial conditions:					Discretization:				
water pressure	p_w	=	hydrostatic	[Pa]	step size (space)	Δx	=	0.25	[m]
oil saturation	S_n	=	0.0	[-]	step size (time)	Δt	=	3600	[s]



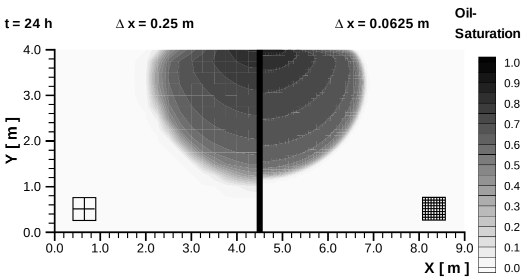


Figure 3.11: Oil saturation after 24 h

Figure 3.12: Comparison of the simulations with different mesh widths

4 Adaptive methods

As stated in Section 2.4, for the Box Method the complete problem domain has to be described by a mesh. This mesh can be structured or unstructured and certain areas of interest, for example lenses or contamination sources, can be discretized with smaller elements than the surrounding domain. Yet one disadvantage remains: all these meshes remain static throughout the computation. As, however, mostly transient problems have to be solved in which the behavior of the solution over the simulated time is usually not known, the mesh created originally may not be appropriate after various time steps. Therefore it is desirable for the efficiency of the numerical method to adjust the space discretization according to the current solution. This can be achieved with adaptive methods.

Generally, one can distinguish between space and time adaptive methods. Whereas in space adaptive methods for example the mesh or order of the weighting functions are changed, time adaptive methods adjust the time step.

In the course of this work, only a space adaptive method is applied. On the one hand, as in this work not only a space adaptive method is expanded, but an adaptive choice of a suitable numerical scheme is developed and investigated also, an additional consideration of time adaptive methods would exceed the scope of this work. On the other hand, space adaptive methods can be considered 'more urgent' for (large) engineering problems due to the limitation of storage capacity and computational time.

However, MUFTE-UG already includes an automatic time step control based on the convergence behavior of the Newton method, as it is explained in Section 2.4.3.

In the following sections, first an overview over different (space) adaptive methods is given. Then, the adaptive method applied in this work is presented.

4.1 Overview

Numerical solutions are always only an approximation of the accurate solution. In the course of arriving at the numerical solution, we may encouter many possible error sources. For example, the solution algorithm may be developed incorrectly, there may be bugs in the program or the boundary conditions are not chosen properly. Additionally, there are always three kinds of systematic errors involved (*Ferziger and Perić* [24]):

1. *Modeling errors*: The exact solution of a mathematical model does not represent the actual flow exactly, as generalizations and assumptions are made to arrive at the mathematical model.
2. *Iteration errors*: They describe the difference between the exact solution of the discretized algebraic equation system and the iterative solution.
3. *Discretization errors*: They comprise the difference between the exact (analytical) solution of the conservation equations and the exact solution of the algebraic system of the discretized equations.

The effect of modeling errors is that the model equations may be fulfilled exactly, the solution, however, is qualitatively wrong. Modeling errors can only be discovered when solutions, where the iteration and discretization errors are very small, are compared with accurate experimental data. Errors due to the rounding-off of the iterative schemes are easier to control (and usually not very problematic), but nonetheless impossible to avoid, as one always has to deal with the 'limited' number representation of computers. And last but not least, discretization errors are also always present due to the nature of this scheme. They decrease, however, as the grid is refined. This implies that discretization errors only give a *rate* at which the error decreases for smaller element sizes - they do not give information about the error on a single grid.

Adaptive methods can help to minimize the discretization errors. Generally, the following adaptive methods can be distinguished (e.g. *Ellsiepen* [23], *Hinkelmann* [33]):

- *h-adaptive methods*
  Single elements are refined or coarsened (change of the element radius *h*). This results in a change of the node and element densities at certain locations. For refinement, the element is divided at the midpoint of a side or at the center of gravity; for coarsening, this process is reversed. The initial elements serve as a reference for the geometry as well as for the associated material properties and must therefore not be coarsened.

  In the h-adaptive method, the quality of the starting mesh is of high importance, as the angles and aspect ratios of the side lengths of the elements are kept throughout the computation. In some cases, this may be regarded as a disadvantage of this method.

  Another requirement of the h-adaptive method is a dynamic data structure, as the number of elements is constantly changing.

- *p-adaptive methods*
  The order *p* of the polynomial testing functions is changed for selected elements. As a consequence, the formulations for an element get more elaborate, which may - especially for 3D - lead to a state where the solving expenses reach a dramatic height. Futhermore, this method requires a very high regularity of the solution, which may not be given for all problems (e.g. mixed hyperbolic / parabolic problems).

- *h-p-adaptive methods*
  As a combination of the two above-mentioned methods, the element size *h* as well as the order of the polynomial *p* are adapted to the local behavior of the solution.

- *r-adaptive methods*
  This method is also called the *moving mesh method*. The aim is to achieve the best orientation of the mesh with a pre-defined number of degrees of freedom for a user-defined criterion. The nodes of the mesh are moved e.g. to the location of the maximum error or in alignment to the streamlines, which results automatically in a reduction of nodes in other parts of the domain. An advantage of this method is the simple data storage (the number of nodes and elements stays constant) and a preservation of the (stored) structure of the original grid.

  On the other hand, however, the r-adaptive method is less flexible than the other methods mentioned above, as the movement of the nodes can only take place up to a certain extent (the angle- and side-ratio criteria still have to be fulfilled). A further disadvantage is the fact that the element matrices have to be computed again for every iteration step, which means that the r-adaptive method is not useful for complex domains and should only be used for simple problems.

- *subgrid methods*
  Arbitrarily orientated quadrilateral grids (”subgrids”) are superimposed over the original grids. These subgrids can have a smaller element size than the underlying basic grid. As the time step is coupled with the element size, the subgrids are computed with a smaller time step. A great disadvantage of this method is the restriction to geometrically simple domains.

- In addition to these methods, which aim at an efficient numerical solution of a given problem, the following methods can also be counted among the adaptive methods:

  - *d-adaptive methods*
    In some applications, the appropriate space-dimension is chosen automatically (switch between 1D, 2D, and 3D).
  - *m-adaptive methods*
    According to a given variable, the suited model concept is chosen (for example, for large *Reynolds* numbers, switch from *Darcy* velocity to *Forchheimer* velocity). Another application is the switching of the primary variables according to certain criteria (see e.g. *Class* [17], *Class et al.* [18]).

Generally, the h- and p-adaptive methods are commonly used in engineering applications. For the two-phase flow problems considered in this work, the h-adaptive method is chosen (see Section 4.2). On the one hand, the data structure provided by MUFTE-UG supports this method very well, on the other hand the dramatic increase in computational effort which results from the p-adaptive method would outweigh the possible improvement of the numerical results.

In order to refine or coarsen the mesh, error criteria are applied. The error evaluation can be divided into two classes:

1. **a priori** error evaluation:
   These estimates are evaluated before the calculation is started. They use exclusively

known data, as for example initial and boundary conditions, as well as material parameters. A priori estimators are only rarely used for practical purposes.

2. **a posteriori** error evaluation:
   Here, the error is given in terms of the starting data and computed solution (for example after each time step).

In the following, only a posteriori error evaluators will be regarded.

Another distinction which can be made is between

- **error estimators**, which are mathematically based,
  and
- **error indicators**, which are usually physically motivated and obtained by heuristic methods.

Basically, a posteriori error estimators and indicators can be roughly divided into the following classes, which will be explained here briefly. A detailed description can be found in e.g. *Ellsiepen* [23] and *Verfürth* [60].

1. **Residual-based error estimator**
   With the help of the residuum appertaining to the strong form of the partial differential equation, the error of the numerical simulation is measured. This can be regarded as an elementwise controlling of whether the partial differential equations are fulfilled or not.

   The basis of this approach, which was first presented by *Babuška and Rheinboldt* [2], [3], is the estimation of the error in the energy or L^2 norm via the residual of the finite element solution in the domain and at the boundary. Various others scientists, for example *Johnson* and his co-workers (e.g. *Johnson and Hansbo* [39]) took up and extended this approach.

2. **Error estimation by solving local problems**
   For small areas (for example single elements or element patches), higher-order solutions are calculated. In these areas, however, not the given problem, but a similar, simpler one is solved. The solution is then taken as a reference solution for the error estimation.

3. **Hierarchical error estimators**
   The residuum of the numerical solution is compared with a solution that was calculated in a higher finite element space (higher order of test functions or finer meshes). The idea behind this is that the solution in a higher FE-space is expected to be more accurate than on a lower FE space (convergence of the p-method). For further details, see e.g. *Ellsiepen* [23].

4. **Gradient-based error indicators**
   This kind of indicator, which was originally developed for structural mechanics, is often referred to as the Z^2-error indicator, after the initials of the authors that first introduced this idea in 1987 (*Zienkiewicz and Zhu* [61]).

   The gradients of FE solutions with linear shape functions are usually not continuous over the element edges. For certain gradients (for example the tension σ in the case of the Z^2 indicator or the velocity v in the case of groundwater flow), a locally 'improved' solution is calculated by computing a 'smoothed' course of the gradients. With the help of the norm of the difference between the gradients of the original and the gradients of the 'improved' solution, the value of the error estimator is calculated.

   Error estimators which are based on the energy or L^2 norm are widespread for parabolic and elliptic problems (e.g. *Johnson* [40], *Papastavrou* [55]). They cannot, however, generally be applied for mixed hyperbolic-parabolic equations.

5. **Empirical error indicators**
   Here, the error criterion is empirically or heuristically derived. As a criterion can serve, for example, steep gradients or large curvatures of the solution.

Bürkle and Ohlberger [16] developed an error estimator for convection-diffusion equations. They also show results for two-phase flow in porous media, however, they concentrate on convection dominated flow only (*Ohlberger* [53] [54]). Additionally, an IMPES-formulation (implicit pressure - explicit saturation) is applied – a method in which the flow equations are decoupled. This approach cannot be taken when, for example, two-phase, two-component flows are considered.

As it is the aim of this work to present an indicator suitable for engineering practice (that is, relatively easy to implement and operate), empirical error indicators will be applied. These are discussed in further detail in Section 4.2.

After each element has been assigned an 'error value' η by an estimator or indicator, the elements are marked for refinement or coarsening. Generally, the error of the element is compared to tolerances for refinement (tol_{refine}) and coarsening ($\text{tol}_{coarsen}$):

$$\begin{array}{lcl} \eta > \text{tol}_{refine} & \longrightarrow & \text{refine} \\ \eta < \text{tol}_{coarsen} & \longrightarrow & \text{coarsen} \end{array} \tag{4.1}$$

For the choice of tolerances, different possibilities exist (e.g. *Barlag* [4]). For example:

- **Absolute value**
  The tolerances are set to fixed values which do not change throughout the simulation. As the magnitude of the errors in the course of the simulation is not known and may vary significantly, this method is not advisable.

- **Mean value**
  The tolerance is given in connection with the arithmetic mean of the errors of all elements. With this method, not only the size of the error, but also the amount of 'erroneous elements' is considered:

$$\begin{aligned} \mathrm{tol}_{refine} &= \mathrm{ref}^{mean}\frac{1}{N}\sum_j \eta_j \\ \mathrm{tol}_{coarsen} &= \mathrm{cor}^{mean}\frac{1}{N}\sum_j \eta_j \end{aligned} \tag{4.2}$$

  Here, N is the amount of elements while η_j represents the error for element j. ref^{mean} should be chosen to be larger and cor^{mean} to be smaller than 1. This method is disadvantageous for small absolute errors.

- **Maximum value**
  The refinement and coarsening criteria represent the percentage of the maximum error in the domain:

$$\begin{aligned} \mathrm{tol}_{refine} &= \mathrm{ref}^{max}\ \max\eta_j \\ \mathrm{tol}_{coarsen} &= \mathrm{cor}^{max}\ \max\eta_j \end{aligned} \tag{4.3}$$

  A value of $\mathrm{ref}^{max} = 0.8$ means that all elements with an indicator value larger than 80% of the maximum indicator value (in this time step) are refined.

Figure 4.1 illustrates the principle of refinement and coarsening for the maximum-value method used in this work. It can be observed that the appropriate choice of refinement and coarsening bounds is very important: if a large peak of the indicator values exists in the domain, other areas with a relatively large indicator value may not be identified. Generally speaking, the choice of the values for the refinement and coarsening criteria always depends on the error estimator or indicator as well as on the problem itself.

Beside the values for e.g. ref^{max} and cor^{max}, the number of refinement (`to`) and coarsening (`from`) levels has to be given. A refinement level of two, for example, means for the h-adaptive method that an element will be divided twice. For a quadratic element, this results in 16 elements with an edge length of $\frac{1}{4}\, l_{org.element}$. Figure 4.2 shows an h-adapted mesh for a refinement level 3 and a coarsening level 1. In the course of this work, these levels will be

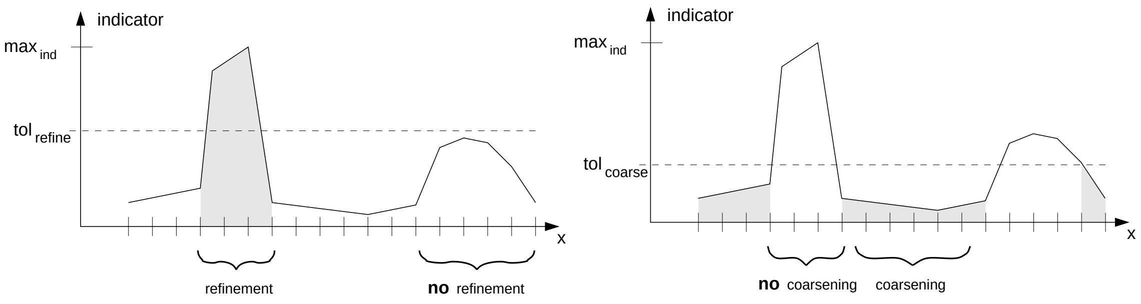


Figure 4.1: Refinement and coarsening bounds for the 'maximum-value' method

indicated by '*fntn*', where f stands for level the coarsest grid (`from`), t represents the level for the finest grid (`to`) and *n* indicates the level number in the hierarchy. For a better overview, hanging nodes still exist in this figure (for an explanation, see Figure 4.3); in MUFTE-UG these are eliminated by introducing irregularly refined elements (see Section 4.2 and *Lang* [44]).

If the difference between the coarsening and refinement level is too large, the difference between the largest and smallest element sizes in the domain will become very large, too. This in turn leads to bad condition numbers of the stiffness matrices, so that the problem may become unsolvable. Here, it is the task of the user to find the appropriate balance.

4.2 Error indicators in MUFTE-UG

In the course of this work, the h-adaptive method is applied. It has the advantage that a visual control of whether the refinement and coarsening identify the desired regions is possible by simply looking at the resulting mesh. And last but not least, the applied numerical simulator MUFTE-UG supports this method very well, as it provides a powerful dynamic data structure and a multigrid environment. For an overview of the implemented structures, refer to e.g. *Bastian* [7] and *Lang* [44].

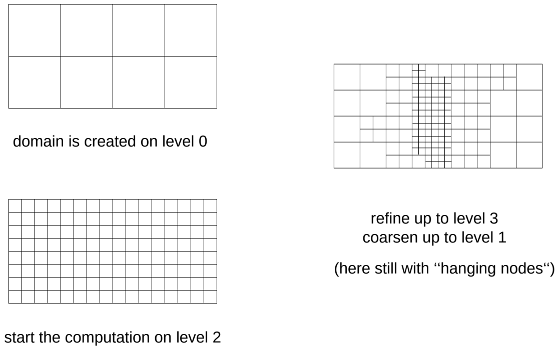

domain is created on level 0
start the computation on level 2
refine up to level 3
coarsen up to level 1
(here still with "hanging nodes")

Figure 4.2: Result of a refinement for `from:1` and `to:3`

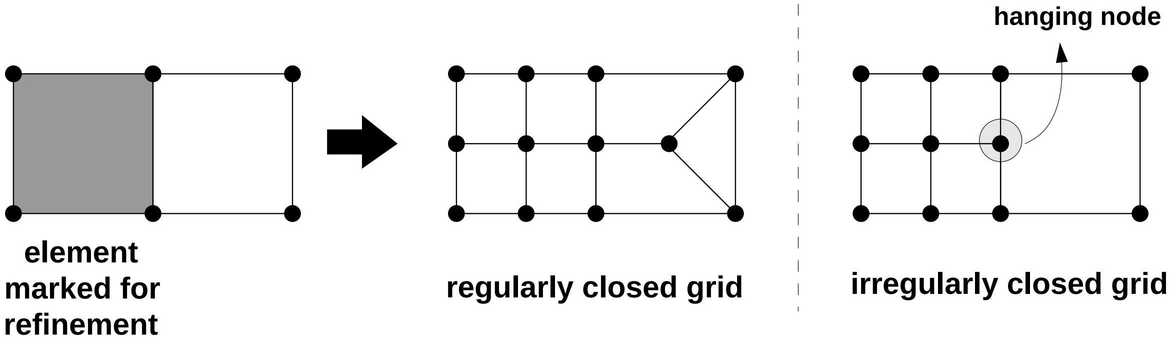


Figure 4.3: Refinement without and with hanging nodes

An h-adapted mesh consists of different types of elements, as can be seen in Figure 4.4. In the case of a multigrid environment, elements which are created by refinement (so-called **son elements**) are inserted in a multigrid level ’above’ the previously marked elements (so-called **father elements**). In a straightforward extension, all sons of an element are called **brothers**.

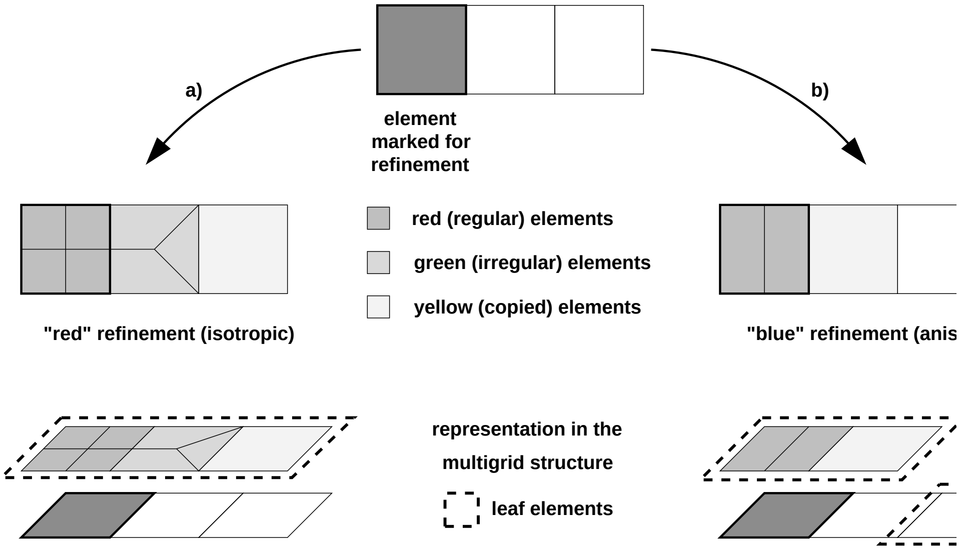


Figure 4.4: Different refinement strategies (refinement depth: one level). View of the leaf elements and the multigrid structure.

If an element is marked for refinement, it will be divided according to the refinement rule (regular (red) or irregular (blue)) and the refinement depth into regular elements, which are also called **red elements**. A regular refinement here denotes a bisection of every element edge, whereas an irregular refinement implies a preferential refinement direction. The angle ratios of red elements are not (seriously) impaired. In order to avoid hanging nodes, the neighboring elements of the marked element usually have to be subdivided, too. These new elements are called **green elements** or irregular elements, as they are only needed for a closure of the grid. According to the closure rules (see e.g. *Lang* [44] and Figure 4.5), green elements usually differ in shape from red elements, comprise acute angles and are never refined themselves. They will be discarded and replaced by elements forming a red pattern when further refinement is required. In addition to red and green elements, **yellow elements** also exist; these are elements copied to the higher grid levels in order to improve the convergence properties of the local multigrid. Yellow elements keep the same angle ratio as the original elements.

Another special kind of element is the **leaf element**. As shown in Figure 4.4, these represent the topmost elements. Leaf elements can lie on different grid levels.

The process of obtaining a numerical solution with an adapted mesh in MUFTE-UG can be represented by the sequence of steps presented in Figure 4.6. First, the problem is solved on all grids in the multigrid hierarchy stemming from the previous time step. Then, for the

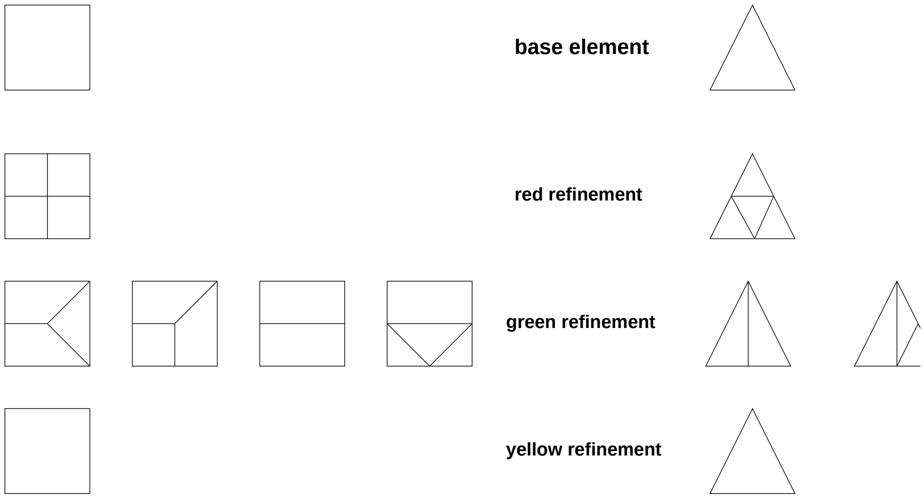


Figure 4.5: Refinement rules for quadrilateral and triangular elements implemented in MUFTE-UG

grid on the toplevel, the error indicator values are computed and the respective elements are marked for coarsening or refinement. After that the mesh is adapted and in the next step the unknowns are transferred to the new mesh structure. Here, also the required new node values are computed. If the corresponding option is set, the cycle for the mesh adaption is repeated for the same time step. If either this option is not set or the chosen number of cycles is reached, the problem is again solved on the finest grid only, which completes the computation for the current time step.

As already stated in Section 4.1, only heuristic indicators are used in this work. On the one hand, as explained before, the author is currently not aware of the existence of estimators for the here attended equations and discretization schemes. On the other hand, the aim of this work is to introduce methods for engineering practice. In this context, a simple handling of the parameters and an easy interpretation of the effects of parameter variation plays a more important role than mathematical accuracy of the indicator. Furthermore, for practical purposes, the results obtained by the use of heuristic indicators are accurate enough (*John* [38]).

It has been shown, for example in *Barlag* [4], that indicators are usually not only easier to implement in the code, but are also easier to compute. Additionally, the computational as well as the storage effort is much less than for estimators. Therefore, simulations controlled by indicators will require much less CPU-time - an important fact for engineering practice.

The basic idea for error indicators in this work is to locate the regions with a sharp interface between the two phases, as this region usually leads to numerical difficulties. A sharp interface can be identified by regarding the gradient of the saturations (*Braun* [11]). As the constraint $S_w + S_n = 1$ holds, it does not matter which saturation is considered; consequently, it is always the primary variable that is taken into account.

The concept is as follows: For the local midpoint of an element, the gradient of the saturation S in x- and in y-direction is calculated. Afterwards, the norm of these two values is computed:

$$\text{grad\_S} = \sqrt{\left(\text{grad}_x(S)\right)^2 + \left(\text{grad}_y(S)\right)^2} \tag{4.4}$$

In the following, this indicator will be called the **gradient indicator *grad_S***.

In addition to this indicator, a **hierarchical indicator *grad_HS*** was also investigated. Here, the gradients on the father element are also considered when the indicator is calculated:

$$\text{grad\_HS} = \sqrt{\left(\text{grad}_{x,son}(S) - \text{grad}_{x,father}(S)\right)^2 + \left(\text{grad}_{y,son}(S) - \text{grad}_{y,father}(S)\right)^2} \tag{4.5}$$

For a comparison, a third indicator (***grad_HpS***) is investigated as well, where not only the gradient of the saturation, but also the gradient of the pressure is taken into account:

$$\begin{aligned} \text{diff}_S &= \sqrt{\left(\text{grad}_{x,son}(S) - \text{grad}_{x,father}(S)\right)^2 + \left(\text{grad}_{y,son}(S) - \text{grad}_{y,father}(S)\right)^2} \\ \text{diff}_p &= \sqrt{\left(\text{grad}_{x,son}(p) - \text{grad}_{x,father}(p)\right)^2 + \left(\text{grad}_{y,son}(p) - \text{grad}_{y,father}(p)\right)^2} \\ \text{grad\_HpS} &= \text{diff}_S + \text{diff}_p \end{aligned} \tag{4.6}$$

For the last indicator, a problem-dependent weighting scheme could be advisory. As the magnitudes of pressures can range in the thousands, but the saturation ranges only between zero and one, both differences need to be scaled accordingly. This approach, however, was not further pursued in this work.

All indicator values are scaled with the area of the concerning element. The refinement and coarsening criteria are chosen as percentages of the maximum error value in the domain, as explained in Section 4.1.

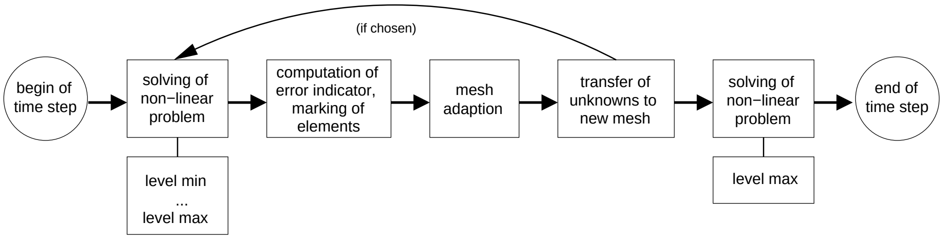

(if chosen)
begin of time step
solving of non−linear problem
computation of error indicator, marking of elements
mesh adaption
transfer of unknowns to new mesh
solving of non−linear problem
end of time step
level min ... level max
level max

Figure 4.6: Program steps for calculating one time step with mesh adaption

5 The aspect of mass conservation

For all (numerical) simulations, it is important to conserve the mass in the system and to balance it correctly with inflow and outflow conditions. This is done by using mass-conservative discretization schemes, as for example the Box Method (see Section 2.4.2). This method is locally mass conservative: the inflow and outflow over the boundary of a control volume are in equilibrium.

However, when solutions obtained on different meshes are compared, the mass of the fluids in the system can still be different. This is due to the choice of initial conditions for the system, which have a large influence on the solution. Additionally, it also has to be ensured that the mass inside the system stays the same for grid manipulations, as is the case for adaptive methods. These aspects will be discussed in the following sections.

5.1 Mass conservation in the uniformly refined case

In the case of a simulation with a uniformly refined mesh, no grid manipulations and therefore no data manipulations are carried out. However, if comparing different realizations with different grid sizes, one has to pay attention to the choice of initial conditions. This will be explained with the help of the *Buckley-Leverett* problem.

The set-up and initial conditions are shown in Figure 5.1. In Fig. 5.1(a), the initial oil saturation in the whole domain is chosen as $S_n = 0.8$. As the unknowns are assigned to the nodes and the nodal values are interpolated linearly, this results in the saturation distribution that is shown in the lower part of Figure 5.1(a). As can clearly be seen, the initial mass in the system is not the same in the two cases shown: the more finely the mesh is refined (here: level 2 compared to level 1), the larger the initial mass of oil in the system. This means that two different problems are solved and should therefore not be compared. If one wants to compare the solutions of different mesh sizes, the approach shown in Figure 5.1(b) has to be followed. For a region comprising the area of the largest element considered, the initial condition is chosen in such a way that the saturation increases linearly from the boundary into the domain. Here, it is ensured that the initial mass in the system (and the initial pressure gradients at the boundary) is the same in both cases.

The influence on the solution of the *Buckley-Leverett* problem can be seen in Figure 5.2. For the different realizations of the initial conditions, the solutions of the different mesh sizes meet at one point, but only in the right picture, the area and therefore mass left and right of the intersection point is the same (equal-area rule, see Section 2.3).

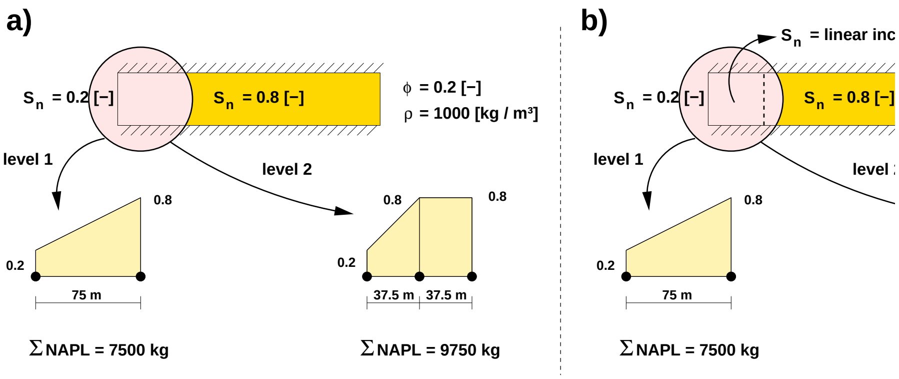


Figure 5.1: Initial conditions and their representation on different meshes

For Figure 5.2.b, the initial conditions are chosen as follows:

$$S_{n\mathrm{init}} = \begin{cases} \frac{0.6}{18.75}x + 0.2 & \text{if } 0\,\mathrm{m} \le x \le 18.75\,\mathrm{m} \\ 0.8 & \text{if } x > 18.75\,\mathrm{m} \end{cases}$$

This represents the initial conditions for a uniform refinement level of 2, where the initial oil saturation is set constant in the domain.

If the solutions with the modified initial conditions are compared to the analytical result, as shown in Figure 5.3(a), the outcome may at first be surprising: the solution with the modified initial condition does not agree with the analytical result. This behavior, however, can be explained with Figure 5.3(b):

For the analytical solution, the domain is initially completely filled with oil, as no segmentation into elements is made. Here, S_n experiences a jump at the boundary from 0.2 to 0.8. The mass distribution of the numerical solution with 'standard' initial conditions converges towards the analytical solution with decreasing element size - the influence of the linear interpolation of the saturation becomes smaller. The mass distribution of the numerical simulation with the 'fixed' initial conditions, however, stays the same for all element sizes, as was anticipated.

Therefore, if one wants to compare the solution to an analytical result, the 'standard' initial conditions can be applied. However, one has to be aware that, in this case, the mass in the system is not the same at the same point in time for different discretization lengths. If one wants to compare the masses for simulation with different mesh widths, one has to adjust the initial conditions accordingly.

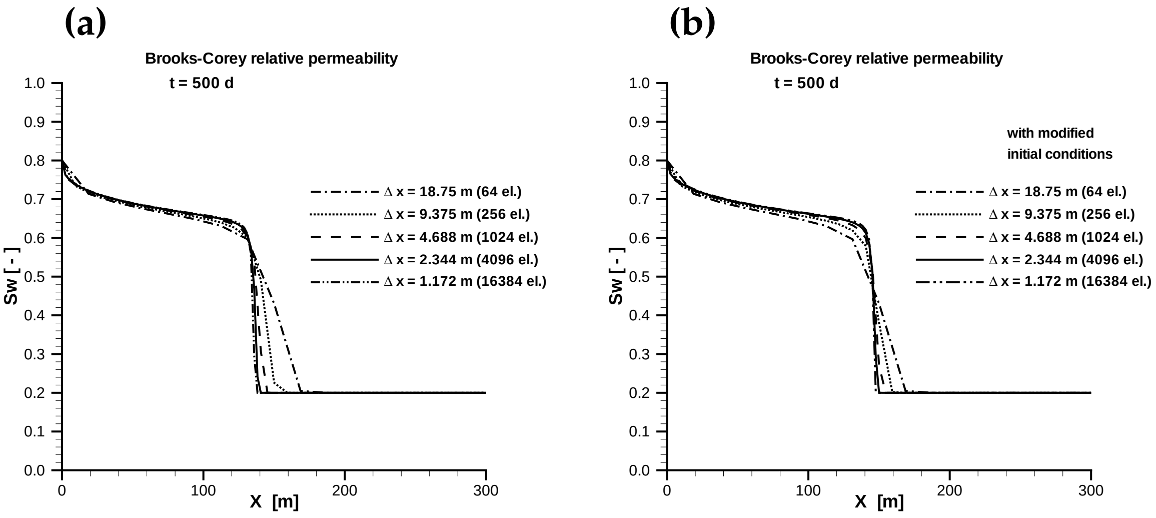


Figure 5.2: *Buckley-Leverett* problem: Comparison of different initial conditions. Left: S_n constant for the inner domain; right: S_n adjusted to the mesh width

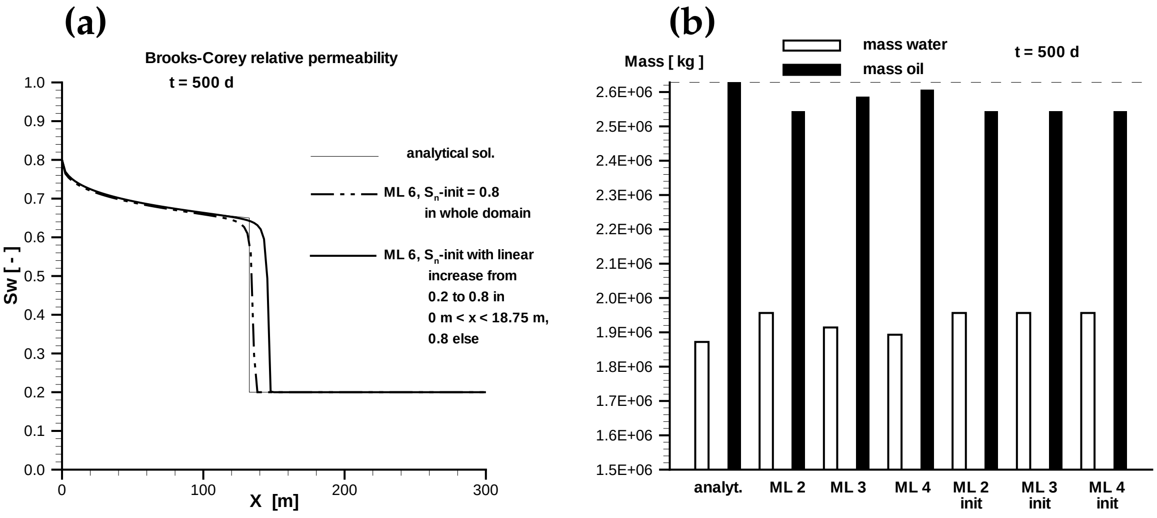


Figure 5.3: *Buckley-Leverett* problem: Comparison with the analytical solution: standard and modified initial conditions

5.2 Mass conservation in the adaptively refined case

When h-adaptive methods are used, two basic cases can occur (either alone or simultaneously):

- new elements/nodes are created
- elements/nodes are omitted

In both cases, the relevant elements are marked by the indicator on the current grid, which is then changed in the next step (see Figure 4.6).

In MUFTE_UG, the grid adaption is a purely geometrical process. This means that no node values are considered when the mesh is adapted. At first, all nodes that are not needed any longer are removed (and with them the information of the solution on these nodes), then the new nodes are inserted. After the grid adaption is completed, new values are interpolated for the newly created nodes.

The solution for the current time step is then calculated again on the now adapted grid. This means that if the transfer of the solution to the new grid results in a small difference of the mass in the system, this is only a different starting point for the Newton algorithm which will then find the correct solution again.

However, in order to compute the solution for a given time step with the backward *Euler* difference scheme, the information of the previous time step is also needed (see Section 2.4.1). This means that the grid has to be adapted for this 'old' solution as well. If the 'old' solution undergoes a small change due to data manipulations, it has an influence on the current time step, as this now finds a solution corresponding to a different problem.

Therefore, it is very important to preserve the mass distribution on a given grid for each point in time. This is illustrated in Figure 5.4. The saturation distribution for the *Buckley-Leverett* problem at t = 500 d is shown for a uniformly refined mesh up to ML 4 and two adaptively refined meshes. In all cases, the smallest element sizes are the same and the initial conditions are chosen appropriately. One can see that, for the case without a correction, the saturation front shows a significant difference to the uniform solution, whereas the curve for the adaptive case with correction basically resembles the uniform case. The reason for the difference of the curves at the foot of the saturation front is caused by the fact that, for the adaptive solutions, the elements in this region already start to become larger again (see Section 6.1).

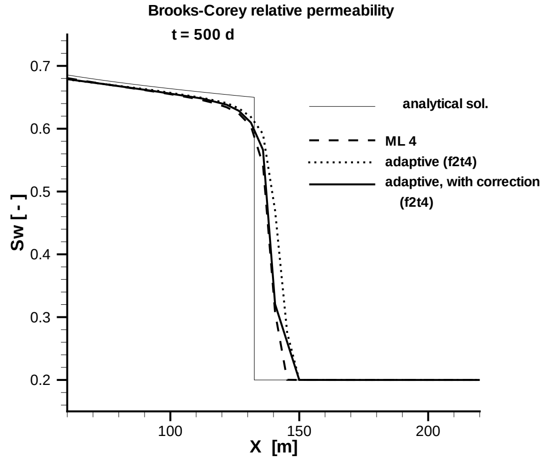


Figure 5.4: S_w (*Buckley Leverett* problem). Uniform and adaptive mesh, without and with correction

In the following, the different cases of creating and omitting nodes and the aspect of mass conservation that is developed in this work will be discussed in further detail.

5.2.1 Insertion of new nodes

Figure 5.5 shows the representation of a mesh in the multigrid structure on which four elements on level 2 are marked for refinement and a new level is created. After the grid adaption, red, green, and yellow refined elements exist on both levels 2 and 3. Note that only red elements can be refined further - the green elements on level 2 first have to be transformed into a red pattern, before sons (here: green and yellow) can be created. Note also that yellow elements are copy elements - this means, that also green elements on level 2 can be copied to level 3.

Figure 5.5 also shows that new nodes not only are created on level 3, but also on level 2. For all new nodes, values are needed. These are obtained by using the (linear) basis functions.

The mass of the non-wetting phase for an element is computed by

$$\sum_{i=0}^{\text{n\_nodes}} S_{n_i} scv_i \; \phi_i \; \rho_{n_i} \qquad \text{with n\_nodes : number of nodes per element} \qquad (5.1)$$

The node value is assumed constant over the subcontrol volume, in compliance with the weighting functions. This implies that not only the value of the saturation, but also the area of the subcontrol volumes plays an important role (see also Figure 5.6).

If node values are now computed for a green refined element, the problem as depicted in Figure 5.7 may occur: Due to the irregular shapes of the subcontrol volumes, the mass in a green refined element is not only different to the mass of the father element - it also depends

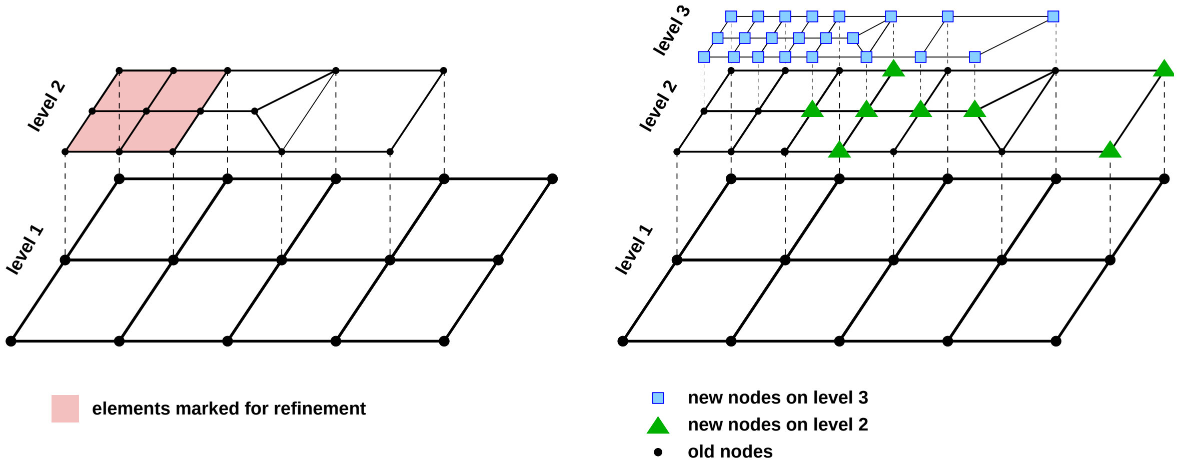


Figure 5.5: Multigrid structure of a refined grid

on the distribution of 'small' and 'large' node values inside the element. (Here, for illustration purposes, the node values were chosen as 4 and 16; the shaded square represents the area of one subcontrol volume of the father element.)

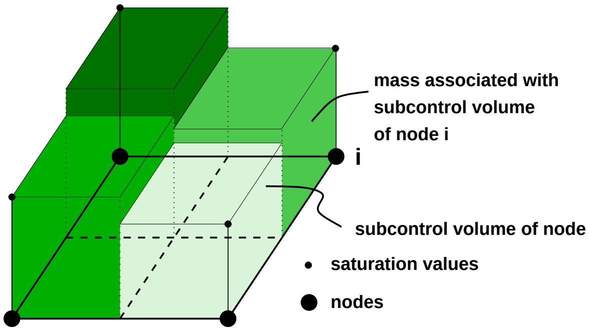


Figure 5.6: Computation of the mass inside an element via the subcontrol volumes

A different problem occurs when green refined brother elements are converted to red refined brother elements, as shown in Figure 5.8. One thing one can note is the difference of mass between an element and all its sons after a time step (left). This is due to the fact that the solution only lives on the topmost elements (leaf elements), where also more nodes are available for information distribution. After the grid adaption, the only information that is left is that of the old nodes. Therefore, it is not possible to retrieve the mass that was stored for the now lost sons. The new node values are then interpolated from the father element. However, as some nodes (here: one) already existed before the grid adaption, their information is also kept. This may lead to an unsymmetric value distribution, as is shown in the right part of the picture. As a result, neither the mass of the 'old' sons nor the mass of the father is now represented by the 'new' sons.

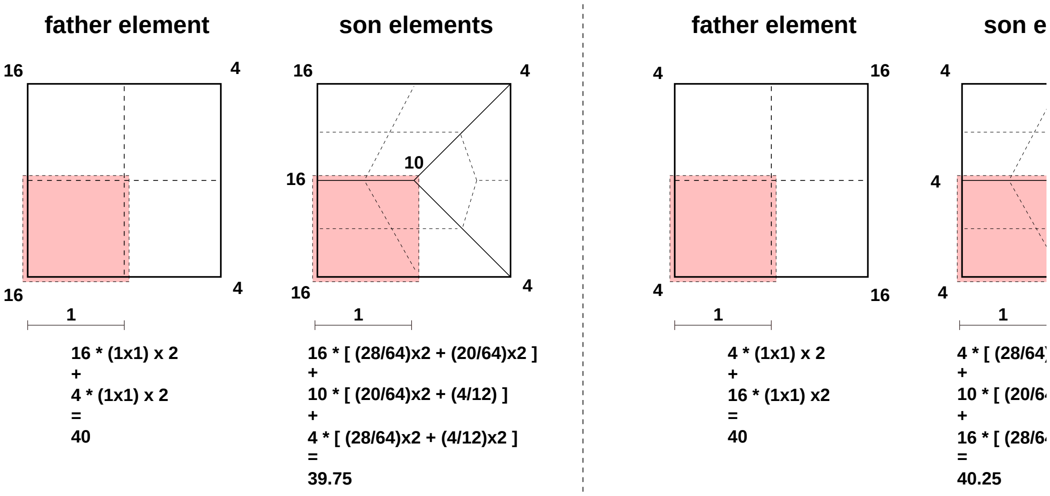


Figure 5.7: Different masses resulting from green refinement

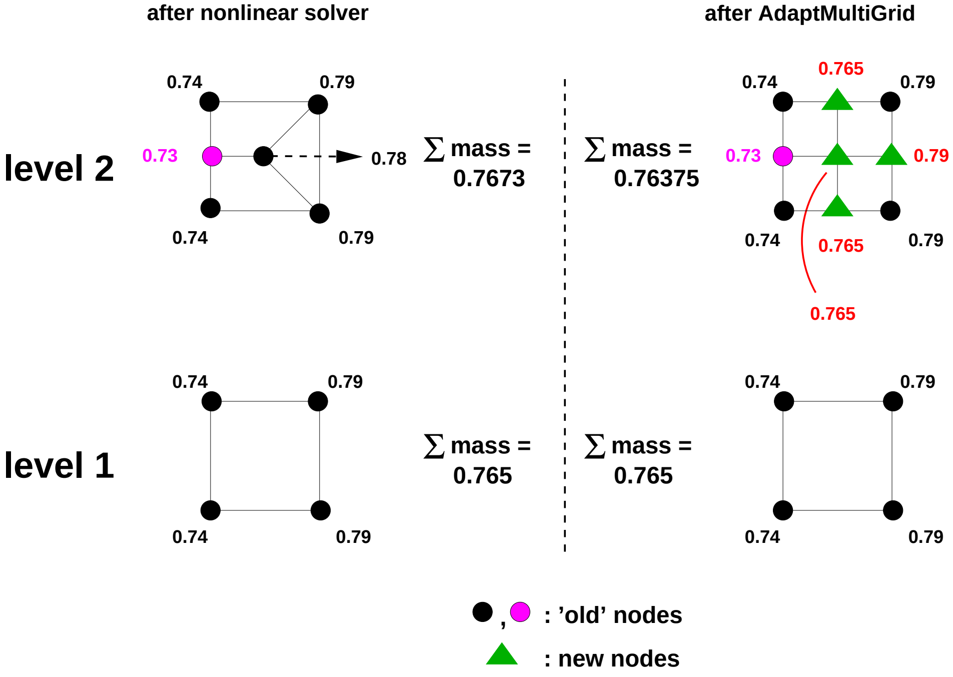


Figure 5.8: Interpolation of new node values from the father element

5.2.2 Coarsening of elements

When elements are coarsened, nodes or whole elements are omitted and thus the solution living on them. The problem of mass conservation for the transition of a red refined element to a yellow element is shown in Figure 5.9. As the corner nodes of the element already existed before the grid adaption, their values are kept. The control volumes (cv's) belonging to the nodes on the adapted grid are larger than on the previous grid. This, however, cannot compensate for the loss of the mass of the middle node, as is shown by comparing the x's on the left and right side of Figure 5.9.

Another aspect can be seen by a look at Figure 5.5 from right to left (all elements on the toplevel are coarsened): Here, it is clearly shown that the whole solution living on level 3 is lost. This results in a mass error that can not be compensated by just solving the non-linear problem again.

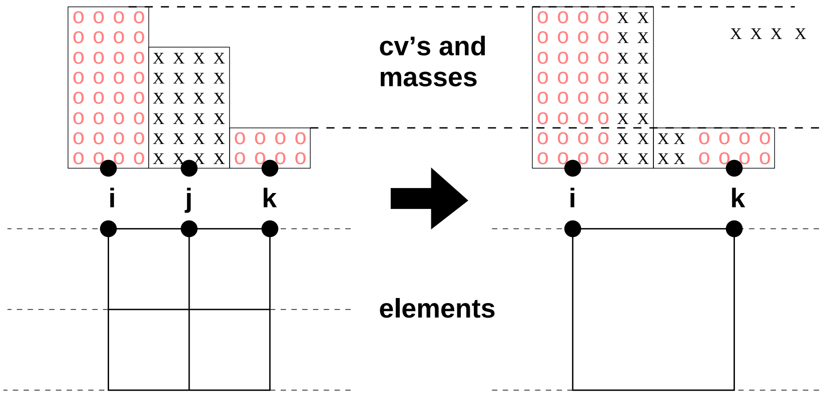


Figure 5.9: Coarsening of an element by omitting nodes

5.2.3 An algorithm ensuring mass conservation

In order to improve the mass conservation for the h-adaptive simulations, different functions were included or changed in the code of UG . The new algorithm for the computation of a time step looks as follows:

```
BDFTimeStep:

    for each level from lowlevel to toplevel
    {
        solve non-linear problem on current level
    }

    if toplevel has been reached
    {
        compute error indicators

        save current mass distribution

        adapt multigrid

        interpolate new node values

        correct mass in elements

        solve non-linear problem again
    }
```

where the bold lines indicate a change. These functions will be explained in the following.

1. Saving the current mass distribution

For each element, a mass associated with it is stored. From the topmost level downward, for every element the decision is made whether it is a leaf element or not. If it is, the mass is stored for this element and added to the entry for its father. If it is not a leaf element, the stored mass for this element is only added to the entry for its father. This method ensures that, even if all sons are omitted by coarsening, the sum of the mass which corresponds to the projected area of the grandfather element is kept. (Grandfather elements are the elements on the bottom level which cannot be coarsened any further). This is also illustrated in Figure 5.10.

2. Interpolation of new node values

New node values are interpolated using the linear basis functions. For the center node in an element, however, an exception is made (see Figure 5.11). Again, the node values are chosen as 4 and 16 for illustration purposes. Here, the difference between the mass associated with the father element (in this case: 45) and the mass associated with all son nodes except the center node is calculated. This value divided by the area of the control volume of the center node then yields the new node value. In order to keep the corresponding problem physically correct and to avoid complications with the Newton solver, in should be ensured that the new node saturation values do not exceed 1. This constraint, however, is not yet implemented.

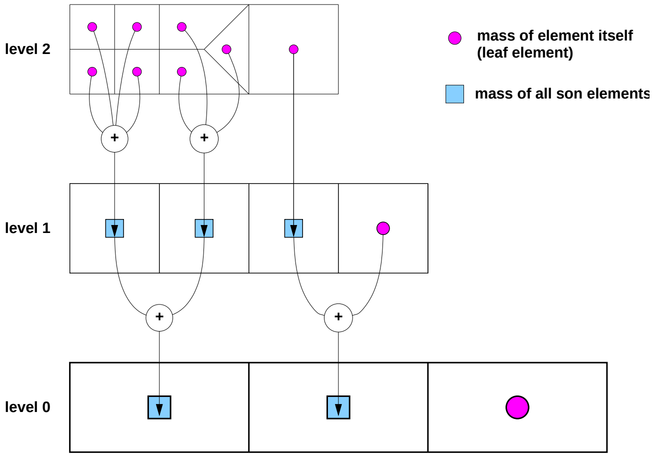


Figure 5.10: Projection of the masses to the father elements

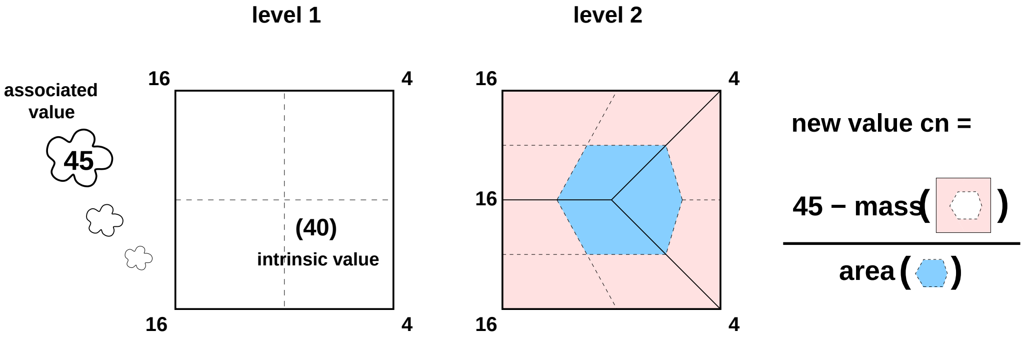


Figure 5.11: Calculation of the node value for the center node (cn)

3. Correction of the mass in an element

For all elements with a new center node, the mass is now correct. For elements with no center node, however, the mass still has to be corrected in some cases. So for all these elements, the current mass of all brother elements is compared with the reference value kept by the father. The difference is then put into proportion to the sum of the control volumes of all son nodes (see Figure 5.12) and a share for each node is computed. After this has been done for all elements and all the shares for the nodes have been summed, they are then added to the respective nodes in a final loop.

The effect of the implemented mass correction can be seen in Figures 5.13 and 5.14. Fig. 5.13 shows the evolution of the mass of the non-wetting phase over time for the *Buckley-Leverett* problem (see also Figure 5.4). In Fig. 5.14 the difference of mass in comparison to the analytical solution is shown. At the start of the simulation, the masses for all numerical simulations are the same. They differ, however, from the initial mass for the analytical solution, as was explained in Section 5.1.

As the applied discretization scheme is mass conservative, the difference to the analytical solution stays the same for the simulation with the uniformly refined mesh (represented by the dotted line). For the 'standard' adaptive solution (solid line), however, the mass decrease over time is **larger** than the mass decrease due to the outflow condition. Figure 5.14 shows nicely, that the process of mass decrease happens in 'jumps': every time the mesh is adapted (refined or coarsened), a change in mass occurs. In contrast to this behavior, the mass content of the adaptive simulation with correction (dashed line) resembles the mass content of the uniform simulation: the difference to the analytical result stays constant.

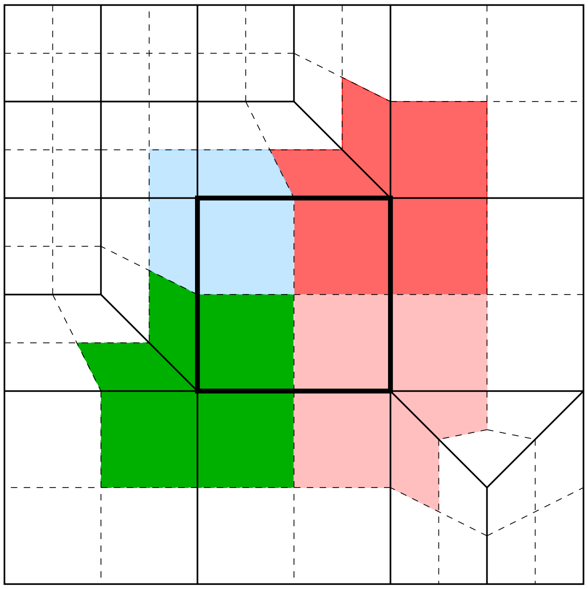


Figure 5.12: Control volumes for all nodes in an element. The solid lines indicate element edges, the dashed lines represent subcontrol volume faces.

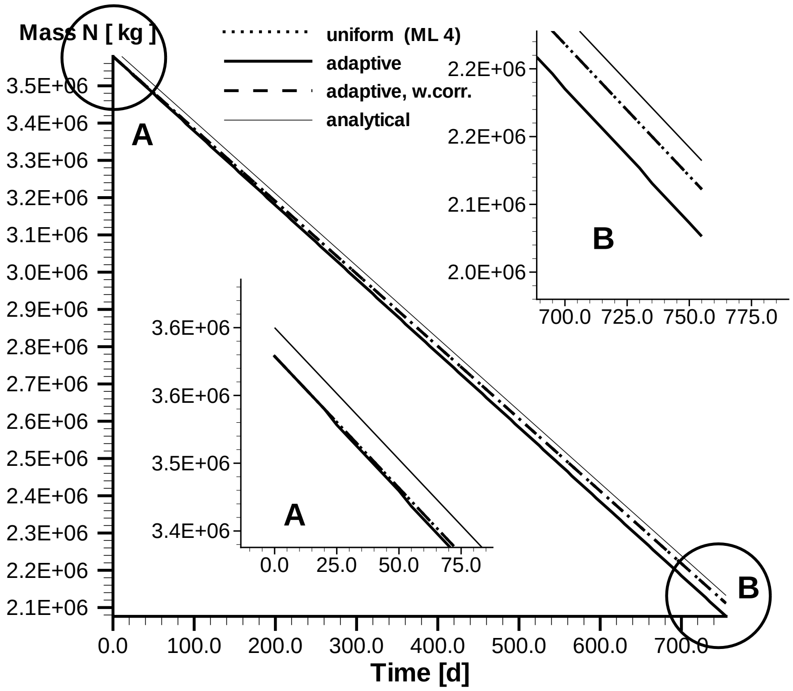


Figure 5.13: Evolution of the mass over time for the *Buckley-Leverett* problem

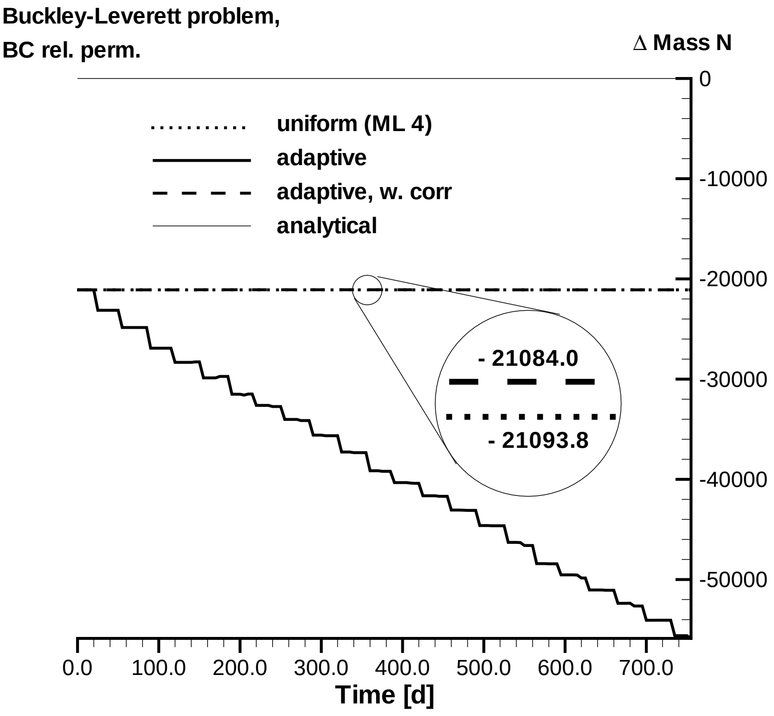


Figure 5.14: Difference of mass over time for the *Buckley-Leverett* problem

6 Adaptive simulation of the homogeneous test cases

In the following sections, the test cases introduced in Chapter 3 are simulated with the adaptive methods described in the previous chapters. The aim is to identify a suitable error indicator for advection-dominant and diffusion-dominant flows as well as for a typical two-phase flow problem, where both effects mentioned are present simultaneously. The chosen error indicator will then be applied to a heterogeneous system in Chapter 8. For the homogenous test cases, the performace of the adaptive simulation is compared to the uniform simulation w.r.t. execution time, number of elements and nodes needed and the amount of non-linear and linear iterations required for each time step.

6.1 *Buckley-Leverett* problem

As already pointed out in Chapter 4, in order to use adaptive methods, one has not only to choose an appropriate indicator (or estimator), but also to give the margins for refinement and coarsening.

For the basic test case of the *Buckley-Leverett* problem with a (uniform) discretization length of Δx = 9.375 m and a time-step of 5 days, the values of several indicators are calculated. Figure 6.1 shows the values for the indicators *grad_HpS*, *grad_HS*, and *grad_S* (see Section 4.2).

Figure 6.1 shows that

- different problems (e.g. linear - non-linear relative permeability) demand different refinement and coarsening criteria, and
- different indicators require different refinement and coarsening criteria.

In most engineering applications, however, it is not feasible to invest a large amount of time in finding the best indicator and the best range of tolerances for each individual problem. The goal of this work is to find an indicator which works reasonably well for a large number of problems.

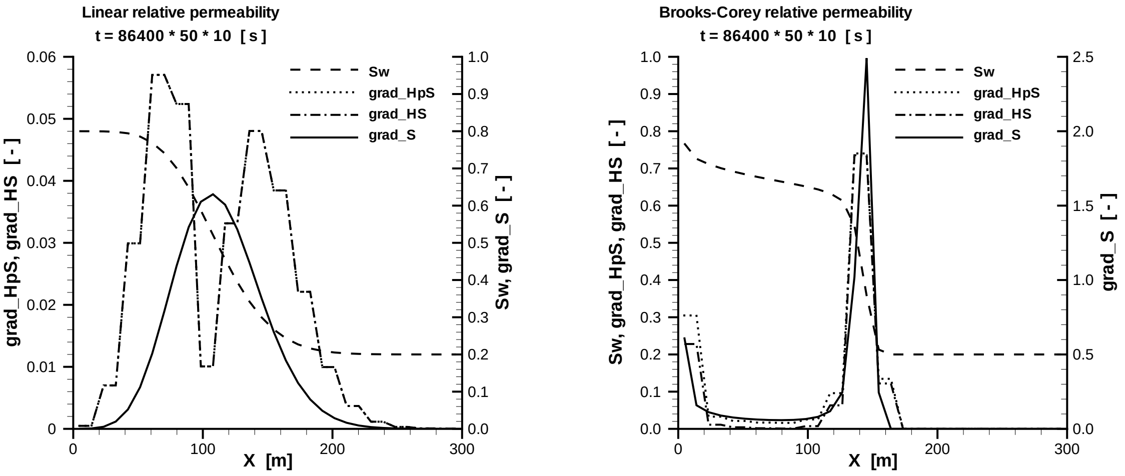


Figure 6.1: Comparison of the values of three different gradient indicators (*grad_HpS*, *grad_HS*, and *grad_S*), computed on a uniform mesh (ML 3) (Left: linear; right: BC)

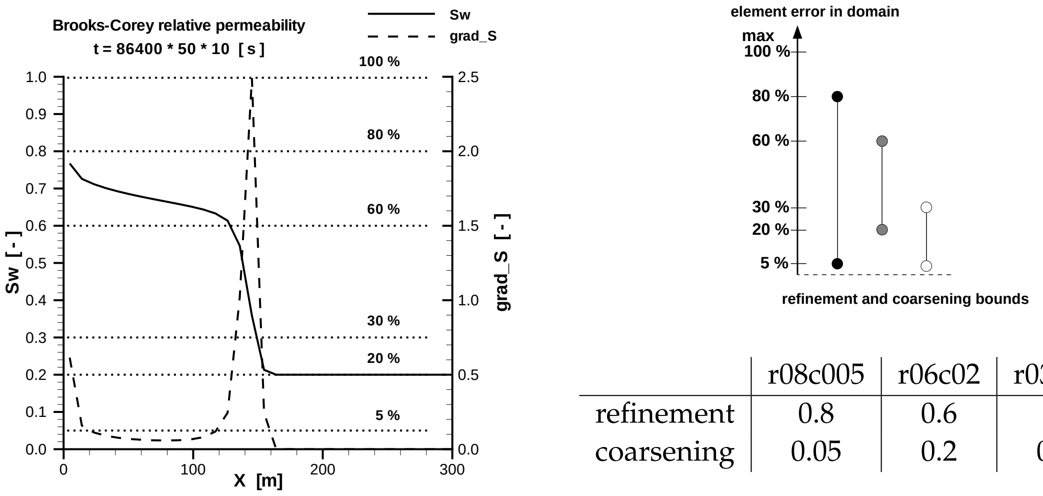
6.1.1 Preliminary investigations

In order to choose a suitable indicator as well as refinement and coarsening bounds (rc-criteria), preliminary investigations are carried out. This is done for an adaptive mesh, where the elements, starting with Δx = 9.375 m (ML 3), are allowed to be refined or coarsened once. Therefore, the element sizes range from Δx = 4.8675 m (ML 2) up to Δx = 18.75 m (ML 4).

At first, three different combinations of regions of rc-criteria are chosen according to the indicator distribution for the relative permeability after *Brooks-Corey* (see Figure 6.2). In the following, a rc-criterium will be indicated by ′r*n*c*n*′, where r stands for refinement bound, c for coarsening bound and *n* denotes the percentage related to the maximum error value (e.g. ′c005′ stands for ′coarsening bound: max. error value times 0.05′ → 5 %).

The results of the adaptive simulations are exemplarily shown for the *grad_S*-indicator in Figure 6.3 with the corresponding meshes depicted in Figure 6.4. As can be seen, the results for the three different rc-criteria are very similar. This behavior is also observed for the *grad_HS*- and *grad_HpS*-indicators (not shown here).

For a numerical simulation, it is desirable to mediate between a sufficient large number of elements to resolve critical regions and a sufficient small number of elements to minimize the computational cost. Therefore (if not otherwise stated), the rc-criterion r08c005 is chosen in the the further investigations.



	r08c005	r06c02	r03c005
refinement	0.8	0.6	0.3
coarsening	0.05	0.2	0.05

Figure 6.2: *Buckley-Leverett* problem: Three different refinement and coarsening tolerances

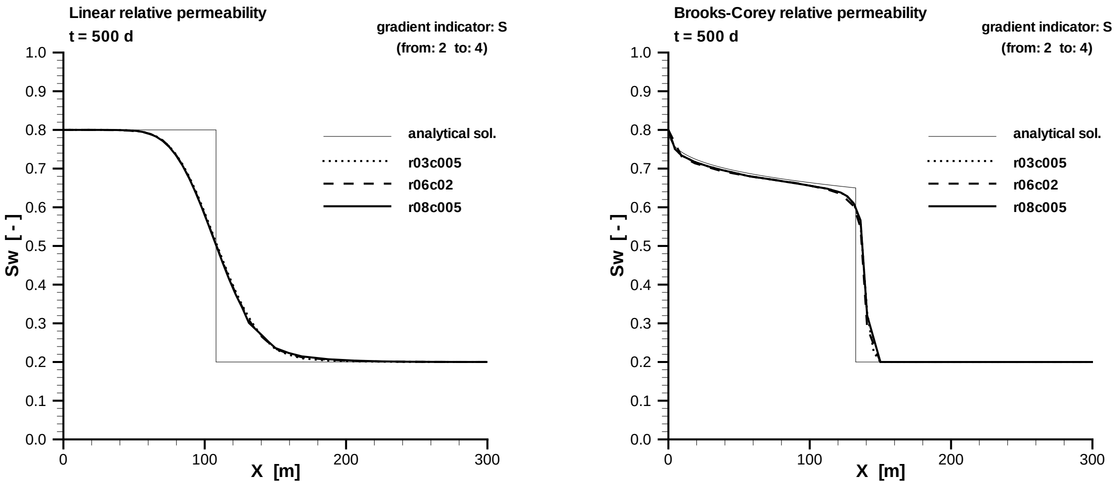


Figure 6.3: Comparison of three different rc-criteria for the *grad_S*-indicator (Left: linear; right: BC)

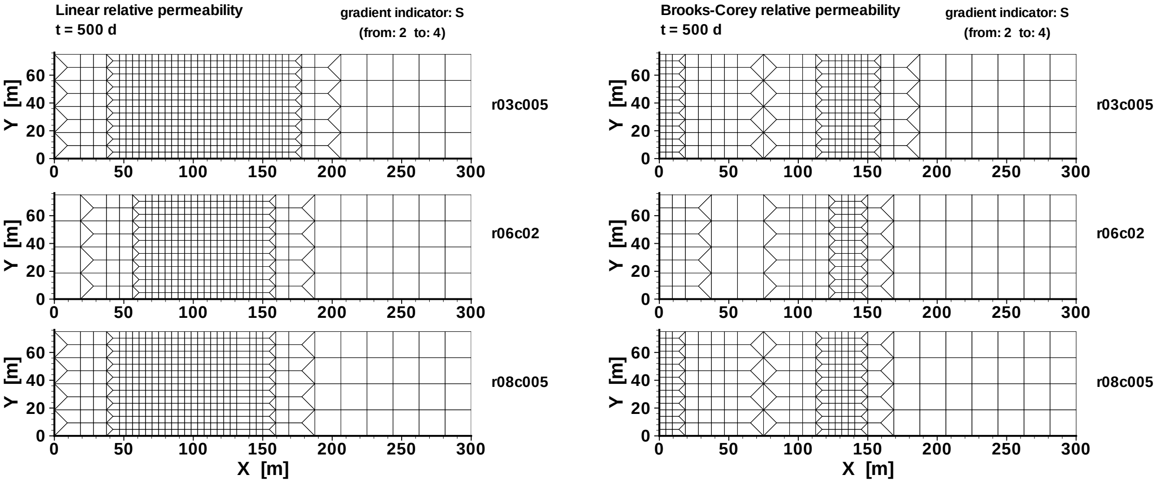


Figure 6.4: Comparison of the meshes for three different rc-criteria for the *grad_S*-indicator (Left: linear; right: BC)

6.1.2 Comparison uniform - adaptive simulation

In Figure 6.5, the results for three different indicators are shown in comparison to the analytical solution and the solutions for ML 4. As anticipated, the adaptive results tend towards the analytical solution and show hardly any difference compared to the uniform result. Also, the difference between the results for the three indicators is marginal. The corresponding meshes are depicted in Figure 6.6. In the further investigations, only the *grad_S* indicator is considered.

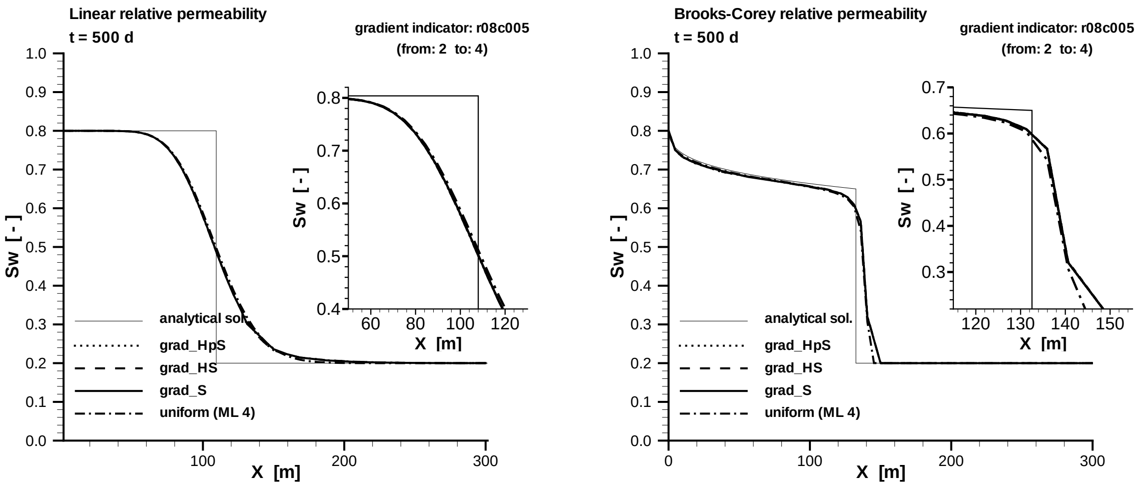


Figure 6.5: Comparison of the three different gradient indicators for refinement: 0.8, coarsening: 0.05 (Left: linear; right: BC)

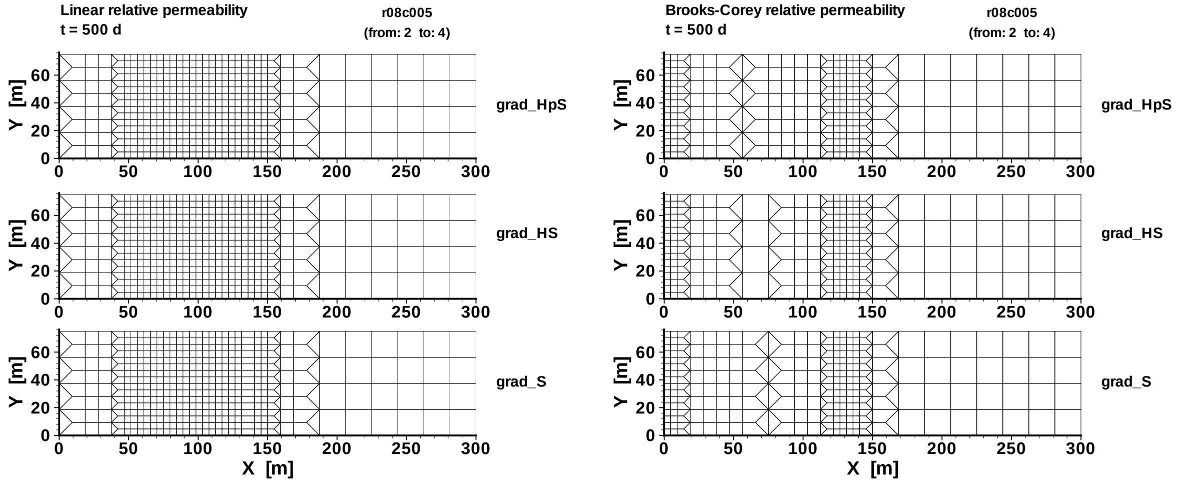


Figure 6.6: Comparison of the meshes for three different indicators (Left: linear; right: BC)

The following table shows a comparison of the execution time and the number of elements and nodes for the uniform and adaptive case (relative permeabilities after *Brooks-Corey*) for meshes with the same smallest element size (ML 4 and an adaptive refinement from level 2 to level 4). For the adaptive cases, the averaged element number over the whole simulation time is given. As the number of elements changes throughout the simulation, no absolute number can be given. The values in brackets represent the ratio between the adaptive and the uniform requirements.

	Linear		*Brooks-Corey*	
	uniform	adaptive	uniform	adaptive
execution time [s]	49.5	25.8 (0.52)	275.1	101.9 (0.37)
elements	1024	349.5 (0.34)	1024	265.5 (0.26)
nodes	1105	415.4 (0.38)	1105	309.7 (0.28)

In order to reach the same precision for the relative permeability after *Brooks-Corey* as a mesh with a uniform refinement (level 4), the adaptive algorithms require approximately one third of the number of elements and computational time for this example. A similar behavior can be observed for the linear relative permeability. One has to note, that the here presented problem with approx. 1000 elements is still comparatively small.

In Figure 6.7, some comparisons between the uniform and adaptive simulation are shown. Here, the number of non-linear iterations (Newton iterations, see Section 2.4.3) for each time step and the averaged number of linear iterations per time step are depicted. The latter represents the mean value of linear iterations per one Newton step in one time step. For the uniform case, the number of non-linear iterations ranges between 3 and 4 in the non-linear case and even stays constant in the linear case. If, however, the grid is adaptively changed, the number of non-linear iterations rises significantly.

An interpretation of these results is that, due to the presence of different element sizes, more non-linearities are introduced into the system, which then lead to a significant increase of the numerical effort. Only due to the smaller element number does the overall execution time (in most cases) stay below the time for the uniform simulation.

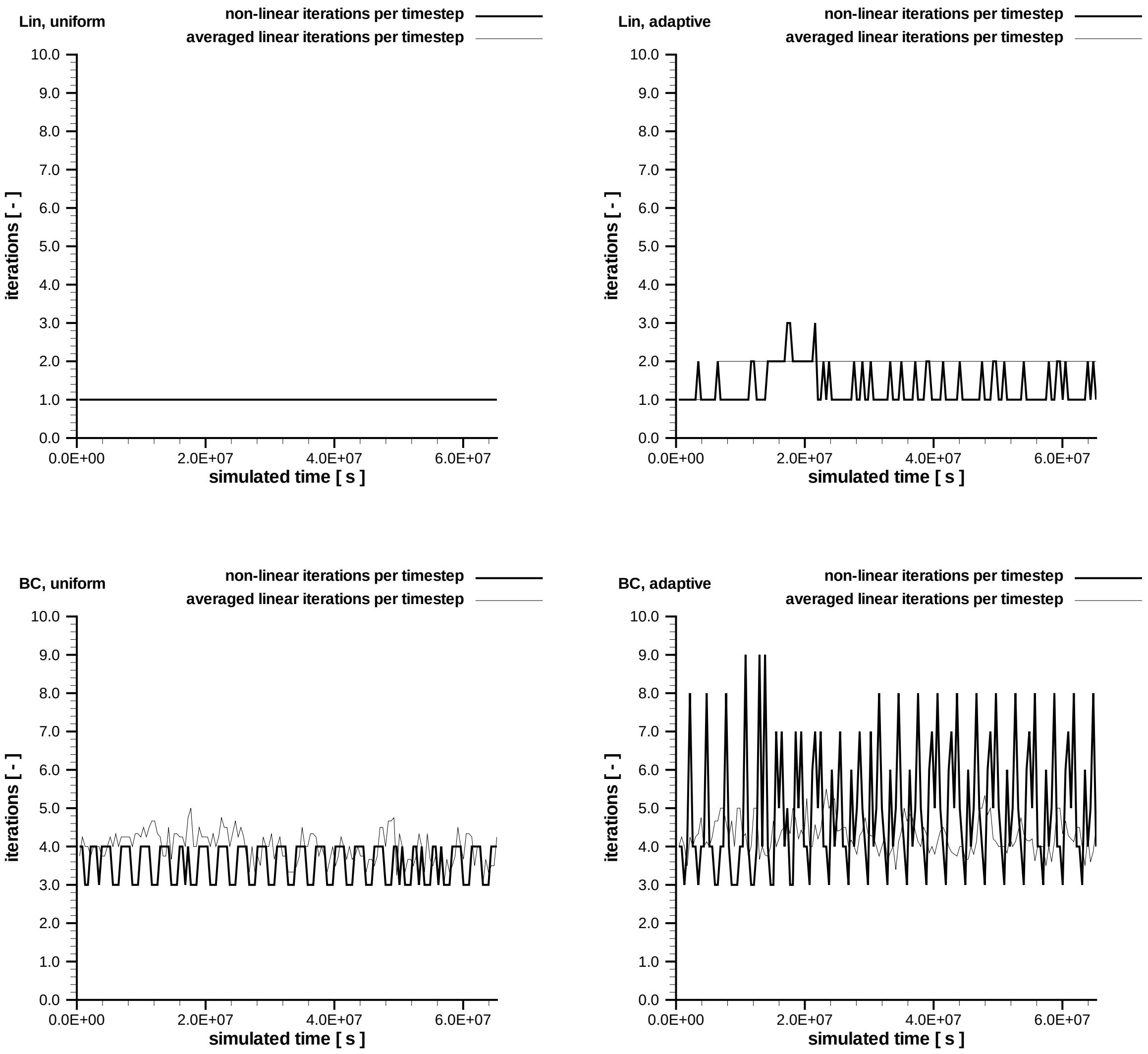


Figure 6.7: Comparison of the number of non-linear iterations and averaged linear iterations per time step for linear relative permeability and after *Brooks-Corey*, uniform and adaptive case

6.2 *McWhorter* problem

For an adaptive refinement, an appropriate error indicator first has to be chosen. The values for different indicators already described in Section 4.2 together with the water saturation and the error are shown in Figure 6.8. As can be seen, the gradient indicator for the saturation (*grad_S*) locates the areas with a large error very well. It is therefore chosen for the following simulations. In Figure 6.9, the chosen bounds for this indicator for the refinement and coarsening are shown: 'refinement bound = 0.2' and 'coarsening bound = 0.1'.

The result of the adaptive computation is compared to the uniform solution in Figure 6.10. The mesh is adaptively refined using a 'blue' strategy. This is possible, because the *McWhorter* problem resembles a quasi-one-dimensional case. The two results are shown for different points in time - this is due to the fact that the time step is automatically reduced in the course of the adaptive computation (see Section 2.4.3). Therefore, the exact location of the saturation front cannot be compared. However, as can be seen, another effect can be observed: the error for the adaptive simulation becomes larger than for the uniform computation. This is clearly an effect which is not desired.

The following table compares the execution time and the elements and nodes of a uniform and an adaptive simulation. For all cases, the size of the smallest element is the same (ML 4 for uniform computation, refinement from level 2 to level 4 for the adaptive case). As already stated in Section 6.1.2, no fixed value for the number of elements for the adaptive simulation is given, but an average over the simulation time. The numbers in brackets show the relationship between uniform and adaptive results. As can be seen, although the number of degrees of freedom (represented by the nodes) is approximately halved, the decrease in execution time is only minimal.

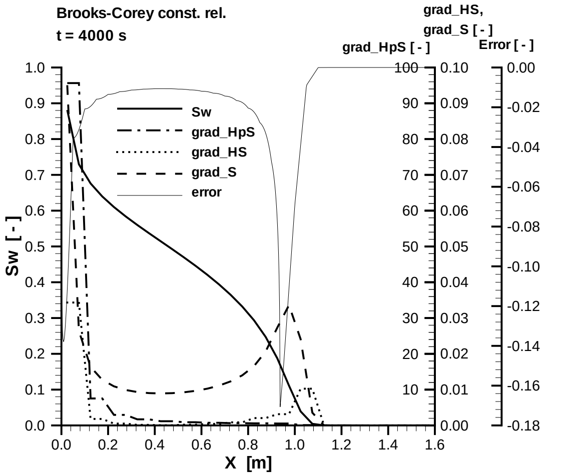


Figure 6.8: Water saturation S_w and indicator distribution for three indicators

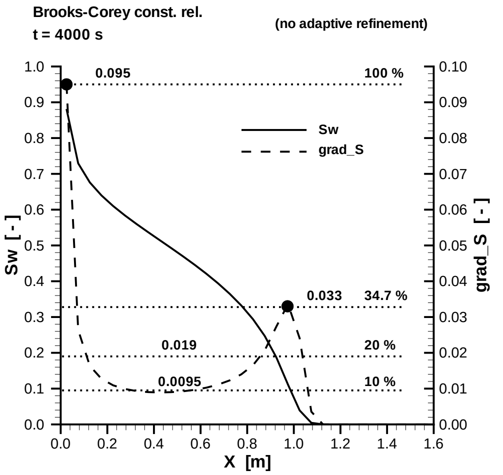


Figure 6.9: Refinement and coarsening bounds for the *grad_S* indicator

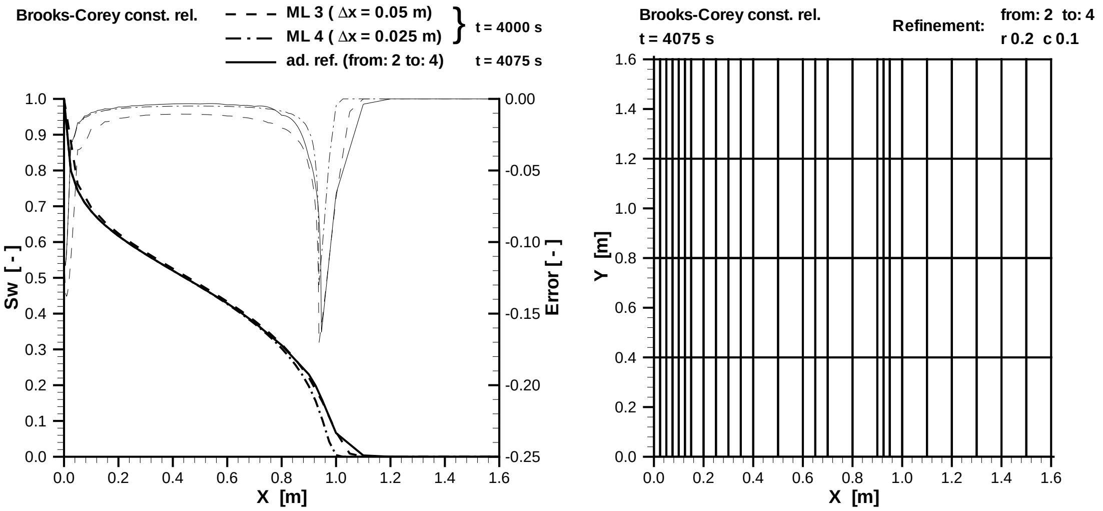


Figure 6.10: Water saturation S_w (thick lines) and error distribution (thin lines) for uniform and adaptive refinement (left) and corresponding adaptive mesh (right)

	Linear		*Brooks-Corey*	
	uniform	adaptive	uniform	adaptive
execution time [s]	21.9	19.2 (0.88)	36.8	32.7 (0.89)
elements	256	124.6 (0.49)	256	115.1 (0.45)
nodes	325	167.7 (0.52)	325	164.6 (0.51)

Further effects of the adaptivity can be observed in Figures 6.11 to 6.13:

First, the course of the execution time over the simulated time is shown in Figure 6.11. Here, the curves for the adaptive simulations are much more uneven than for the uniform cases. This behavior can be explained by Figure 6.12: Whereas the time step Δt stays constant for the uniform simulation, it is automatically changed (decreased) for the adaptive simulations. This effect is greater if more non-linearities are present, as is the case for the *Brooks-Corey* constitutive relationships.

The influence of the non-linearities can be seen clearly in Figure 6.13. Here, the number of non-linear iterations for each time step is shown. Also, the averaged number of linear steps for each non-linear step is depicted. For the case with a uniform mesh and linear constitutive relations, both these numbers stay constant over the whole simulation. If, however, the mesh is adapted, the course of the numbers of non-linear iterations shows some peaks, whereas the number of linear iterations stays nearly the same. This shows that the linearized system of equations is 'good natured' and the problems (and the decrease of the time step) stem from the non-linearities. This becomes even clearer when looking at the case of non-linear (*Brooks-Corey*) constitutive relationships: for the uniform case, the number of iterations fluctuates slightly more than for the linear case, and for the adaptive case this effect is even worse.

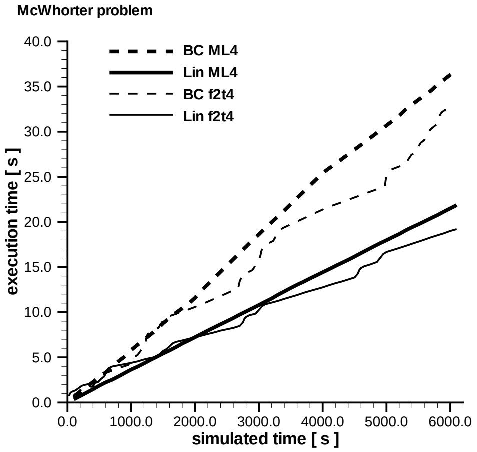


Figure 6.11: Execution time over simulated time for linear constitutive relationships and after *Brooks-Corey*, uniform and adaptive case

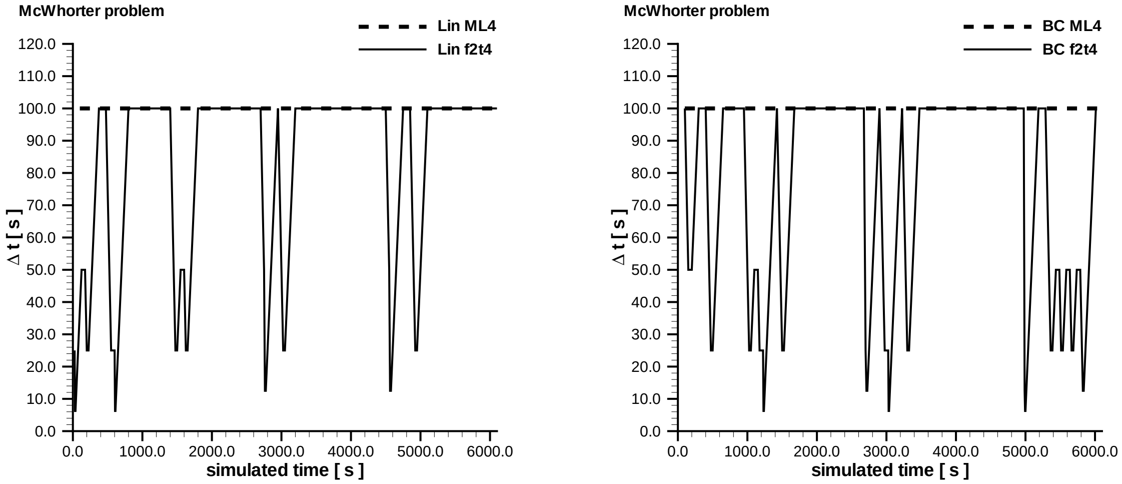


Figure 6.12: Comparison of the time step Δt for linear constitutive relationships and after *Brooks-Corey*, uniform and adaptive case

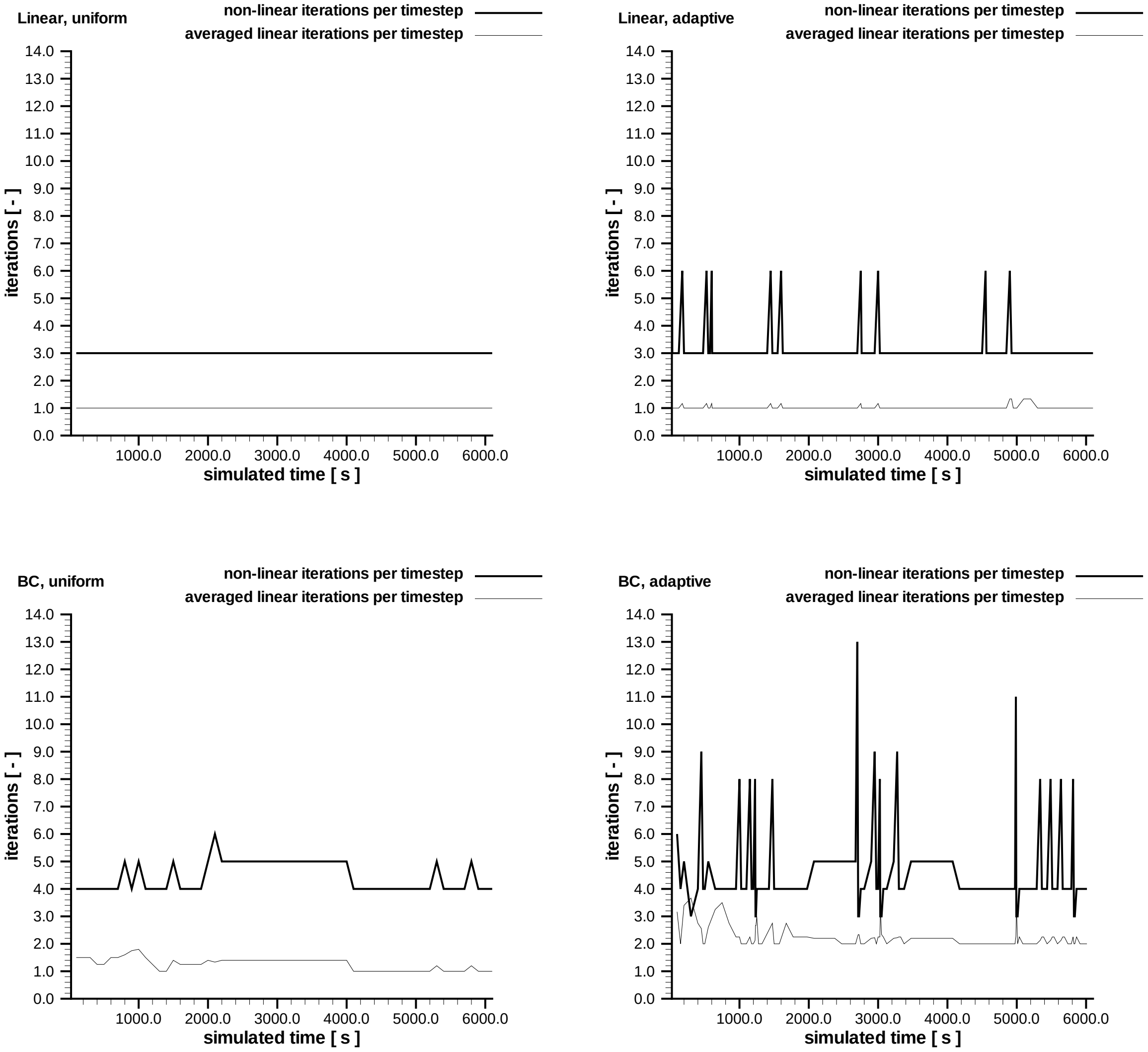


Figure 6.13: *McWhorter* problem: Comparison of the number of non-linear iterations and averaged linear iterations per time step for linear constitutive relationships and after *Brooks-Corey*, uniform and adaptive case

6.3 Homogeneous model problem (Sandbox)

In order to choose suitable bounds for the refinement and coarsening criteria, a simulation is first carried out on a uniform mesh and the indicator value for the *grad_S* indicator computed. The result can be seen in Figure 6.14. The distribution will be quantitatively different when an adaptive mesh is turned to, but the qualitative distribution of the high indicator values will remain the same. It can be observed that the indicator locates the front of the non-wetting phase very well. One difficulty, however, arises: the indicator values show a peak for two locations along the north boundary. As the refinement and coarsening bounds are computed relative to this maximum value, the bounds have to be chosen carefully in order to still localize the front.

For the adaptive simulation, the refinement bound is chosen as 0.6 and the coarsening bound as 0.2. The mesh is allowed to coarsen down to level 1 (Δx = 0.5 m) and is refined up to level 3 (Δx = 0.125 m). Figure 6.15 shows the oil saturation distribution and the mesh after 24 hours. The adaptive refinement follows the proceeding front very nicely. Behind the front, the elements are coarsened again. A comparison of the execution time for the adaptive and the uniform case is given in Figure 6.16. Here, the adaptive simulation clearly shows an improvement. As for the two previous test cases, again the relations between number on non-linear iterations and averaged linear iterations are shown in Figure 6.17. Again, the adapted problem seems to introduce more non-linearities and therefore more computational effort is needed. The performance of the Newton algorithm, however, is still good enough: the time step is not reduced (not shown here).

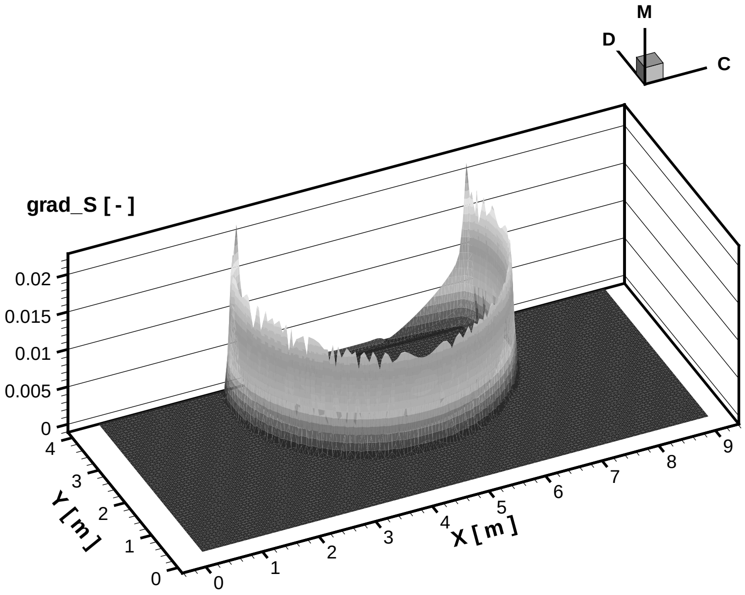


Figure 6.14: Distribution of the indicator values for indicator *grad_S* (Sandbox problem)

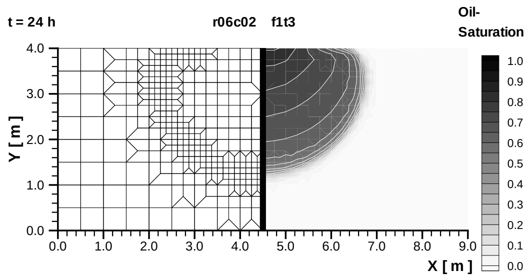


Figure 6.15: Mesh and oil saturation distribution after 24 hours

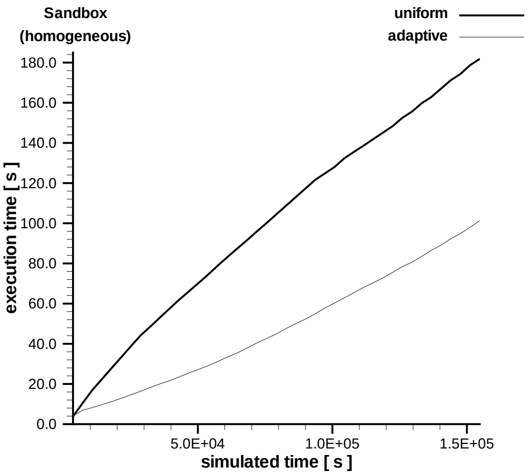


Figure 6.16: Execution time over simulated time for the uniform and adaptive case

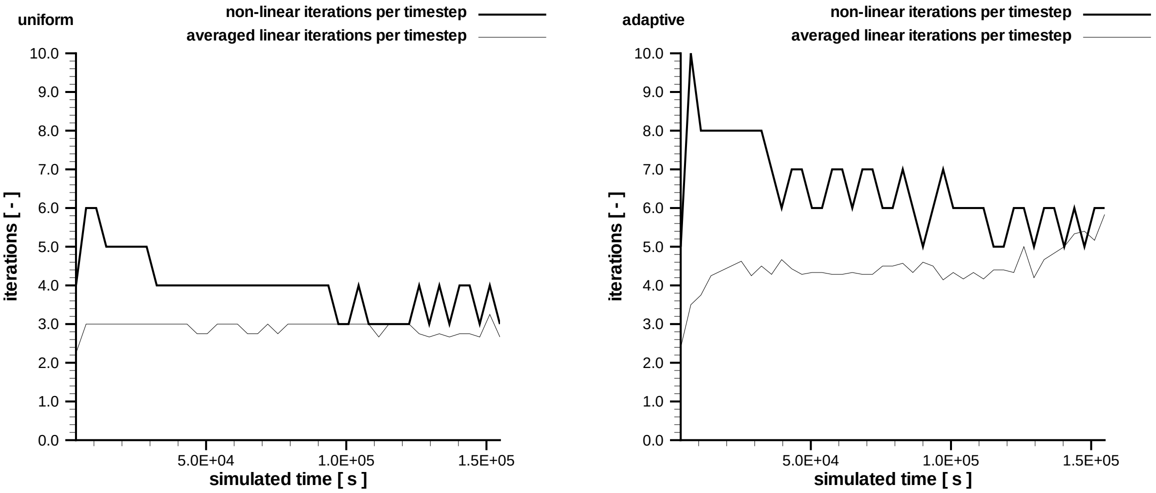


Figure 6.17: Comparison of the number of non-linear iterations and averaged linear iterations per time step, uniform and adaptive case

6.4 Conclusion

In this chapter, the homogeneous test cases introduced in Chapter 3 were simulated with adaptive methods. Of the three error indicators described in Section 4.2, the *grad_S* indicator was chosen for all three test cases, as it located the error between the numerical simulation and the analytical solution for the *Buckley Leverett* and *McWhorter* problem very well.

A general recommendation for suitable refinement and coarsening criteria cannot be given for the reasons stated in Section 6.1: each problem requires a unique set of refinement and coarsening criteria. For the *Buckley Leverett* problem, a comparison between three different criteria is shown.

For all three test cases, the non-linear and linear iterations per time step were compared for the uniform and the adaptive simulation. Here, the clear statement can be given that h-adaptive methods increase the numerical complexity of the problems because of different element sizes in the domain. However, h-adaptive methods remain a powerful tool for obtaining an accuracy of solutions which would hardly be attainable for uniform refinement because of the sheer amount of computational storage needed for the otherwise huge number of elements and nodes.

The aim of the adaptive methods presented here is on the one hand to provide a tool for an efficient solution technique. On the other hand, the focus lies on the representation and tracking of sharp fronts in the system. As these always pose numerical difficulties, a special treatment is needed. This is accounted for in this work by applying the fully upwind method for advection-dominated problems. The fully upwind strategy is of first order and introduces artificial diffusion into the system (see e.g. *Helmig* [31]). However, it is generally desirable to use methods of higher order. This could be achieved with small effort by using a centrally weighted mobility, resulting in a method of second order, as was explained in Section 2.4.2.1 - yet, this procedure is not stable and shows a non-monotonous behavior when confronted with sharp fronts.

Also, fully upwinding introduces a dependence on the flow direction due to its nature. This, however, is not favorable when regarding diffusive processes. Therefore, the physical behavior can be represented better when a centrally weighted mobility is used in these cases.

Therefore, in the following chapter, a criterion is to be developed which identifies regions where advection or diffusion is dominant. The aim is then to utilize fully upwinding only in the regions where the hyperbolic part of the two-phase flow equations has a strong influence.

When the diffusion-dominated regions are identified in the system and regarded with a second-order scheme, the affected elements can be increased in size again. This has the advantage that the physical process is still well enough represented by an overall smaller number of degrees of freedom in the system.

7 Simulation of the homogeneous test cases with the *Peclet* criterion

7.1 Derivation of the two-phase *Peclet* number

As explained in the previous section, a criterion which identifies regions of advective of diffusive dominance is the aim. A measure for the relation of these properties is the dimensionless *Peclet* number (*Pe*).

Originally derived for the classical transport equations, the *Peclet* number is defined in general as follows:

$$Pe = \frac{\mathbf{v} \cdot \Delta l}{\mathbf{D}} \left(= \frac{\text{advection}}{\text{diffusion}} \right) \qquad \begin{array}{l} |Pe| \ll 1 \rightarrow \text{diffusion dominates} \\ |Pe| \gg 1 \rightarrow \text{advection dominates} \end{array} \tag{7.1}$$

In this classical definition, **v** represents the velocity in the system and **D** is a general representation of the diffusion effects. If *Pe* is regarded with respect to a discretization mesh, the characteristic length Δl represents the element length (*Helmig* [31]).

For the pressure - saturation formulation of the two-phase flow equations applied here, it is not possible to distinguish exactly between an advective and a diffusive term. Therefore, a different approach is chosen:

The advective and diffusive properties of the $p_w - S_n$-formulation (Eq. (2.40)) are combined in the flux term $\mathbf{F}_\alpha = \rho_\alpha \mathbf{v}_\alpha$ (see Eq. (2.45)).

As explained in Section 2.3, the influence of the advective (hyperbolic) and diffusive (parabolic) effects are steered by the saturation (low saturation: diffusion dominates, high saturation: advection dominates). Therefore, a variation of the flux term (and the velocity in particular, as the density of the fluids is assumed to be constant) with respect to the saturation should result in an expression which describes the influence on diffusion and advection.

Since a $p_w - S_n$-formulation is considered in this work, the following investigations are carried out for the velocity of the non-wetting phase $\mathbf{v}_n$. As the applied discretization scheme is the Box Method (see Section 2.4.2), the points of interest for the variation are the integration points (IPs): The computation of the stiffness matrix is done for each element via a loop over the integration points in an element. Therefore, the *Peclet* number for an element is also considered at the IPs.

$\mathbf{v}_n$ is evaluated at the integration points in the following way:

$$\mathbf{v}_n^{IP} = -\lambda_n^{IP}\,\mathbf{K}\left(\mathbf{grad}\,p_n^{IP} - \rho_n^{IP}\mathbf{g}\right) \tag{7.2}$$

It represents the flow from a node i to a node j across a subcontrol volume face into a box (see Fig. 7.1). In the following, all variations are done with respect to the upstream node i.

Formulating the derivative of the flow velocity with respect to the saturation at node i ($S_{n|i}$) yields the following expression:

$$\frac{\partial \mathbf{v}_n^{IP}}{\partial S_{n|i}} = -\frac{\partial \lambda_n^{IP}}{\partial S_{n|i}}\mathbf{K}\left(\mathbf{grad}\,p_n^{IP} - \rho_n^{IP}\mathbf{g}\right) - \lambda_n^{IP}\,\mathbf{K}\,\frac{\partial\left(\mathbf{grad}\,p_n^{IP} - \rho_n^{IP}\mathbf{g}\right)}{\partial S_{n|i}} \tag{7.3}$$

The gradient of p_n can be expressed as $\mathbf{grad}\,p_n = \mathbf{grad}\,p_w + \mathbf{grad}\,p_c$. As $\mathbf{grad}\,p_w$ as well as $\rho_n\mathbf{g}$ are independent of S_n, Eq. (7.3) can also be written as follows:

$$\frac{\partial \mathbf{v}_n^{IP}}{\partial S_{n|i}} = -\frac{\partial \lambda_n^{IP}}{\partial S_{n|i}}\mathbf{K}\left(\mathbf{grad}\,p_n^{IP} - \rho_n^{IP}\mathbf{g}\right) - \lambda_n^{IP}\,\mathbf{K}\,\frac{\partial\,\mathbf{grad}\,p_c^{IP}}{\partial S_{n|i}} \tag{7.4}$$

In Eq. (7.4), the first term can be judged as the advective part and the second term as the diffusive part. If we put these terms in relation to each other, the two-phase *Peclet* number can be obtained.

But first, some further remarks have to be made:

- In 2D and 3D, the gradient represents a vector. For the discretization used here, only the flow normal to the subcontrol volume face is considered. This can be obtained by multiplying all the vector-type variables by the (unit) normal vector of the subcontrol volume face.
- $\mathbf{grad}\,p_c$ and λ_n are interpolated to the integration point by the shape functions in the element. In this way, the mobility is always weighted centrally, whether upwinding is used or not. This ensures that the conditions for all elements are the same.

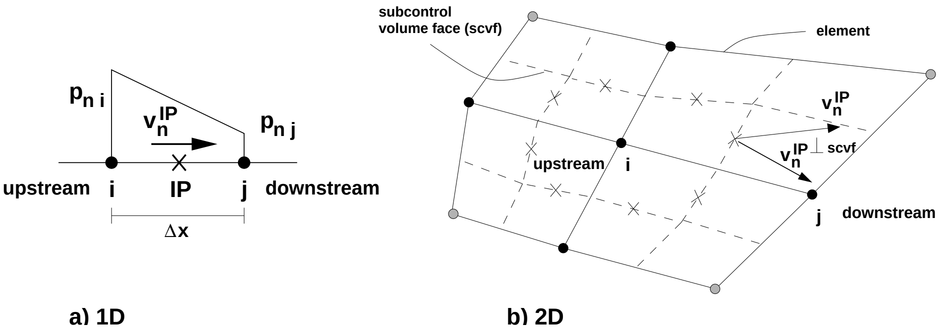


Figure 7.1: Definition of flow direction between nodes i and j

- The diffusive part of the two-phase flow ($\partial\,\mathbf{grad}\,p_c^{IP}/\partial S_{n|i}$) is indirectly a function of the discretization length Δx. This is illustrated with the help of Figure 7.2, which also shows how the variation of S_n was realized in the program.

  The first picture in Figure 7.2 shows the capillary-pressure function $p_c(S_n)$. Here, the different values of p_c at the nodes i and j are shown. Also, the capillary pressure $p_c(S_{ni}+\Delta S_n)$, corresponding to the 'varied' saturation, is plotted.

  The middle part of Figure 7.2 shows the representation of these values for two different mesh widths. If the saturations at the nodes are the same for two different meshes, the capillary pressures as well as the capillary pressure corresponding to the varied saturation are also the same. **grad** p_c is estimated at the integration point between the nodes i and j. The gradient which is present in the system is represented by the line marked with a black dot (1 and 3), the gradient which results from the variation is indicated by a line marked with a grey dot (2 and 4). If these values are transfered to the graph plotted in the lower left corner of Figure 7.2, the derivative of **grad** p_c with respect to S_n can now be calculated. The result is given in the lower right graph of Figure 7.2: the value for a smaller discretization length (Δx_1, square) is larger than the value for a larger discretization length (Δx_2, triangle).

If we assume homogeneous isotropic soil conditions between the nodes i and j, the two-phase *Peclet* number at an integration point can be written as:

$$Pe = \frac{\dfrac{\partial \lambda_n^{IP}}{\partial S_{n|i}}\left((\mathbf{grad}\,p_n^{IP})^{\perp\mathrm{scvf}} - \rho_n^{IP}\mathbf{g}^{\perp\mathrm{scvf}}\right)}{\lambda_n^{IP}\,\dfrac{\partial(\mathbf{grad}\,p_c^{IP})^{\perp\mathrm{scvf}}}{\partial S_{n|i}}} \tag{7.5}$$

In this representation, the *Peclet* number is again a scalar value. Now the two-phase *Peclet* number is derived, two things can be achieved:

On the one hand, it is now possible to switch locally between a fully upwinding or centrally weighted scheme. As was explained in Section 2.4.2, an upwinding scheme is required for advection-dominated problems in order to obtain stability. However, it also introduces artificial diffusion and is only of first order, as opposed to a scheme using central weighting of the mobilities. Therefore, it is desirable to introduce upwinding only, where it is necessary, i.e. where advection dominates.

For transport problems, the issue of an upwind control via the *Peclet* number is not new. Some criteria are given in, for example, *Noorishad et al.* [52], *Thiele* [59], *Neunhäuserer* [51]. As no findings are known for the application to the strongly coupled two-phase flow equations introduced in Section 2.3, the following relationship is utilized in a first approach:

$$\alpha_{up}(Pe) = \begin{cases} 0.0 & \text{if } |Pe| < 1.0 \\ 1.0 & \text{else} \end{cases} \tag{7.6}$$

This may seem rather strict, but is nonetheless on the safe side.

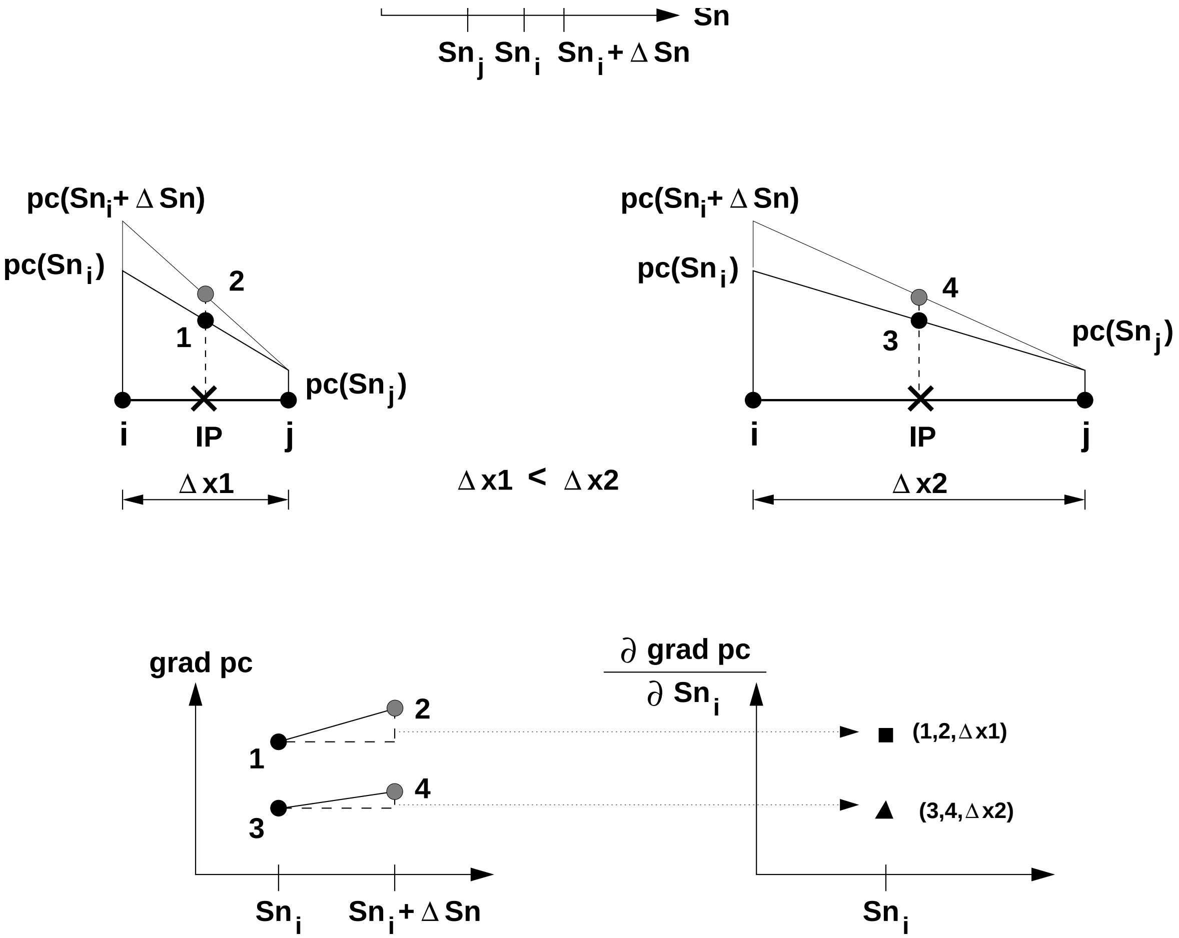


Figure 7.2: Explanation of $\partial \mathbf{grad}\, p_c^{IP} / \partial S_{n|i}$ for a one-dimensional element

On the other hand, it is now also possible with additional means to control the adaptive method. Here, the idea is to increase the element size in regions where diffusion effects are predominant. From the point of view of a physical interpretation of a diffusive process, the information at one point spreads out in all directions immediately throughout the system. Therefore, one is not restricted to small elements which reproduce the stepwise progression of the information in the hyperbolic case, but one can choose large elements. As an error indicator, however, may still point to diffusion dominant regions to refine the elements here, an additional criterion has to be introduced. This is now realized with the *Peclet* criterion, which overrides the refinement marks set by the error indicator.

For the upwinding control as well as for the adaptive indicator, an element *Peclet* number is computed by

$$Pe_{\text{elem}} = \max_{j \in IP(k)} |Pe_j| \tag{7.7}$$

where $IP(k)$ is the set of integration points for element k. This ensures for the upwinding control that the same upwind value is taken for one element. A variation from IP to IP would be possible, too. Considering, for example, the *Buckley-Leverett* problem, one can see that, for the integration points at subcontrol volume faces perpendicular to the flow, *Pe* becomes very large due to the large pressure gradient. For the integration points on subcontrol volume faces parallel to the flow, however, the value for *Pe* becomes very small, as for the quasi one-dimensional problem the pressure is constant over the height of the domain.

For the adaptivity control, only an element value makes sense, as only a whole element can be marked for refinement or coarsening.

Remark:
The two-phase *Peclet* number as it is shown here depends on the solution of the numerical simulation in each time step. This means, that for the Newton solver, where the node values are infinitesimally varied, also the *Peclet* number changes, which in turn would affect the solution. In order to circumvent this problem, in the simulation the *Peclet* number is once computed in the beginning of a time-step and then held constant.

7.2 *Buckley-Leverett* problem

At first, the *Peclet* criterion is applied to the uniform simulation, where only the upwind value is chosen adaptively. As the *Buckley-Leverett* problem is a purely advection-dominated problem, the *Peclet* number is always above 1 and thus no change in the discretization scheme takes place. The result is therefore identical to the results shown in Section 3.1 and are not shown here.

Then, the upwinding as well as the adaption criterion are employed. The distribution of the *Peclet* number in the domain for t = 500 days is shown in Figure 7.3. The output for *Pe* is for each element midpoint; the contour of the *Peclet* number therefore follows the line connecting those midpoints. As can be seen, the values are all well above 1, as anticipated for this advection-dominated problem.

One remark has to be made: For the *Buckley-Leverett* problem, the capillary pressure is set to zero. This implies that $(\partial\,\mathbf{grad}\,p_c)/(\partial S_n)$ is not defined. Mathematically, this would lead to $Pe = \infty$. For the numerical simulation, $(\partial\,\mathbf{grad}\,p_c)/(\partial S_n)$ was set to $\mathtt{max}\ (\varepsilon = 1.0e - 20, (\partial\,\mathbf{grad}\,p_c)/(\partial S_n))$. The same was done for $\lambda_n(S_n = 0)$.

The result in Figure 7.3 shows that, for the *Buckley-Leverett* problem, no changes occurs due to the *Pe*-criterion: as $Pe > 1.0$ is valid everywhere, fully upwinding is applied and the mesh is changed due to the chosen indicator only.

7.3 *McWhorter* problem

7.3.1 Uniform simulation

At first, a computation was carried out with a uniform mesh and the *Pe*-criterion applied to the upwind value. The resulting saturation (thick lines) and error (thin lines) distribution is shown in Figure 7.4. It can be observed that the deviation towards the analytical solution is marginally larger when the upwind value is chosen automatically. The reason for this can be found in Figure 7.5: Although the *McWhorter* problem is a purely parabolic (diffusion-dominated) problem, the *Peclet* number becomes larger than 1 in the region around the front. Therefore, the upwind value is changed from $\alpha_{up} = 0.0$, which is the default case for this problem, to $\alpha_{up} = 1.0$.

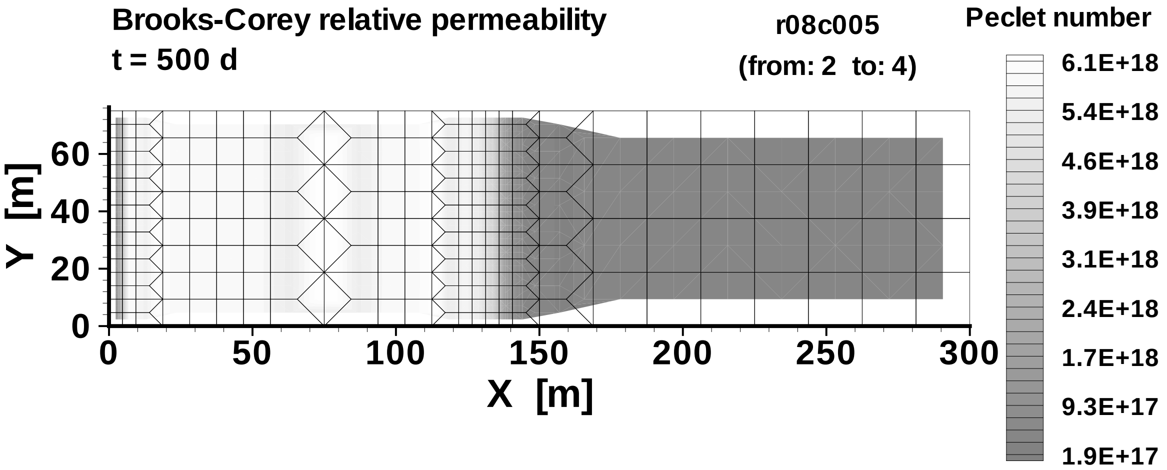


Figure 7.3: *Peclet* number distribution for the *Buckley-Leverett* problem

This phenomenon can be explained as follows:

The two-phase *Peclet* number for the $p_n - S_w$-formulation which is used for this problem is given as:

$$Pe = \frac{\frac{\partial \lambda_w^{IP}}{\partial S_{w|i}} \left((\mathbf{grad}\, p_w^{IP})^{\perp \mathrm{scvf}} - \rho_w^{IP} \mathbf{g}^{\perp \mathrm{scvf}} \right)}{\lambda_w^{IP} \, \frac{\partial (\mathbf{grad}\, p_c^{IP})^{\perp \mathrm{scvf}}}{\partial S_{w|i}}} \tag{7.8}$$

Figure 7.6 shows the water saturation and the phase and capillary pressures for t = 4000 s. The values of the components of *Pe* are given in Figure 7.7.

For the region between the inflow boundary and the saturation front, the gradient of the water pressure is very low, which leads to a small *Peclet* number. As, however, the water saturation decreases, p_c increases strongly for the *Brooks-Corey* function applied here (compare Figure 2.3). As the water pressure is computed via $p_w = p_n - p_c$ (see Eq. (2.9)), this results in a large decrease in p_w, which again leads to a large (absolute) value for $\mathbf{grad}\, p_w$. If *Pe* is now computed, a value larger than 1 is obtained. On the downstream side of the saturation front, $\mathbf{grad}\, p_w$ and thus *Pe* are again small.

7.3.2 Adaptive simulation

In a next step, the *Peclet*-upwinding criterion was applied to an adaptive simulation. The resulting saturation and error distribution is depicted in Figure 7.8. Here, the result is even worse than for the uniform case. Not only is the error larger, the execution time also increases dramatically when α_{up} is chosen according to the *Peclet* number (Figure 7.9).

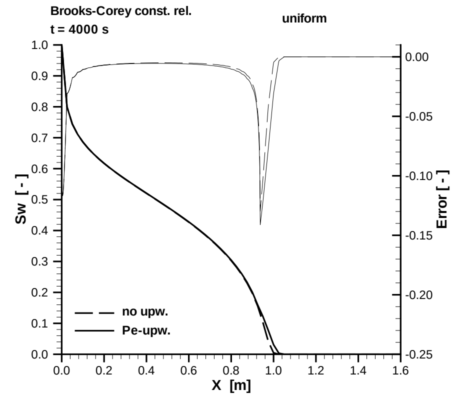


Figure 7.4: Water saturation and error distribution for the *McWhorter* problem, comparison between 'no upwinding' and '*Peclet* upwinding'

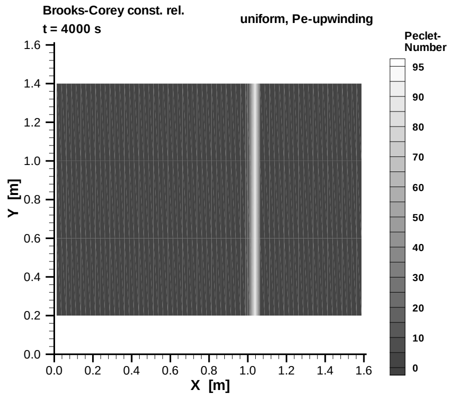


Figure 7.5: *Peclet* number distribution for the *McWhorter* problem, simulation with '*Peclet* upwinding'

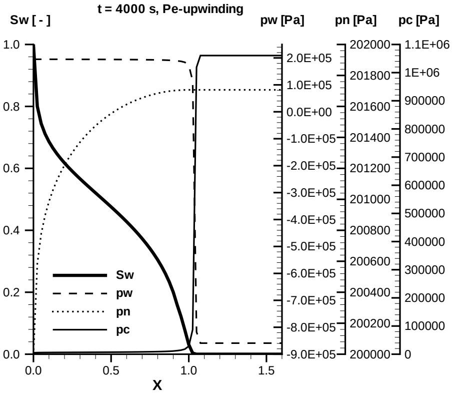


Figure 7.6: Water saturation, phase pressures and capillary pressure

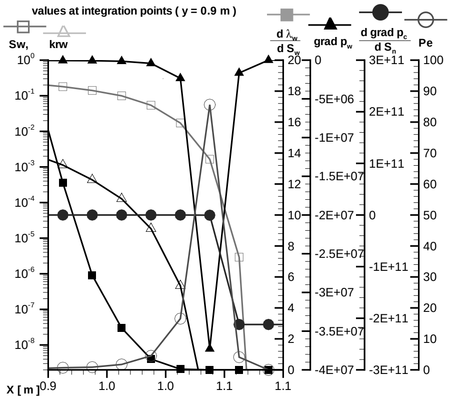


Figure 7.7: Distribution of the *Peclet* number and its components

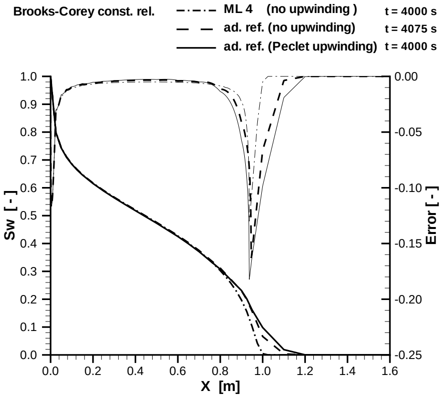


Figure 7.8: Water saturation and error distribution for the *McWhorter* problem, comparison between 'no upwinding' and '*Peclet* upwinding' for an adaptive simulation (thick lines: S_w, thin lines: error)

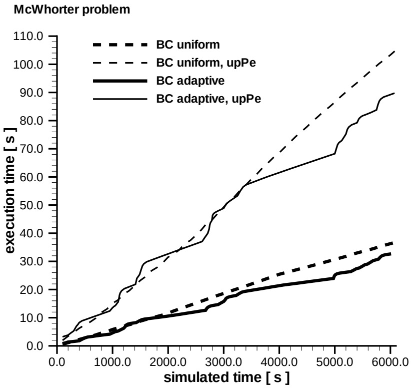


Figure 7.9: Execution time over simulated time for *Brooks-Corey* constitutive relationships. Uniform and adaptive case with 'no upwinding' and '*Peclet* upwinding'(upPe).

As explained in the introduction to this chapter, another application of the *Pe*-criterion is possible: a controlling of the adaptive refinement. Here, the idea is to coarsen regularly refined (red) elements when $Pe < 1$.

This method was applied to the adaptive simulation of the *McWhorter* problem. However, a result cannot be shown: within the first few time steps, the refinement of the first column of elements alternates between level 3 and 4 for each consequent time step and no convergence is reached.

A reason for this can be found when it is recalled that the derivative of the gradient of the capillary pressure after the saturation depends on the element size (compare Figure 7.2). If the element size Δx is increased, $(\partial \, \mathbf{grad} \, p_c)/(\partial S_w)$ decreases, which results again in an increase of the *Peclet* number. So for the elements near the boundary, the following loop is passed through:

- The element size is small: $Pe < 1$.
- As $Pe < 1$, the element is marked for coarsening.
- The element size is large: $Pe > 1$.
- As $Pe > 1$, a marking due to the error indicator is valid. The element gets marked for refinement.
- The element size is again small.

One way of avoiding this behavior and excluding the boundary influence from disturbing the adaptive refinement would be to apply a secondary criterion which applies the *Pe*-criterion for interior elements only.

7.4 Homogeneous model problem (Sandbox)

7.4.1 Uniform simulation

Again, the first step was to apply the *Pe*-criterion to the upwind value for a uniform case. The result is shown in Figures 7.10 and 7.11. In the left picture, the *Peclet* number and the resulting upwind value distribution can be seen. The value of *Pe* is around one behind the front and then increases very steeply. The right picture compares the saturation distribution of the standard case with the solution for the adaptively chosen upwind switch. It can be observed that the application of the *Pe*-criterion leads to saturation isolines which are not as smooth as for the fully upwind case. If we take a closer look at the location of the dents, we see that they are placed near the borderline between $\alpha_{up} = 0.0$ and $\alpha_{up} = 1.0$.

So far, the upwind value was chosen only between these two values. For the flow velocity, this means a hard transition from a centrally averaged mobility to a value due to the upwind

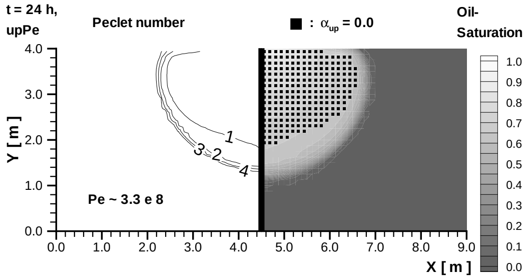


Figure 7.10: *Pe*-distribution and oil-saturation for the uniform case. The black squares indicate the elements, where $\alpha_{up} = 0.0$

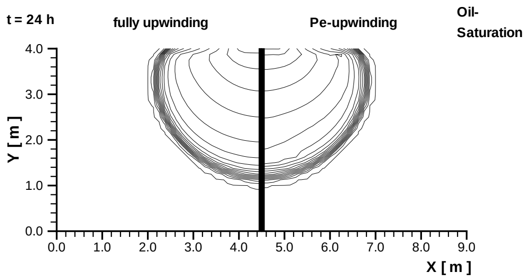


Figure 7.11: Comparison of the results 'fully upwinding' and '*Peclet* upwinding' for the uniform case

node. Therefore, in a second step, a 'blending' function for the upwind value was introduced (*Bastian* [6]):

$$\alpha_{up}(Pe) = \begin{cases} 0.0 & \text{if } Pe < Pe_{low} \\ 2\,b^2 & \text{if } Pe_{low} \leq Pe \leq Pe_{mid} \\ 1 - 2(1-b)^2 & \text{if } Pe_{mid} < Pe \leq Pe_{high} \\ 1.0 & \text{if } Pe_{high} < Pe \end{cases} \qquad \text{with } b = \frac{Pe - Pe_{low}}{Pe_{high} - Pe_{low}} \tag{7.9}$$

Here, Pe_{mid} represents the arithmetic mean between Pe_{low} and Pe_{high}. Figure 7.12 shows the result for Pe_{low} = 1.0 and Pe_{high} = 10.0. Here, all diffusion dominated areas ($Pe < 1$) are calculated with a centrally weighted mobility. As can be observed, this choice of lower and upper bounds does not lead to a correct solution. It seems that the advection clearly dominates the system as soon as Pe = 1 is reached and not using fully upwinding in these regions does not reproduce the physical behavior.

For the result shown in Figure 7.13, the bounds were chosen as Pe_{low} = 0.5 and Pe_{high} = 2.0. Here, the saturation distribution looks much smoother. Near the inflow, however, some sharp bends can still be observed. An explanation for this behavior is shown in Figure 7.10: when looking at the distribution of the upwind values, we notice that, for the top row of the mesh, fewer elements are marked with a value of $\alpha_{up} = 0.0$ than for the row immediately below. This means that, along the boundary, the velocity is higher and therefore the front propagates faster.

In order to compensate for this effect, all boundary elements were excluded from the *Pe*-criterion, leading to α_{up} = 1.0 for these elements. The results are shown in Figure 7.14 without and with blending. It shows that, with this method, the isolines are relatively smooth again.

7.4.2 Adaptive simulation

In a second step, the *Pe*-criterion was applied to an adaptive case. At first, only the upwinding was adapted. The result (without blending) can be seen in Figure 7.15. The results are even worse than for the uniform case. It can be observed that the number of elements with $Pe < 1$ increased.

Next, the upwinding value was again set to fully upwinding in the whole domain and the grid adaptivity was controlled. Figure 7.16 shows the results: the mesh was coarsened in some regions due to a *Peclet* number below one. The saturation distribution looks very similar to the 'standard adaptive' case.

Finally, both aspects of the *Pe*-criterion were combined. However, there is a discrepancy between the resulting saturation distribution and the reference solution - the effects of the *Peclet* upwinding (Figure 7.15) are even worsened. (see Figure 7.17).

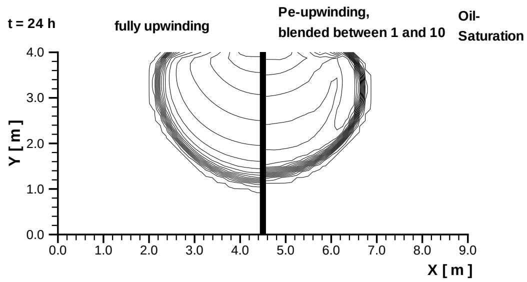


Figure 7.12: Comparison of the results 'fully upwinding' and '*Peclet* upwinding' with a blending for α_{up} between Pe = 1 and 10 (uniform case)

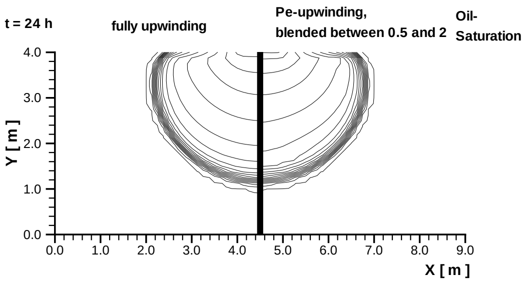


Figure 7.13: Comparison of the results 'fully upwinding' and '*Peclet* upwinding' with a blending for α_{up} between Pe = 0.5 and 2.0 (uniform case)

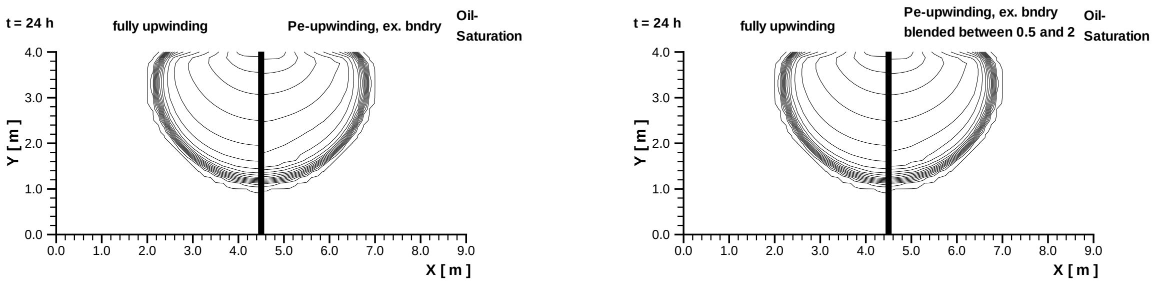


Figure 7.14: Comparison of the results 'fully upwinding' and '*Peclet* upwinding' with excluded boundary elements for the uniform case. Left: hard switch between α_{up} = 0.0 and 1.0; right: blending for α_{up} between Pe = 0.5 and 2.

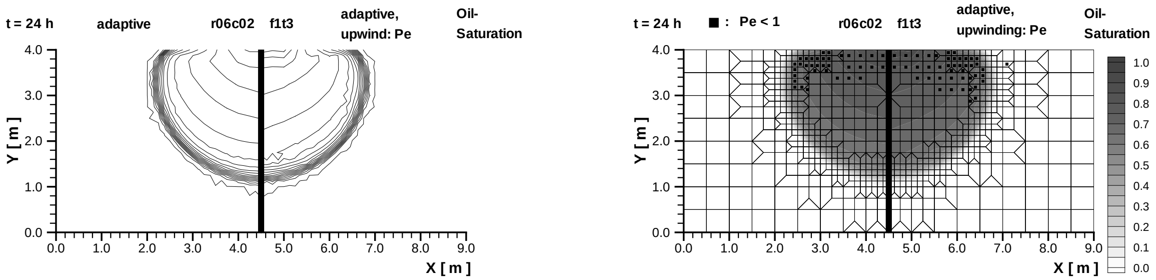


Figure 7.15: Comparison of the results 'fully upwinding' and '*Peclet* upwinding', adaptive case. Left: saturation distribution; right: mesh. Elements with $Pe < 1$ are marked.

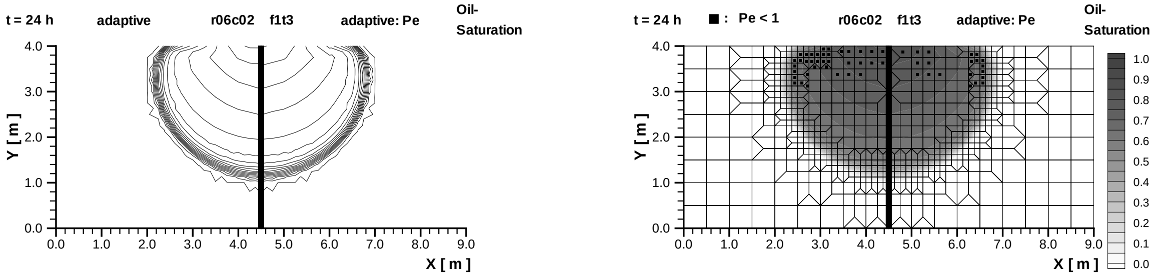


Figure 7.16: Comparison of the results 'standard adaptive' and '*Peclet* adaptive'

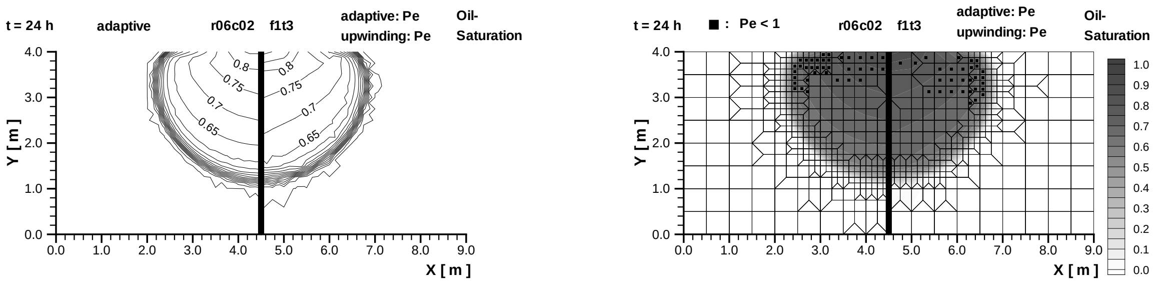


Figure 7.17: Comparison of the results 'standard adaptive' and '*Peclet* adaptive with *Peclet* upwinding'

8 Two-phase flow in a heterogeneous medium: the test case

8.1 Heterogeneous model problem (Sandbox with lense)

In addition to the homogeneous test case as introduced in Section 3.3, a heterogeneous case was also investigated. As outlined in the introduction, heterogeneities have an important impact on the behavior of subsurface flow. As they influence the flow paths strongly, they may, for example, decelerate the spreading of contaminants or direct them to different regions.

Heterogeneities are important for the flow processes, yet their numerical consideration is difficult. The two-phase flow equations and constitutive relationships themselves are already highly non-linear. When domains with different properties, such as, for example, different permeabilities, are introduced as well, the coefficients in the coefficient matrices vary to a larger extent, which makes the numerical computation even more challenging.

The set-up of the chosen test case can be found in Figure 8.1. It resembles the homogeneous test case introduced in Section 3.3 and is leaned on cases in the literature (see e.g. *Helmig* [31]). However, the soil and fluid properties are changed so that they are now based on the values used by *Kueper et al.* [42]. The applied values are given in Tables 8.1 and 8.2.

The simulation is carried over a time period of 240 min with the parameters presented in Table 8.3.

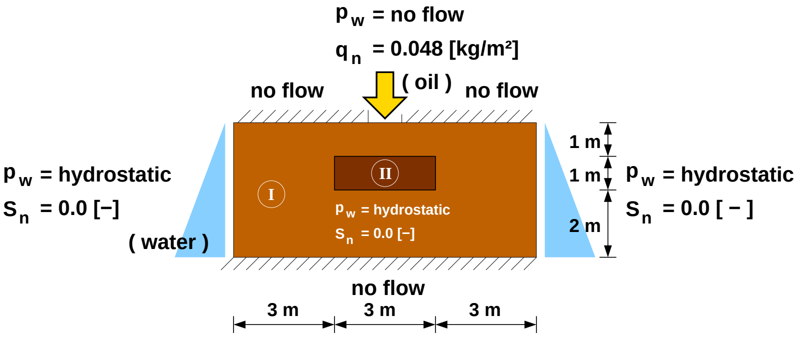


Figure 8.1: System, initial and boundary conditions of the heterogeneous system

Table 8.1: Fluid properties (heterogeneous model problem)

water density	ρ_w	=	998	[kg/m^3]	dyn. visc. water	μ_w	=	0.001	[kg/(m s)]
oil density	ρ_n	=	1631	[kg/m^3]	dyn. visc. oil	μ_n	=	0.0009	[kg/(m s)]

Table 8.2: Soil properties (heterogeneous model problem)

Domain 1					Domain 2 (Lense)				
abs. perm.	$\mathbf{K}_1$	=	5.04 x 10^{-10}	[m^2]	abs. perm.	$\mathbf{K}_2$	=	5.26 x 10^{-11}	[m^2]
porosity	ϕ_1	=	0.4	[-]	porosity	ϕ_2	=	0.4	[-]
res. sat. water	$S_{wr,1}$	=	0.078	[-]	res. sat. water	$S_{wr,2}$	=	0.098	[-]
res. sat. oil	$S_{nr,1}$	=	0.0	[-]	res. sat. oil	$S_{nr,2}$	=	0.0	[-]
entry pres.	$p_{d,1}$	=	370.0	[Pa]	entry pres.	$p_{d,2}$	=	1324.0	[Pa]
distr. ind.	λ_1	=	3.86	[-]	distr. ind.	λ_2	=	2.49	[-]

Table 8.3: Simulation parameters (heterogeneous model problem)

Initial conditions:					Discretization:				
water pressure	p_w	=	hydrostatic	[Pa]	step size (space)	Δx	=	0.125	[m]
oil saturation	S_n	=	0.0	[-]	step size (time)	Δt	=	120	[s]

8.2 Uniform simulation of the heterogeneous test case

The results obtained with the Box Method and with the Box-Trans Method are shown in Figure 8.2. Here, two distinct differences can be observed:

1. The oil infiltrates into the lense when the Box Method is applied whereas, for the Box-Trans Method, a pool is formed on top of the lense and no infiltration takes place.

   This behavior is due to the phenomenon explained in Section 2.4.2.2 and Figure 2.12 therein: for the Box Method an infiltration can take place before S_w^* is reached. We will return to this problem in the next section.

2. The boundary of the lense seems to be located differently for the two discretization schemes.

   The reason for this can be found when Figures 2.8 and 2.13 are referred to again. Whereas the interface of the two subdomains is located along the subcontrol volume faces for the Box Method, it is located along the element edges for the Box-Trans Method. This results in a positioning of the oil pool one element length above the interface for the Box Method (for the visualization, node values are chosen and interpolated).

In a next step, the upwinding was chosen according to the *Peclet* criterion. The results for $t = 60$ min are shown in Figure 8.3. The points in time for the results with '*Peclet* upwinding' differ from the reference solution, as the time step was adaptively reduced due to numerical difficulties. The elements for which $Pe < 1$ is valid are marked by black squares. It can be noted that they are all positioned around the interface of the lense. For all these elements, α_{up} was set to zero.

It can be observed that the oil saturation becomes negative. This phenomenon was described in Section 2.4.2.2 (Figure 2.11).

The subfigures 8.3.a through d represent the saturation distribution at two cuts across the domain. Figures a) and b) show the saturation at $t = 2722$ sec. Here, the oil has just reached the lense and has not yet spread to the sides. Due to the centrally weighted mobility, oil starts flowing out of the lense, which leads to negative saturations. Figures c) and d) show the saturation state before the simulation was aborted. Note that the peaks of the negative values are now located above the interface. Here, it has to be recalled that the position of the interface for the simulation is half an element length above the dotted line.

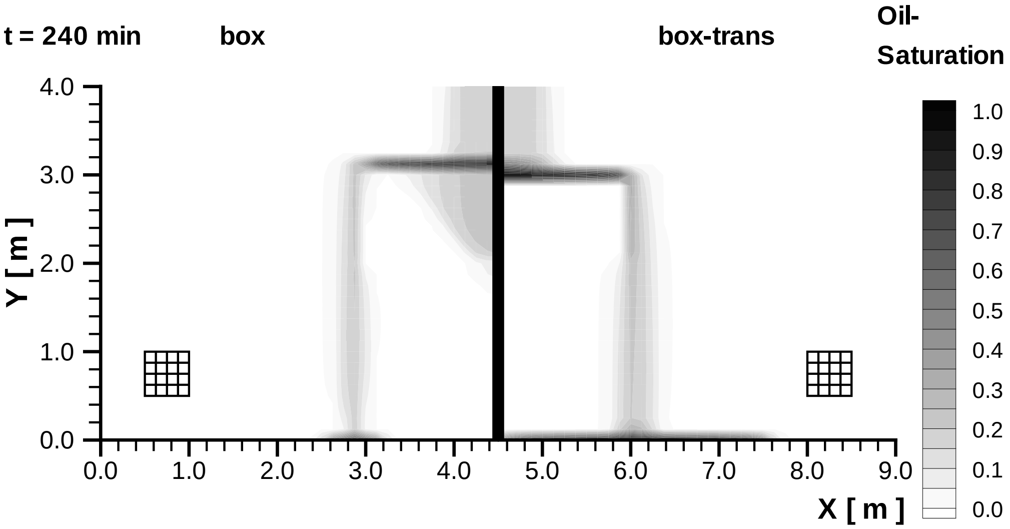


Figure 8.2: Oil saturation distribution after 240 minutes. Left: Box Method; right: Box-Trans Method. The applied mesh size is indicated by the clippings in the corners.

The results show that, for the case of a pooling of the oil on the lense, the flow takes on a diffusive character in this region. This would lead to a choice of an arithmetically weighted mobility according to the *Peclet* criterion. As explained before, this choice is not acceptable when interfaces between domains with different properties are considered. Therefore, the use of the *Peclet* criterion is not advisory when simulating heterogeneous domains with the Box Method.

8.3 Adaptive simulation of the heterogeneous test case

Before the adaptive simulation can be carried out, some preliminary investigations have to be made. The central question concerns the appropriate choice of the values for the refinement and coarsening criterion. As already shown for the homogeneous case, the size of the indicator values varies throughout the domain. Therefore, at first, a uniform simulation was carried out for the Box Method and the indicator values in the domain were calculated. Figure 8.4 shows the result for t = 240 min.

Here, the complex distribution of the values is clearly visible. It can be seen that the indicator takes on very large values for the elements located above the lense. For the adaptive simulation, not only this region needs to be located, but also the regions where the oil flows down from the lense. In these areas, however, the indicator values are relatively small. The task which has to be solved is to find values which locate the crucial regions, but exclude those parts where a refinement is not necessary. In order to do this, several cuts were taken

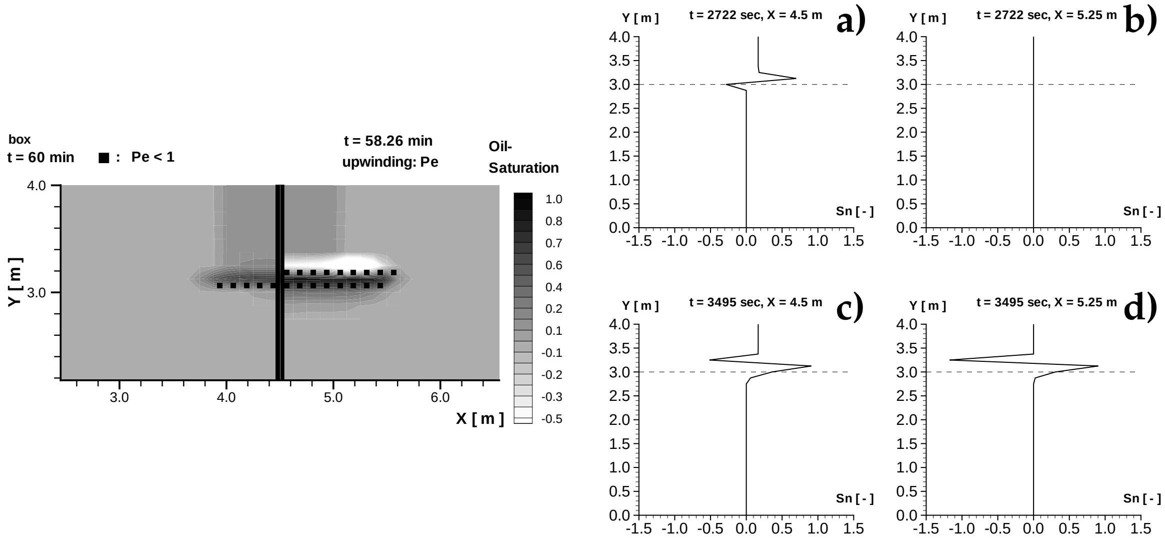


Figure 8.3: Left: Comparison of the saturation distribution with 'fully upwinding' and '*Peclet* upwinding' for the uniform case (detail). Right: Saturation distribution at cuts through the system for the case '*Peclet* upwinding' for different points in time.

throughout the domain. These are shown in Figure 8.5. Based on these results, the criteria ′refinement bound = 0.25′ and ′coarsening bound = 0.05′ were chosen.

The corresponding results of the adaptive simulation carried out with these parameters are shown in Figure 8.6. Here, one can see that the refinement is located only around the critical regions.

As mentioned in Section 8.2, an infiltration takes place for the simulation with the Box Method, in contrast to the behavior exhibited by the Box-Trans Method. With the help of adaptive methods, it is now conveniently possible to reproduce the correct infiltration performance for the Box Method by choosing appropriately small elements along the subdomain interface. The result for a refinement up to level 6 is shown in Figure 8.6 (right). Here, it becomes clearly visible to which extent the discretization length influences the entering of oil into the lense.

A comparison of the required number of elements and nodes for the case presented here is given in the following table:

	uniform		adaptive	
	ML 3	ML 6	ML 3	ML 6
elements	2304	147456	698 (0.3)	10322 (0.07)
nodes	2409	148289	718 (0.3)	10834 (0.07)

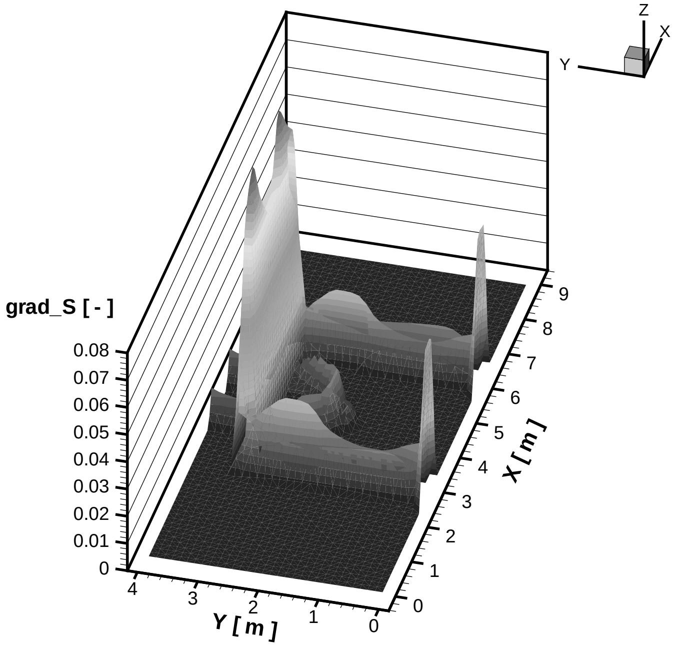


Figure 8.4: Distribution of the indicator values for the *grad_S* indicator at t = 240 min, based on a uniform simulation.

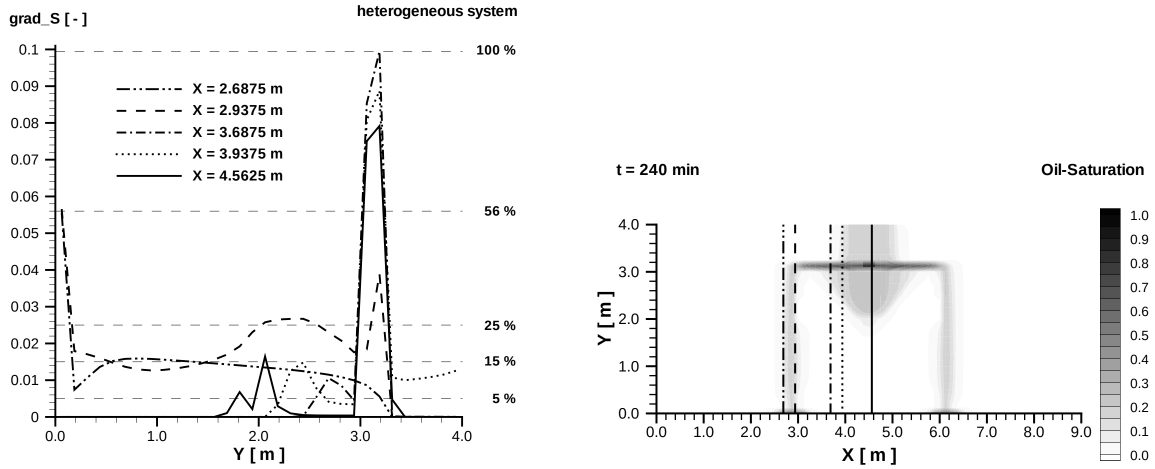


Figure 8.5: Values of the *grad_S* indicator for different locations in the domain. The positions of the cuts are given in the right picture.

It shows that the number of nodes needed by the adaptive algorithm in order to obtain the same resolution as the uniform case is by 30 % for a maximum refinement level of 3 ($\Delta x = 0.125$ m) and even only 7 % for a maximum refinement level of 6 ($\Delta x = 0.015625$ m)!

At this point, the aspect of mass conservation is again brought up. The following table shows the masses of the wetting and non-wetting phases present in the system at t = 240 min for adaptive refinement up to level 3.

		with mass correction	without mass correction
water	[kg]	13 948.3	13 968.7
oil	[kg]	691.1	657.8

The values for the case with mass correction equal the values when the infiltration rate is summed up over time. The difference for the simulations with and without mass correction

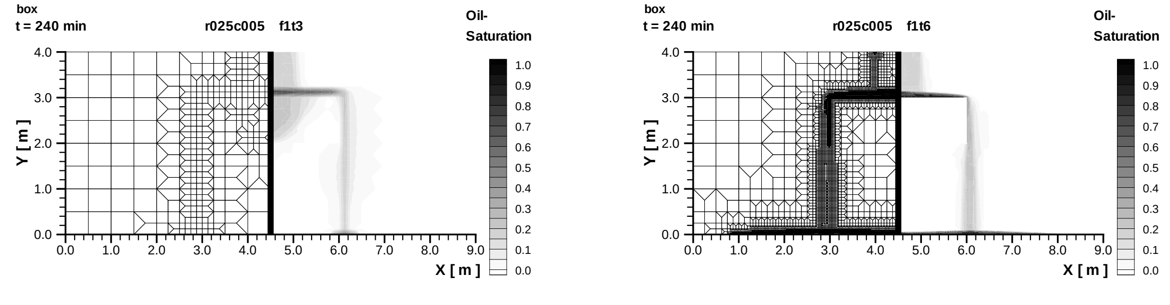


Figure 8.6: Comparison of the adaptive simulation with different refinement levels for the Box Method. Left: refinement up to level 3; right: refinement up to level 6

is visible. Although the percentage of the deviation may not be large in the present case, it does represent an error which accumulates over time. Especially when one is interested in the exact point in time for the infiltration, the existing mass in the system plays an important role. As the elements located above a lense have first to be filled with oil to a certain extent in order to overcome the entry pressure, more oil in the system, for example, will lead to a faster increase in the pressure and therefore to an infiltration time which is smaller than the time obtained by the uniform reference simulation.

Following the same procedure as for the homogeneous case in Section 7.4.2, the *Peclet* criterion was applied to the adaptive simulation of the heterogeneous case.

At first, only the upwinding was controlled. The result can be found in Figure 8.7. Here, the same non-physical saturation distribution is obtained as in the uniform case. Due to the *Peclet* criterion, the upwinding is switched off for the elements along the lense and with central weighting, negative saturations occur.

Then, the *Pe*-criterion was applied to the adaptive refinement. Figure 8.8 shows the saturation distribution for the Box Method (left) and the Box-Trans Method (right). With the help of these results, it becomes visible to which extent the mesh geometry influences the solution. Although the upwind value α_{up} is set to 1.0 for the whole domain, oil saturations up to -2.0 develop at nodes located above the lense when the *Pe*-criterion is applied. (Note that the colors of the contour legend were reversed for this case.) An explanation for this behavior can be given as follows:

As the elements with $Pe < 1.0$ are located on top of the lense, these elements are marked for coarsening, as explained before. However, the same rules as for regular coarsening have to

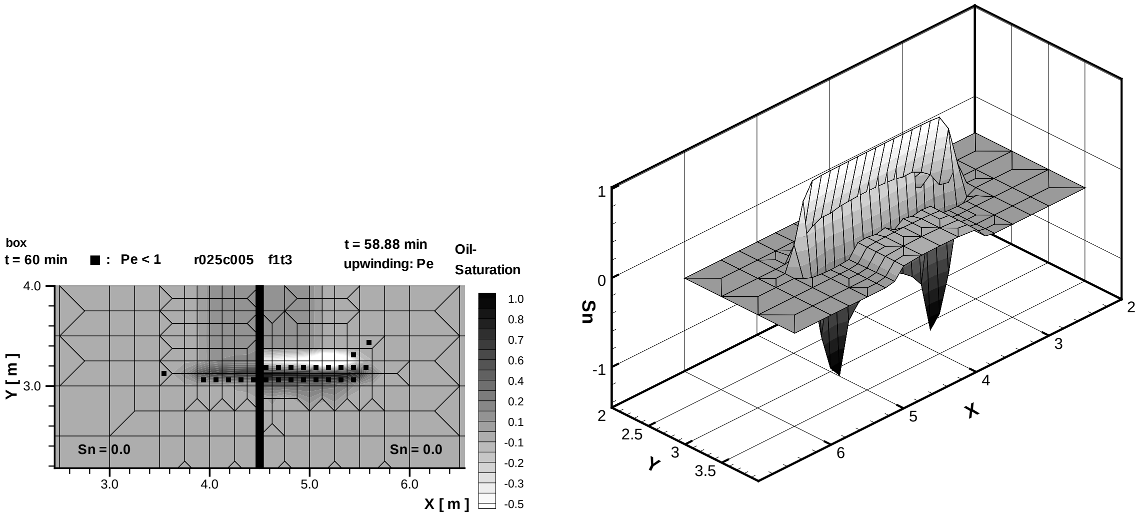


Figure 8.7: Left: Comparison of the oil saturation distribution for the adaptive simulation with fully upwinding and *Pe*-upwinding (detail). Right: Detail of the oil saturation above the lense for *Pe*-upwinding.

be obeyed: an element can only be coarsened if all brother elements are marked for coarsening, too. This leads to the fact that, in some areas, larger elements exist along the interface than in other areas. This is marked by a white circle on the left of Figure 8.8. As the interface between two subdomains runs in the middle between two nodes for the Box Method, its course experiences a peak in the indicated area. This behavior together with an unfortunate element geometry for this case then leads to oil saturations below zero.

Another effect can be observed when looking at the results for the Box-Trans Method in the right part of Figure 8.8. Here, the trail of small elements accompanying the flow of the oil down from the lense is broken in the middle. Due to the larger elements now present, the saturation distribution is now spread to the sides much more.

To complete this chapter, the results are shown for combining the *'Peclet* upwinding' with the *'Peclet* refinement' in Figure 8.9. It can be seen that the negative effects - saturations below zero, irregular boundary of the lense due to different element geometries, smearing of the saturation due to large elements - influence each other and lead to a very non-physical result.

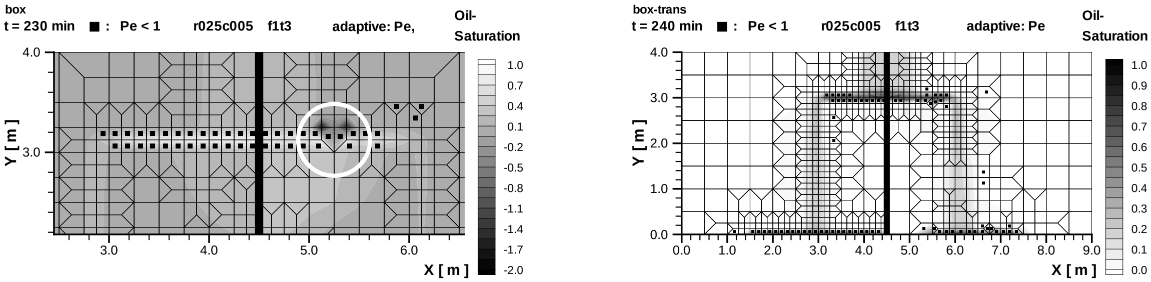


Figure 8.8: Mesh and oil saturation distribution for the adaptive simulation with 'standard adaptivity' and *'Peclet* adaptivity'. Left: Box Method; right: Box-Trans Method

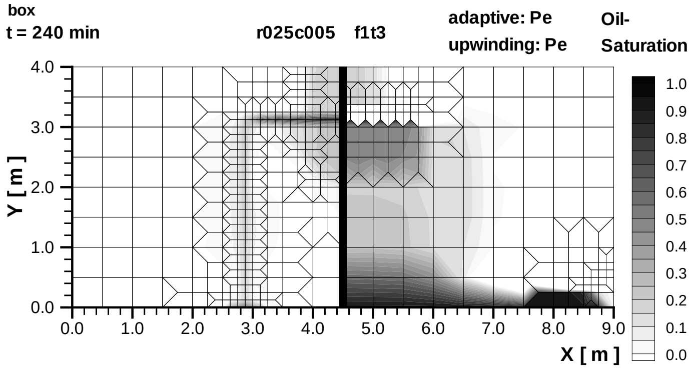


Figure 8.9: Comparison of oil saturation and mesh for the standard Box Method and the combination of *'Peclet* upwinding' and *'Peclet* refinement'

9 Applications

9.1 Simulation of experimental results

As a conclusion of the preceding chapters, the h-adaptive method introduced is now utilized to reproduce an experiment which was carried out by the author of this thesis at the Institut für Hydromechanik, ETH Zürich, Switzerland (*Paul et al.* [56]).

The use of the developed *Peclet* criterion is abandoned for this case, as the results presented in this work show that further investigations concerning the specific application of this method are needed.

The chosen experiment reproduces the flow of water through a dike with a slanted core. This situation is idealized by a *Hele-Shaw* cell (see Figure 9.1). For the case of no capillary-pressure effects, the laminar flow through a thin slot between a model and the glass front of the container is a valid approximation of the flows occurring in nature. In the experiment, different permeabilities inside the model are realized by applying different slot widths. The underlying relation is given by

$$\frac{k_1}{k_2} = \frac{{d_1}^3}{{d_2}^3} \tag{9.1}$$

where k_i are the different conductivities and d_i represent the slot widths of the zones i (*Bear* [10]).

Figure 9.2 shows the geometry of the dike model used. It consists of two zones with the same permeability (zones 1 and 3) and a zone with a lower permeability (zone 2). Zone 4 is added for the numerical simulation, which is described below.

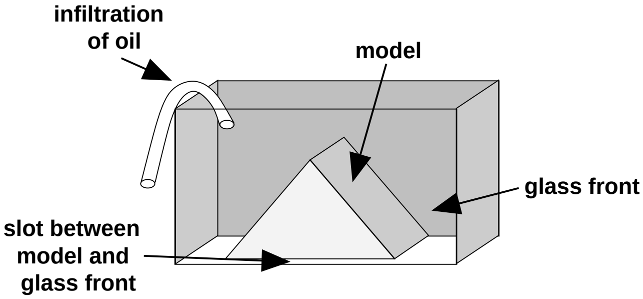


Figure 9.1: Schematic sketch of a *Hele-Shaw* cell

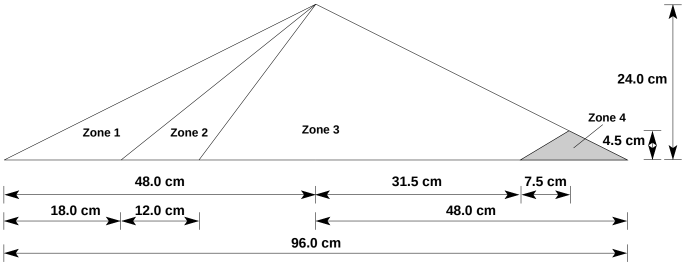


Figure 9.2: Geometry of the dike model

The course of the experiment is as follows: the oil level on the left side of the air-filled dike was gradually raised to a height of 20.6 cm. The oil table is then held constant, until a steady-flow state inside the dike has been reached. The resulting oil distribution is shown in Figure 9.3. Here, the white dashed line indicates the oil table serving as an inflow condition for the system.

The figure shows that the oil level stays nearly constant in the first zone it encounters. Inside the core (zone 2), the free surface shows a small decline. In zone 3, the phreatic surface is significantly lower than in the other two zones and proceeds in a curve downwards to the exit point located in the right part of the domain. In this picture, the influence of a heterogeneity can clearly be observed: the oil is held back and can flow only gradually into zone 3. The transition zone between the core and the rightmost part of the system is covered by a veil of oil. Here, oil was running down the front plate in the experiment and the slot was not completely filled by oil.

This is an effect which can be modelled well by a two-phase flow simulation tool. As opposed to a classical groundwater-flow model, it has the advantage that even complex forms of the surface between water and oil (or air) can be modelled. Another advantage is that the dropping zone of the oil that developed here, where the saturation of each phase is less than one, can also be accounted for.

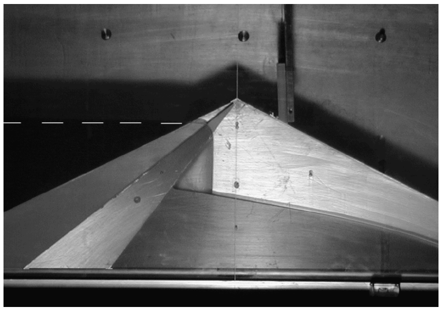


Figure 9.3: Experiment in the *Hele-Shaw* cell, steady state (after *Paul et al.* [56])

Table 9.1: Soil properties (dike-experiment)

	Zone 1	Zone 2	Zone 3	Zone 4
$\mathbf{K}$ $[m^2]$	9.8 x 10^{-10}	3.6 x 10^{-11}	9.8 x 10^{-10}	9.8 x 10^{-8}
Porosity $[-]$	0.37	0.35	0.37	0.20
S_{wr} $[-]$	0.05	0.10	0.05	0.15

The numerical simulation was carried out with the soil properties given in Table 9.1. The constraint which had to be obeyed was the ratio of the permeabilities, which was chosen as $k_1/k_2 = 1 : 27.3$ in the experiment. The values applied here are based on real sands (*Sheta* [58]). The residual saturation of the non-wetting phase was set to zero in all zones.

When capillary effects are excluded, as in the present case, the difference between the viscosities of the fluids utilized in the experiment does not influence the steady state shown here. Therefore, in order to reproduce the infiltration into a real dike more accurately, water was chosen as the infiltrating fluid for the numerical simulation.

The results of the numerical simulation are shown in Figures 9.4 through 9.6. At first, the results of a uniform simulation are shown (mesh and water-saturation distribution, Figure 9.4). It can be observed that the agreement of the numerical and experimental results are very good: even the dropping zone mentioned above was correctly represented. However, due to the numerical interpolation of the saturation values for the elements located at the interface between water and air, the phreatic surface does not resemble a smooth curve as in the experiment.

Figures 9.5 and 9.6 show the mesh and water saturation for the adaptive simulation at two different points in time. The pictures illustrate nicely, that the adaptive algorithm correctly identified the sharp interface between water and air. Also, due to the smaller element size near the interface, the free surface resembles more the results obtained from the experiment.

The following table shows a comparison of the number of nodes and elements for the uniform and adaptive simulations.

	number of nodes	number of elements	smallest element edge (approx.)
uniform	1393	2592	0.0125 [m]
adaptive	1518	2553	0.0041 [m]

It can be seen that, for approximately the same number of elements, the adaptive simulation shows a much better resolution with elements of the water-air interface.

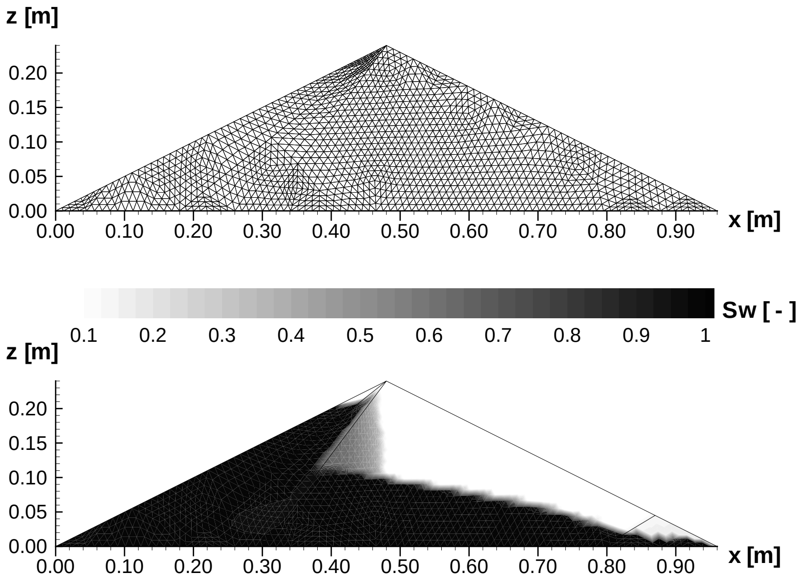


Figure 9.4: Mesh (top) and saturation distribution (bottom) resulting from the uniform simulation (t = 149 s)

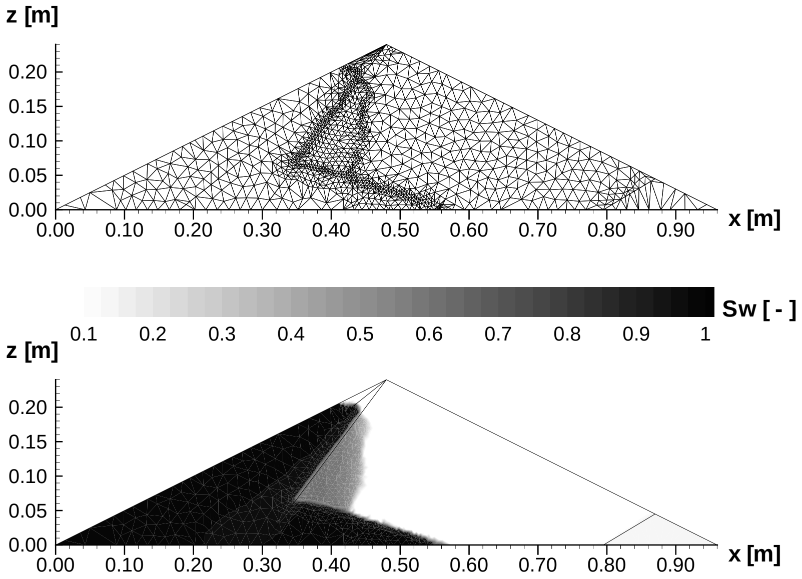


Figure 9.5: Mesh (top) and saturation distribution (bottom) resulting from the adaptive simulation (t = 53 s)

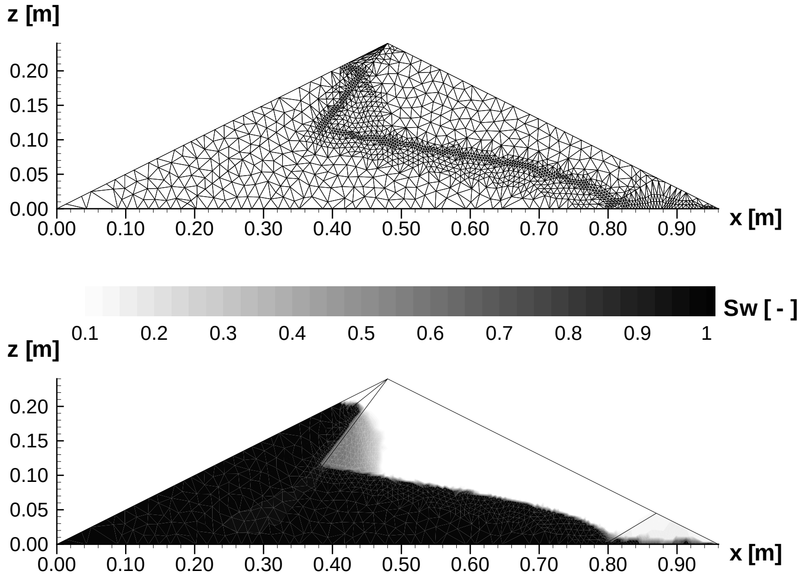


Figure 9.6: Mesh (top) and saturation distribution (bottom) resulting from the adaptive simulation (t = 149 s)

9.2 Simulation of a real-life dam

In this section, the h-adaptive method is applied to a real-life dam. The domain is shown in Figure 9.7. As it can be seen, the structure is very heterogeneous. The dam was built up continuously over time, which can also be seen by the location of the different subdomains: At first, a base dam (domain 2 in the lower right) was erected and the void space was filled up. As the filling material increased, further support was necessary, so the domains 14, 13, and 15 were added.

The surface of the composite dam very gently slopes upwards from left to right. On top of the dam, a lake is situated. It ranges from the left side to the point marked with 'a' in Figure 9.7. As this water level has been constant over a long time, the saturation distribution in the domain has reached a quasi steady-state. Due to higher inflow in the lake, the surface spreads further out over the dam. The spreading is not continuously, but given in different steps: After an initial period of one day, the water spreads to point 'b'. At day 45, the water increases further to point 'c' and at day 60 the last position, point 'd', is reached.

The question is now, how the water will infiltrate into the dam. As the stability of a system, and especially of a composite sytem as it is the case here, is highly influenced by the water content in the soil, it is very important to develop an understanding of how the water will spread out in the system.

Starting with an initial water distribution in the dam, shown in Figure 9.8, the infiltration of water into the domain over time is simulated. The soil properties of the different subdomains are given in Table 9.2. As no other data was available, the porosity ϕ was set to 0.4 for all subdomains. Also, S_{wr} was set to 0.25 and S_{nr} to 0.0 in the whole domain. The capillary pressure was set to zero.

Figures 9.9 and 9.10 show the saturation distribution after 205 days and 6 hours for an uniform and adaptive refinement, respectively. For these two cases, the size of the smallest elements were chosen the same. It can be clearly seen that the water infiltrates along the whole lake length from the top into the domain. As the hydraulic conductivity in subdomain 12 is larger than in domain 6, the water infiltrates here faster. This results in a large air bubble which is entrapped in the system. Over time, this entrapped air will leave the system, also, but it becomes obvious that a two-phase approach is needed to account for this behaviour. It can also be observed that the water does not infiltrate into the base dam, which has a very low hydraulic conductivity.

The following table shows the execution time and amount of elements and nodes needed for the simulations. For the adaptive simulation, the largest elements were equal in size than for the case 'ML 0'. The smallest elements resembled the size for the case 'ML 2'. As it can be seen, the gain in time with the adaptive method is clearly visible.

	uniform		adaptive
	ML 0	ML 2	f0t2
execution time [s]	518	57 079	11 983
elements	5 130	82 080	11 672
nodes	2 746	41 761	6 629

When looking at the execution time, one has to bear in mind that the total number of nonlinear and linear iterations is different for all the three cases. For the uniform simulation with coarse elements, the total sum is 612, for the fine mesh 24450 and for the adaptively refined mesh 28766.

Figure 9.11 shows a detail of the adaptive mesh. It can clearly be observed that the applied (gradient) indicator detects the saturation front very well, even in the case of a heterogeneous medium.

Table 9.2: Soil properties of the different subdomains

Subdomain	**K** [m^2]	Subdomain	**K** [m^2]	Subdomain	**K** [m^2]
1	1 x 10^{-15}	6	1 x 10^{-13}	11	1 x 10^{-14}
2	1 x 10^{-15}	7	5 x 10^{-15}	12	5 x 10^{-13}
3	5 x 10^{-14}	8	1 x 10^{-13}	13	4 x 10^{-12}
4	1 x 10^{-15}	9	1 x 10^{-15}	14	1 x 10^{-13}
5	5 x 10^{-14}	10	5 x 10^{-13}	15	1 x 10^{-12}

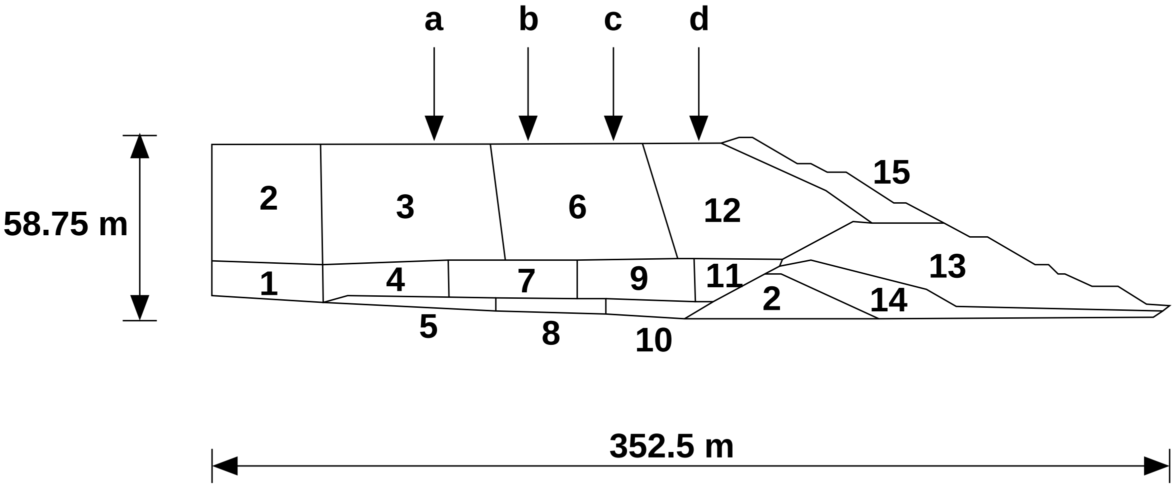


Figure 9.7: Domain and subdomains of the dam

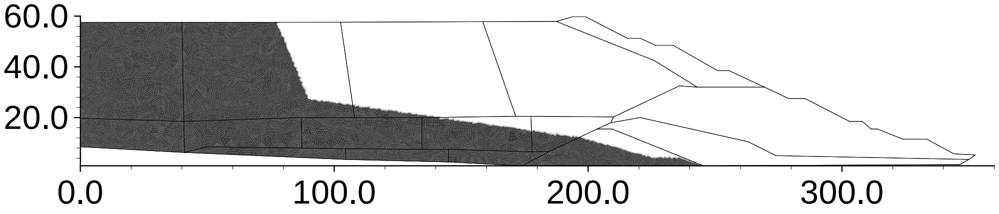


Figure 9.8: Initial saturation of the dam

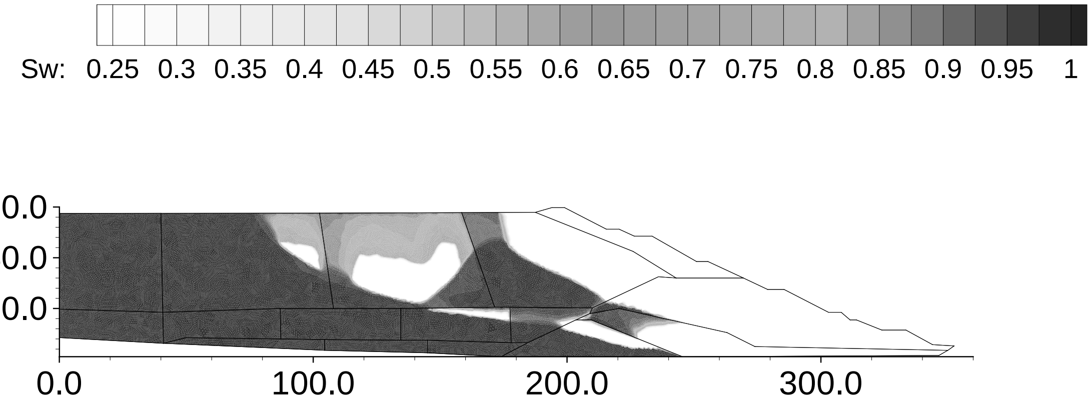


Figure 9.9: Water saturation after 205 days 6 hours for the uniform refinement (ML 2)

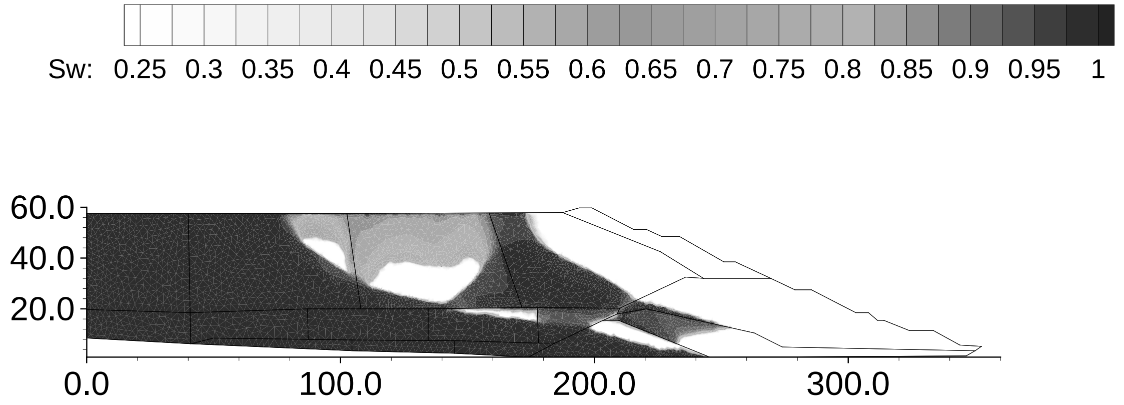


Figure 9.10: Water saturation after 205 days 6 hours for the adaptive refinement

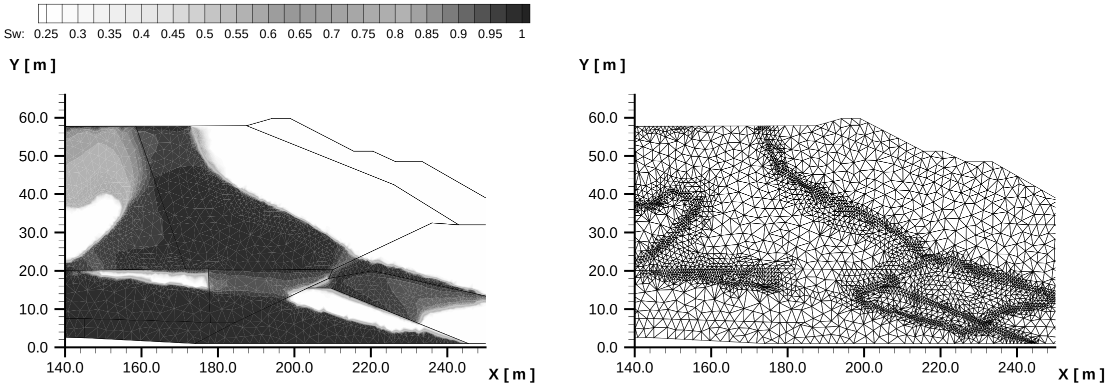


Figure 9.11: Zoom into the refined mesh. Left: saturation distribution; right: adapted mesh

10 Summary and outlook

When encountering engineering problems of two-phase flow processes in porous media, one often has to deal with large and complex systems. As the specific points of interest located in the domain may have comparatively small extensions, for example shafts and roads located in a large mine system (*Breiting* [12]), a sufficiently fine element resolution is required in order to be able to account for the relevant processes. This, however, leads to huge computational efforts and storage requirements.

Basically it is possible to choose the initial mesh in such a way that some regions are covered by a fine and other regions by a coarse element resolution. The resulting meshes, however, have the disadvantage of being static throughout the course of the simulations. This means that it is not possible react to transient processes as, for example, the proceeding of a saturation front. For this problem, h-adaptive methods represent an efficient solution technique. While reducing the number of degrees of freedom calculated for one time step, they still ensure a high accuracy of the solution in the vital areas.

In this work, the h-adaptive methods incorporated into the numerical simulation tool MUFTE-UG are applied to various test cases. Here, the aim is to develop an understanding of the relevant processes and parameters with the help of well-known basic problems of two-phase flow. These basic problems are extended to a simple problem without and with a heterogeneity. Then, with the experience gathered in these applications, the adaptive algorithm is finally employed for the simulation of an experiment.

Before the homogeneous test cases are introduced in Chapter 3, the governing equations and the used discretization schemes are explained in Chapter 2. Afterwards, an overview of adaptive methods in general and the methods particularly applied in this work is given in Chapter 4.

As the aspect of mass conservation plays an important role in engineering problems (for example when considering nuclear-waste deposition sites (*Helmig et al.* [32])) a whole chapter (Chapter 5) was devoted to this topic. Here, the different aspects which are important when uniform and adaptive refinements are considered are stated. Especially when h-adaptive methods are used, care has to be taken, as nodes, and thus the mass represented by them, may be omitted due to the manipulation of the mesh. Therefore, an algorithm ensuring mass conservation for these cases is introduced, which is developed during this work.

In Chapter 6, the adaptive simulation of the introduced homogeneous test cases is presented. Here, the influence of different refinement and coarsening criteria as well as of different indicators is shown. Whereas no specific recommendation for the choice of refinement and coarsening bounds can be made, as each problem requires its own carefully chosen set, the *grad_S* indicator is chosen for all subsequent simulations. This indicator identifies large gradients of the saturation, which usually occur in regions where a sharp front proceeds through the system and therefore a fine element resolution is required. This indicator is also easy to implement and to compute, which means that no (large) additional computational effort has to be made in order to calculate the indicator values. Otherwise, if the computation of error indicators or mainly estimators becomes very tedious, the advantage of a smaller number of degrees of freedom melts under the supplementary computational load.

Beside the simulation results, the number of non-linear iterations and averaged number of linear iterations per Newton step are also compared over the simulation time for the homogeneous test cases. Here, it becomes noticeable that the numerical challenge posed by the problems to the solvers increases not only with the choice of non-linear constitutive relationships as opposed to linear ones, but also with the introduction of differently sized elements in the domain.

After the adaptive simulation of the homogeneous test cases, the two-phase *Peclet* number is derived as a criterion indicating diffusion- or advection-dominant regions (Section 7.1). The background for this is to be found in the fact that advection- and diffusion-dominated processes each require different discretization techniques (e.g. upwinding). As in the case of the two-phase flow equations, these processes are strongly coupled and each extremum may occur in a problem domain at the same point in time. Therefore, the *Peclet* criterion is introduced in order to control the individual upwinding parameter. In addition to the upwinding, the coarsening of elements can also be controlled by the *Peclet* criterion. Here, the aim is to coarsen domains where diffusion is dominant, as these regions do not necessarily require a fine resolution.

Both aspects of the *Peclet* criterion are applied to the homogenous test cases in Chapter 7. For the *Buckley Leverett* problem, the *Peclet* number is always larger than 1, so both criteria are not enabled. For the *McWhorter* and the sandbox problem, however, the numerical solution with *Peclet* criteria differs from the uniform reference solution. For the *McWhorter* problem, the reason can be found in the singularity of the *Peclet* number located directly at the interface. For the sandbox problem, the explanation is not that obvious. It is shown that the numerical simulation is very sensitive to a change in the upwind values. Here, the developed criteria can only be understood as a step towards a better understanding of the predominant processes.

Subsequently, a heterogenous test case is introduced in Chapter 8. Here, the difficulties that arise when encountering domains with different properties are explained. Beside the Box Method, the Box-Trans Method is also applied, which explicitly consideres the entry conditions at an interface.

For a medium-refined mesh, an infiltration of oil into the fine sand lense of the test domain takes place for the Box Method, whereas the oil forms a pool on top of the lense for the Box-Trans Method. For this example, also the influence of the discretization length when heterogeneities are regarded can be clearly noted: If the refinement is increased, the Box Method shows the same pooling behavior for the oil as the Box-Trans Method. This underlines again the need for h-adaptive methods.

When the *Peclet*-upwind criterion is applied, it can be noted that the quality of the performance of the Box Method decreases dramatically: physically wrong solutions (oil saturations below zero) are obtained. The reason for this is that the *Peclet* number becomes smaller than 1 in the vicinity of the interface between the two media. This would lead to an upwind value of $\alpha_{up} = 0.0$ (centrally weighted mobility). However, especially in the crucial region around the interface, a fully upwinding technique has to be utilized for the Box Method.

For the Box Method as well as for the Box-Trans Method, the h-adaptive algorithms introduced show good results. However, special care has to be taken when the refinement and coarsening bounds are chosen, as the heterogeneity has a considerate influence on the indicator distribution. Due to the pooling of the oil on the top of the lense, the gradient of the saturations in this region becomes very large.

The use of the *Peclet* coarsening criterion for the heterogeneous case shows only moderate results. Here, the reason lies in the fact that, with this criterion, some elements are coarsened along the subdomain boundary, which leads to a shifting of the interface. This, again, results in numerical difficulties and unphysical results.

In Chapter 9 of this thesis, the adaptive method is applied on the one hand to the numerical simulation of an experiment and on the other hand to the simulation of a real-life dam.

The experimental example shown is based upon a common engineering problem: the flow of water through a dike. Although the experimental set-up was kept small and capillary pressure effects were excluded, the resulting problem shows most of the effects which are crucial for a numerical simulation: regions with different soil parameters and a sharp front as well as a region where the saturation is more smeared, are present. A comparison between experimental and numerical results shows that the adaptive component of MUFTE-UG handles this problem very well.

The application to the dam-problem clearly shows the advantage of the adaptive method. As the domain is very large, usually a large number of elements would be necessary to resolve the moving saturation front for a static mesh. With the adaptive method it is possible to keep a small element size in the front region while coarsening elements in regions where no relevant flow takes place. With this method, the execution time needed can clearly be decreased. Here again it is shown that the applied indicator is suitable for heterogeneous media also.

Outlook

As the investigations for transport equations show (*Neunhäuserer* [51]), upwind-control is a sensible and useful measure. However, great care has to be taken when choosing the appropriate values for two-phase flow processes, as the examples in this thesis show. The investigations carried out here may serve as a basis for further studies concerning suitable transition functions from $\alpha_{up} = 0.0$ to $\alpha_{up} = 1.0$ for the applications shown here.

Especially for the additional criterion shown here regarding the coarsening of elements, a better adjustment between element size and the *Peclet* number required needs to be made. The strong non-linear components of the two-phase *Peclet* number introduced here makes this no easy task.

The investigations undertaken to find a suitable error indicator are carried out with an extension to multiphase-multicomponent flows in mind. Especially for the case of non-isothermal multicomponent systems, where a switch of primary variables may take place (e.g. *Class* [17]), adjustments have to be made. This will also be required for the mass-conservation algorithm.

A further expansion of the adaptive algorithm could be to include the calculation of the gradient indicator and mass-conservation for the Box-Trans-Method. As in this method the stored node values need not necessarily be the relevant values when looking at the node from different elements, it has to be made sure to take the 'correct' value for the indicator and mass calculation.

And last but not least, a further task would be to expand this algorithm to parallel simulations and problems in three dimensions.

Bibliography

[1] Aziz, K. and Settari, A. *Petroleum Reservoir Simulation*. Elsevier, London, New York, 1979.

[2] Babuška, I. and Rheinboldt, W. C. A-posteriori error estimates for the finite element method. *Int. J. Numer. Methods Eng.*, 12:1597 – 1615, 1978.

[3] Babuška, I. and Rheinboldt, W. C. Error estimates for adaptive finite element computations. *SIAM J. Numer. Analys.*, 15:736 – 754, 1978.

[4] Barlag, C. *Adaptive Methoden zur Modellierung von Stofftransport im Kluftgestein*. Dissertation, 1997.

[5] Bastian, P., Birken, K., Johannsen, K., Lang, S., Neuss, N., Rentz-Reichert, H., and Wieners, C. UG - a flexible software toolbox for solving partial differential equations. *Computation and Visualization in Science*, 1, 1997.

[6] Bastian, P. `box2p.c`. MUFTE, Box discretization for two-phase flow, dimension independent, compressible.

[7] Bastian, P. Numerical Computation of Multiphase Flows in Porous Media. Habilitation, Christian-Albrechts-Universität Kiel, 1999.

[8] Bastian, P., Chen, Z., Ewing, R. E., Helmig, R., Jakobs, H., and Reichenberger, V. Numerical Simulation of Multiphase Flow in Fractured Porous Media. In Chen, Ewing, and Shi, Hrsg., *Numerical Treatment of Multiphase Flows in Porous Media*, Lecture Notes in Physics, Berlin, Heidelberg, 2000. Springer Verlag.

[9] Bastian, P. and Wittum, G. Adaptive Multigrid Methods: The UG Concept. In Hackbusch, W. and Wittum, G., Hrsg., *Adaptive Methods - Algorithms, Theory and Applications. Proceedings of the Ninth GAMM-Seminar*, S. 17 – 37. Vieweg, Braunschweig.

[10] Bear, J. *Dynamics of Fluids in Porous Media*. Elsevier, New York, 1972.

[11] Braun, C. Simulation von Zweiphasenströmungen im Untergrund mit Hilfe einer adaptiven Finite-Element-Formulierung. Diplomarbeit, Institut für Wasserbau, Universität Stuttgart, 1993.

[12] Breiting, T. Modellierung und Analyse von Mehrphasenprozessen zur Simulation von Methanausgasung im Untergrund. Forschungsbericht, Institut für ComputerAnwendungen im Bauingenieurwesen, Technische Universität Brauschweig, 2000.

[13] Breiting, T., Hinkelmann, R., and Helmig, R. Modeling of Hydrosystems with MUFTE-UG: Multiphase Flow and Transport Processes in the Subsurface. In *4th International Conference on Hydroinformatics*, Iowa, USA, 2000.

[14] Brooks, R. and Corey, A. Hydraulic Properties of Porous Media. In *Hydrol. Pap.*, Band 3, Fort Collins, 1964. Colorado State University.

[15] Buckley, S. E. and Leverett, M. C. Mechanism of Fluid Displacements in Sands. *Transactions of the AIME*, 146:107 – 116, 1942.

[16] Bürkle, D. and Ohlberger, M. Adaptive finite volume methods for displacement problems in porous media. Institut für Angewandte Mathematik, Universität Freiburg, 2001.

[17] Class, H. *Theorie und numerische Modellierung nichtisothermer Mehrphasenprozesse in NAPL-kontaminierten porösen Medien*. Dissertation, Universität Stuttgart, Institut für Wasserbau, 2001.

[18] Class, H., Helmig, R., and Bastian, P. Numerical Simulation of Nonisothermal Multiphase Multicomponent Processes in Porous Media - 1. An Efficient Solution Technique. *Advances in Water Resources*, to appear.

[19] Class, H., Paul, M., Manthey, S., Jakobs, H., and Sheta, H. MUFTE-UG: A Documentation. Lehrstuhl für Hydromechanik und Hydrosystemmodellierung, Institut für Wasserbau, Universität Stuttgart, 2002.

[20] van Duijn, C., Molenaar, J., and de Neef, M. Effects of capillary forces on immiscible two-phase flow in heterogeneous porous media. *Transport in Porous Media*, 21:71 – 93, 1995.

[21] Dullien, F. A. L. *Porous Media: Fluid transport and pore structure*. Academic Press, San Diego, London, 1992.

[22] Dyck, S. and Peschke, G. *Grundlagen der Hydrologie*. Verlag für Bauwesen, Berlin, 1995.

[23] Ellsiepen, P. *Zeit- und ortsadaptive Verfahren angewandt auf Mehrphasenprobleme poröser Medien*. Dissertation, Universität Stuttgart, Institut für Mechanik, Lehrstuhl II, 1999.

[24] Ferziger, J. H. and Perić, M. *Computational Methods for Fluid Dynamics*. Springer, Berlin, Heidelberg, New York, 2002.

[25] Forsyth, P. A control volume finite element approach to NAPL groundwater contamination. *SIAM J. Sci. Stat. Comp.*, 12:1029 – 1057, 1991.

[26] Hackbusch, W. *Multi-grid methods and applications*. Springer - Verlag, Berlin, 1985.

[27] Helmig, R. and Bastian, P. Efficient fully-coupled solution techniques for two-phase flow in porous media - Parallel multigrid solution and large scale computations. *Advances in Water Resources*, 23:199 – 216, 1999.

[28] Helmig, R. and Huber, R. Comparison of Galerkin-type discretization techniques for two-phase flow in heterogeneous porous media. *Advances in Water Resources*, 21(8):697 – 711, 1998.

[29] Helmig, R. et al. Architecture of the Modular Program System MUFTE-UG for Simulating Multiphase Flow and Transport Processes in Heterogeneous Porous Media. *Mathematische Geologie*, 2, 1998.

[30] Helmig, R. *Theorie und Numerik der Mehrphasenströmungen in geklüftet-porösen Medien*. Dissertation, Universität Hannover, Institut für Strömungsmechanik und Elektron. Rechnen im Bauwesen, 1993.

[31] Helmig, R. *Multiphase Flow and Transport Processes in the Subsurface*. Springer - Verlag, Heidelberg, 1997.

[32] Helmig, R., Bastian, P., Jakobs, H., and Reichenberger, V. Multiphase Multicomponent Processes in Fractured Porous Media . Final report 10/1997 - 12/2000, Institut für ComputerAnwendungen im Bauingenieurwesen, Technische Universität Braunschweig, 2000. BMWi - BmBF – Unterauftrag zum Vorhaben 02 E 9027 8.

[33] Hinkelmann, R. Efficient Numerical Methods and Information-Processing Techniques in Environment Water. Habilitation, Universität Stuttgart, 2002.

[34] Huber, R. *Compositional Multiphase Flow and Transport in Heterogeneous Porous Media*. Dissertation, Technische Universität Brauschweig, Institut für Computeranwendungen im Bauingenieurwesen, 1999.

[35] Huber, R. and Helmig, R. Multiphase flow in heterogeneous porous media: A classical Finite Element Method versus an Implicit Pressure - Explicit Saturation-Based mixed Finite Element - Finite Volume approach. *International Journal for Numerical Methods in Fluids*, 29:899 – 920, 1999.

[36] Huyakorn, P. and Pinder, G. *Computational Methods in Subsurface Flow*. Academic Press, London, 1983.

[37] Jakobs, H., Helmig, R., Miller, C. T., Class, H., Hilpert, M., and Kees, C. E. Multiphase flow and transport modeling in heterogeneous porous media. *Advances in Water Resources*, 2002. to be published.

[38] John, V. A Comparison of Some Error Estimators for Convection-Diffusion Problems on a Parallel Computer. Forschungsbericht, Institut für Analysis und Numerik, Otto-von-Guericke-Universität Magdeburg, 1994.

[39] Johnson, C. and Hansbo, P. Adaptive finite element methods in computational mechanics. *Comput. Methods Appl. Mech. Engrg.*, 101:143 – 181, 1992.

[40] Johnson, C. Adaptive Finite Element Methods for Diffusion and Convection Problems. *Comput. Methods Appl. Mech. Engrg.*, 82:301 – 322, 1990.

[41] Kröner, D. *Numerical schemes for conservation laws*. Wiley-Teubner, Chichester, 1997.

[42] Kueper, B., Abbott, W., and Farquhar, G. Experimental Observations of Multiphase Flow in Heterogeneous Porous Media . *Journal Contaminant Hydrology*, 5:83–95, 1989.

[43] Lake, L. *Enhanced Oil Recovery*. Prentice–Hall, Inc., Englewood Cliffs, New Jersey, 1989.

[44] Lang, S. *Parallele Numerische Simulation instationärer Probleme mit adaptiven Methoden auf unstrukturierten Gittern*. Dissertation, Universität Stuttgart, Institut für Wasserbau, 2001.

[45] Lantz, R. Quantitative evaluation of numerical diffusion (truncation error). *Society of Petroleum Engineers Journal*, 251:315–320, 1971.

[46] Leonard, B. "A survey of finite differences of opinion on numerical modelling of incomprehensible defective confusion equation". In Hughes, T., Hrsg., *Finite Element Methods for Convection Dominated Flow*. AMD–ASME, New York, 1979.

[47] LeVeque, R. J. *Numerical Methods for Conservation Laws*. Birkhäuser - Verlag, Basel, 1992.

[48] McWhorter, D. B. and Sunada, D. K. Exact Integral Solutions for Two-Phase Flow. *Water Resources Research*, 26(3), 1990.

[49] de Neef, M. and Molenaar, J. Analysis of dnapl infiltration in a medium with a low–permeable lens. *Computational Geosciences*, 1:191–214, 1997.

[50] de Neef, M. *Modelling Capillary Effects in Heterogeneous Porous Media*. Dissertation, Technische Universiteit Delft, 2000.

[51] Neunhäuserer, L. *Diskretisierungsansätze zur Modellierung von Strömungs– und Transportprozessen in geklüftet–porösen Medien*. Dissertation, Universität Stuttgart, Institut für Wasserbau, 2002.

[52] Noorishad, J., Tsang, C., Perrochet, P., and Musy, A. A Perspective on the Numerical Solution of Convection-Dominated Transport Problems: A Price to Pay for the Easy Way Out. *Water Resources Research*, 28(2):551 – 561, 1992.

[53] Ohlberger, M. A posteriori error estimate for vertex centered finite volume approximations of convection-diffusion-reaction equations. *M2AN Math. Model. Numer. Anal.*, 35(2):355 – 387, 2001.

[54] Ohlberger, M. *A Posteriori Error Estimates and Adaptive Methods for Convection Dominated Transport Processes*. Dissertation, Mathematische Fakultät, Universität Freiburg, 2001.

[55] Papastavrou, A. *Adaptive Finite Element Methoden für Konvektions-Diffusionsprobleme*. Dissertation, Ruhr-Universität Bochum, 1998.

[56] Paul, M., Stauffer, F., Hinkelmann, R., and Helmig, R. Calibration of a Water-Gas Flow Model for Dike Systems. In L.R. Bentley, J.F. Sykes, C. B. W. G. G. P., Hrsg., *Proceedings of XIII. Conference on Computational Methods in Water Resources, 25. - 29. June*, Band 1, S. 201 – 208, Calgary, Canada, 2000. A.A. Balkema - Rotterdam, Brookfield.

[57] Scheidegger, A. The Physics of Flow Through Porous Media. In *University of Toronto Press*, Toronto and Buffalo, 3rd Edition, 1974.

[58] Sheta, H. *Simulation von Mehrphasenvorgängen in porösen Medien unter Einbeziehung von Hysterese-Effekten*. Dissertation, Universität Stuttgart, Institut für Wasserbau, 2000.

[59] Thiele, K. *Adaptive Finite Volume Discretisation of Density Driven Flows in Porous Media*. Dissertation, Lehrstuhl für Angewandte Mathematik I, Universität Erlangen-Nürnberg, 1999.

[60] Verfürth, R. *A Review of A Posteriori Error Estimation and Adaptive Mesh-Refinement Techniques*. Wiley-Teubner, Chichester, Stuttgart, 1996.

[61] Zienkiewicz, O. C. and Zhu, J. Z. A simple error estimator and adaptive procedure for practical engineering analysis. *Int. Journal for Numerical Methods in Engineering*, 24:337 – 357, 1987.

Index

Zusammenfassung

Einleitung

Ingenieurfragestellungen, die sich mit Zwei-Phasen-Strömungsprozessen in porösen Medien beschäftigen, beinhalten oft große und komplexe Systeme und werden zunehmend mit Hilfe von numerischen Simulationsmethoden bearbeitet. Da ein speziell interessierender Bereich im Untersuchungsgebiet vergleichsweise kleine Abmessungen aufweisen kann, wie zum Beispiel Schächte und Strecken in einer Kohlengrube, ist eine ausreichend feine Auflösung mit Elementen notwending, um die relevanten Prozesse numerisch erfassen zu können. Diese Bedingung führt jedoch zu großem rechnerischen Aufwand und Speicherplatzbedarf.

Grundsätzlich ist es möglich, das Berechnungsnetz am Anfang so zu wählen, dass einige Bereiche fein und andere Bereiche grob mit Elementen aufgelöst sind. Diese Netze haben jedoch den Nachteil, dass sie im Verlauf der Rechnung statisch sind, d.h. sich nicht an die jeweiligen Prozesse, zum Beispiel das Fortschreiten einer Sättigungsfront, anpassen können. Für diese Problematik liefern adaptive Methoden eine effiziente Lösungsmöglichkeit.

Auf der einen Seite werden durch eine automatische Anpassung der Elementgrößen im Gebiet über die Zeit die Freiheitsgrade und damit der Rechenaufwand und Speicherplatzbedarf reduziert, auf der anderen Seite können die wesentlichen Bereiche weiterhin fein genug aufgelöst werden.

Eine weitere Schwierigkeit bei der Simulation von Zwei-Phasen-Strömungsprozessen in porösen Medien besteht darin, dass sowohl die advektiven als auch die diffusiven Effekte, die zur gleichen Zeit im Untersuchungsgebiet auftreten können, wiedergegeben werden müssen. Da das jeweilige Verhalten eine spezielle Behandlung durch die Diskretisierungsmethode erfordert, wird in dieser Arbeit eine weitere adaptive Methode entwickelt. Hierbei geht es um das Erfassen des jeweiligen vorherrschenden Zustandes mit Hilfe der Zwei-Phasen-*Peclet*-Zahl, die das Verhältnis zwischen Advektion und Diffusion beschreibt.

Grundgleichungen

Den Berechnungen zugrunde liegen die Zwei-Phasen-Strömungsgleichungen, die in der Druck-Sättigungsformulierung verwendet werden:

benetzende Phase (z.B. Wasser) w

$$L_w(p_w, S_n) \;:=\; -\phi\,\frac{\partial(S_n\,\rho_w)}{\partial t} \;-\; \mathrm{div}\,(\rho_w\,\lambda_w\,\mathbf{K}\,(\mathbf{grad}\,p_w \;-\; \rho_w\,\mathbf{g})) \;-\; \rho_w\,Q_w \;=\; 0 \qquad (1)$$

nicht-benetzende Phase (z.B. Öl) n

$$L_n(p_w,S_n) \;:=\; \phi\,\frac{\partial(S_n\,\rho_n)}{\partial t} \;-\; \mathrm{div}\,(\rho_n\,\lambda_n\,\mathbf{K}\,(\,\mathbf{grad}\,p_c \,+\, \mathbf{grad}\,p_w \;-\; \rho_n\,\mathbf{g})) \\ -\,\rho_n\,Q_n \quad = \; 0 \qquad (2)$$

Hierbei bezeichnen die Variablen die folgenden Größen ($\alpha \in w,n$): p_α Druck der jeweiligen Phase α, S_α Sättigung der Phase α, ϕ Porosität, ρ_α Dichte der Phase α, t Zeit, λ_α Mobilität der Phase α, $\mathbf{K}$ Permeabilität, p_c Kapillardruck, $\mathbf{g}$ Vektor der Erdbeschleunigung und Q_α einen Quellen- oder Senkterm für Phase α.

Diskretisierungsmethoden

Die Gleichungen (1) und (2) werden mit Hilfe der Box-Methode diskretisiert. Für den heterogenen Fall wird des Weiteren die Box-Trans-Methode (PPSIC-Methode) eingesetzt. Diese Methoden unterscheiden sich zum einen darin, dass die Lage einer Grenzfläche Γ zweier Gebiete mit unterschiedlichen Materialeigenschaften jeweils anders behandelt wird (siehe Abb. 1). Zum anderen stellt die Box-Trans-Methode sicher, dass ein Eindringen von z.B. einer Öl-Sättigung in ein Gebiet erst stattfinden kann, wenn die Schwellensättigung (threshold saturation) S_w^* erreicht wird. Diese Sättigung beschreibt den Punkt auf der Kapillardruck-Sättigungskurve, in dem der Kapillardruck p_c^1 eines höher durchlässigen Gebietes gleich dem Eindringdruck p_d^2 eines weniger durchlässigen Gebietes ist (siehe auch Abb. 2).

Adaptive Methoden

Es wird eine orts-adaptive Methode, die sogenannte h-adaptive Methode, angewendet, bei der eine Anpassung der Elementgröße durch Verfeinern (Seitenhalbierung) und Vergröbern (Umkehr des Verfeinerungsprozesses) der vorhandenen Elemente erreicht wird. Für die Ent-

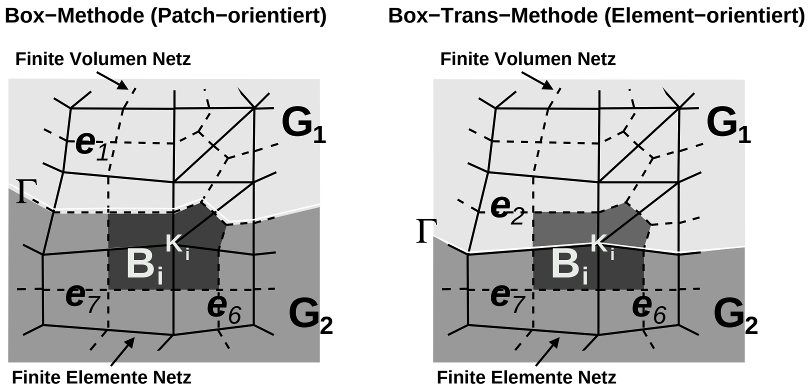


Abb. 1: Box-Methode (Patch-orientiert) und Box-Trans-Methode (Element-orientiert) (nach *Jakobs et al.* [37])

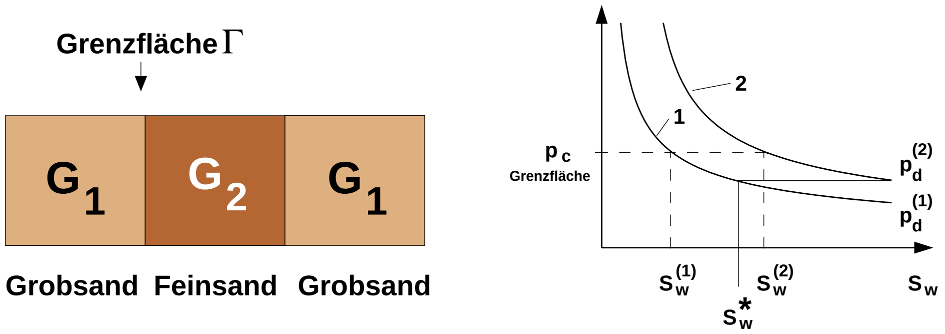


Abb. 2: Sprung der Sättigung an der Grenze zweier unterschiedlicher Medien aufgrund der Kontinuität im Kapillardruck und Erläuterung der Schwellensättigung S_w^*

scheidung, ob verfeinert oder vergröbert werden muss, kommt ein Gradienten-Indikator (*grad_S*) zum Einsatz, der den Gradienten der Sättigung im Element erfasst:

$$\mathrm{grad\_S} = \sqrt{\left(\mathrm{grad}_x(S)\right)^2 + \left(\mathrm{grad}_y(S)\right)^2} \qquad (3)$$

Mit Hilfe dieses Indikators werden diejenigen Elemente verfeinert, in denen ein großer Sättigungsgradient vorliegt. Die Untersuchungen zeigen, dass dieser Indikator sowohl für advektions-dominante und diffusions-dominante Probleme als auch für Fragestellungen, bei denen beide Effekte auftreten, geeignet ist.

Massenkonservativität

Bei der h-adaptiven Methode kommen bei der Verfeinerung neue Knoten hinzu, für die Werte ausgehend von der vorhandene Lösung interpoliert werden müssen. Bei der Vergröberung fallen Knoten weg; dies bedeutet, dass die Lösungswerte, die für diese Knoten vorhanden waren, nun nicht mehr verfügbar sind. Aufgrund dieser Netzmanipulationen kann es zum Verschwinden oder Hinzukommen von (künstlicher) Masse kommen. Da für die Zeitdiskretisierung auch die Netze der Lösungen zum vorherigen Zeitschritt angepasst werden müssen, kann es zu einem Massenfehler kommen, der mit der Zeit akkumuliert. Massenkonservativität ist ein zentraler Punkt von Diskretisierungsverfahren. Um diese wichtige Eigenschaft nicht durch die Netzanpassung zu untergraben, wird ein Massenkorrektur-Algorithmus entwickelt, der den Erhalt der Masse im System sicherstellt. Ein repräsentatives Ergebnis ist in Abb. 3 dargestellt. Hier wird der Unterschied der Massen im System zur Masse der analytischen Lösung für das *Buckley-Leverett* Problem für Rechnungen mit und ohne Massenkorrektur gegenübergestellt. Für eine Rechnung mit einem gleichbleibenden Netz (uniformer Fall) ist die Massendifferenz konstant. Für eine adaptive Rechnung mit Massenkorrektur zeigt sich der gleiche Verlauf, während die Massendifferenz für einen adaptiven Fall ohne Korrektur im Laufe der Rechnung zunimmt. An den Sprüngen der Funktion kann abgelesen werden, dass die Rechnung selbst massenkonservativ ist (horizontaler Verlauf), während eine Änderung der Masse nur dann stattfindet, wenn auch das Netz angepasst wird.

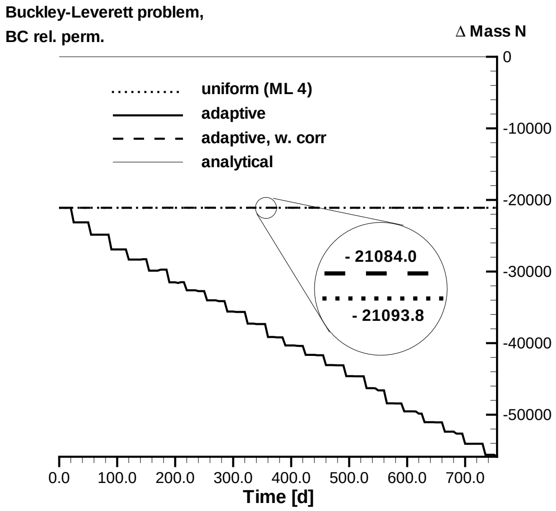


Abb. 3: Verlauf des Massenunterschiedes über die Zeit (*Buckley-Leverett* Problem)

Herleitung der Zwei-Phasen-*Peclet*-Zahl

Zusätzlich zu der beschriebenen orts-adaptiven Methode wird ein adaptives Verfahren angewendet, bei dem das Diskretisierungsschema angepasst wird. Da advektions-dominante Prozesse einer anderen Behandlung bedürfen wie diffusions-dominante Prozesse ('fully upwinding' (vollständiges Gewichten des oberstromigen Knotens) im Gegensatz zu einer 'zentral wichtenden' Methode), wird ein Kriterium hergeleitet, das es ermöglicht, zwischen diesen Prozessen zu unterscheiden. Hierfür wird die Zwei-Phasen-*Peclet*-Zahl *Pe* hergeleitet, die das Verhältnis von Advektion zu Diffusion beschreibt. Sie lautet für die $p_w - S_n$-Formulierung:

$$Pe = \frac{\dfrac{\partial \lambda_n^{IP}}{\partial S_{n|i}} \left((\mathbf{grad}\, p_n^{IP})^{\perp \mathrm{scvf}} - \rho_n^{IP} \mathbf{g}^{\perp \mathrm{scvf}} \right)}{\lambda_n^{IP} \dfrac{\partial (\mathbf{grad}\, p_c^{IP})^{\perp \mathrm{scvf}}}{\partial S_{n|i}}} \tag{4}$$

Diese Formulierung gilt für einen Integrationspunkt (IP). Im Laufe der Berechnung wird eine Element-*Peclet*-Zahl verwendet, die sich aus dem maximalen Absoluwert aller IP-*Peclet*-Zahlen je Element ergibt. Bei dem Einsatz des *Peclet*-Kriteriums wird eine 'fully upwinding' Methode gewählt, wenn gilt $Pe_{\mathrm{elem}} \geq 1.0$, ansonsten wird die Methode der zentralen Gewichtung angewendet.

Homogener Testfall

Beispielhaft für die untersuchten Testfälle werden hier die Ergebnisse für ein Sandbox-Problem vorgestellt, das sowohl Advektion als auch Diffusion im Gebiet enthält. Das Gebiet sowie die Anfangs- und Randbedingungen können Abb. 4 entnommen werden. Die Fluid- und Bodenparameter sind in Tab. 1 zusammengestellt. Für die Berechnung wird die 'fully upwinding' Methode eingesetzt. Abb. 5 zeigt das angepasste Netz und die Verteilung der Öl-Sättigung im Gebiet nach 24 Stunden. Wie zu sehen ist, löst das hier eingesetzte h-adaptive Verfahren den Bereich der Front sehr gut auf.

Anschließend wird das *Peclet*-Kriterium angewendet. Abb. 6 zeigt die Öl-Sättigungsverteilung und die Verteilung der *Peclet*-Zahl für eine Berechnung mit einem uniformen Netz. Wie zu erkennen ist, liegen die Werte für *Pe* im Bereich hinter der Front bei eins, während sie an der Front selbst stark zunehmen. In dieser Abbildung sind die Elemente, für die ein zentral-gewichtetes Schema angewendet wurde (α_{up} = 0.0) durch ein schwarzes Quadrat gekennzeichnet. Das Ergebnis dieser Simulation ist in Abb. 7 dem Ergebnis mit 'fully upwinding' gegenüber gestellt. Man kann sehen, dass die Wahl des Gewichtungsfaktors den Sättigungsverlauf beeinflusst.

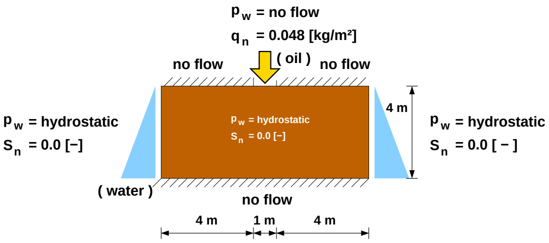


Abb. 4: System, Anfangs- und Randbedingungen des Sandbox-Problems

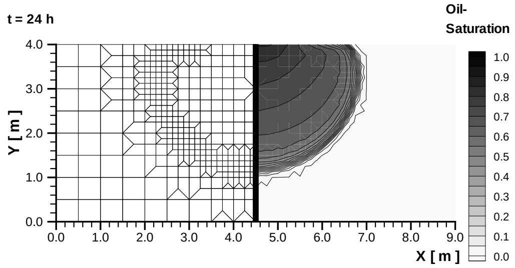


Abb. 5: Netz und Öl-Sättigungsverteilung nach 24 Stunden

Tab. 1: Fluid- und Bodeneigenschaften (homogener Testfall)

Fluideigenschaften:				
Dichte Wasser	ρ_w	=	1000	[kg/m^3]
Dichte Öl	ρ_n	=	1460	[kg/m^3]
dyn. Viskosität Wasser	μ_w	=	0.001	[kg/(m s)]
dyn. Viskosität Öl	μ_n	=	0.0057	[kg/(m s)]

Bodeneigenschaften:			
abs. Permeabilität	$\mathbf{K}$	=8 x 10^{-12}	[m^2]
Porosität	ϕ	=0.4	[-]
Residualsättigung Wasser	S_{wr}	=0.05	[-]
Residualsättigung Öl	S_{nr}	=0.0	[-]
Eindringdruck	p_d	=700.0	[Pa]
Porengrößenverteilungsindex	λ	=2.0	[-]

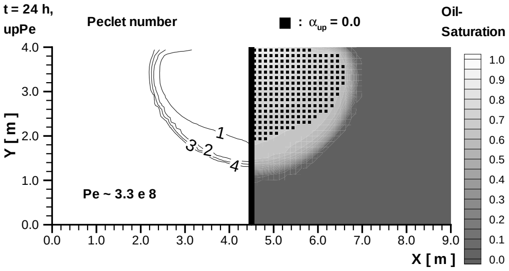


Abb. 6: Verteilung der *Peclet*-Zahl und Öl-Sättigung für den uniformen Fall. Die schwarzen Quadrate markieren die Elemente, in denen α_{up} = 0.0 gilt.

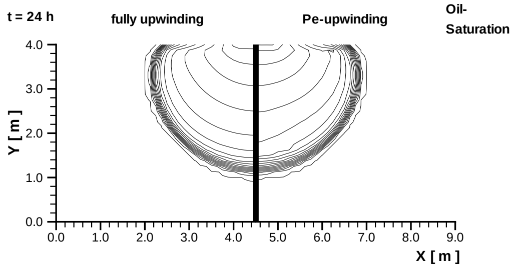


Abb. 7: Vergleich der Ergebnisse 'fully upwinding' und '*Peclet*-upwinding' für den uniformen Fall

Heterogener Testfall

Zusätzlich zu dem homogenen Testfall wird auch ein heterogener Testfall untersucht. Hierbei handelt es sich in der Geometrie um das oben beschriebene Sandbox-Problem, in das eine Linse mit einer anderen Durchlässigkeit als das umgebende Medium eingefügt ist. Die Abmessungen der Linse betragen 3 x 1 Meter und sie liegt zentriert einer Meter unterhalb des oberen Randes. Des Weiteren werden die Fluid- und Bodenparameter verändert (s. Tabellen 2 und 3). Die Ergebnisse der Rechnungen mit der Box-Methode und der Box-Trans-Methode sind in Abb. 8 gegenübergestellt. Der Unterschied wird deutlich: während das von oben einströmende Öl im Falle der Box-Methode in die Linse eindringen kann, staut sich das Öl im Falle der Box-Trans-Methode auf der Linse auf. Wenn jedoch für die Box-Methode der Bereich der Linse adaptiv verfeinert wird, kann das selbe Eindringverhalten wie mit der Box-Trans-Methode erreicht werden.

Auch für den heterogenen Fall wird eine Simulation mit dem *Peclet*-Kriterium durchgeführt (nur Box-Methode). Das Ergebnis für die h-adaptive Berechnung ist in Abb. 9 gezeigt. Wie deutlich zu erkennen ist, führt die Anwendung des Kriteriums zu negativen Sättigungen über der Linse, welches einen klar nicht-physikalischen Effekt darstellt. Hieraus lässt sich die Aussage ableiten, dass bei der Verwendung der Box-Methode sichergestellt sein muss, dass im Bereich von Heterogenitäten ein 'fully upwinding' Schema verwendet wird.

Tab. 2: Fluideigenschaften (heterogener Testfall)

Dichte Wasser	ρ_w	=	998	[kg/m^3]	dyn. Visk. Wasser	μ_w	=	0.001	[kg/(m s)]
Dichte Öl	ρ_n	=	1631	[kg/m^3]	dyn. Visk. Öl	μ_n	=	0.0009	[kg/(m s)]

Tab. 3: Bodeneigenschaften (heterogener Testfall)

	Gebiet 1					Gebiet 2 (Linse)			
abs. Permeabilität	$\mathbf{K}_1$	=	5.04 x 10^{-10}	[m^2]	abs. Permeabilität	$\mathbf{K}_2$	=	5.26 x 10^{-11}	[m^2]
Porosität	ϕ_1	=	0.4	[-]	Porosität	ϕ_2	=	0.4	[-]
Res.-Sätt. Wasser	$S_{wr,1}$	=	0.078	[-]	Res.-Sätt. Wasser	$S_{wr,2}$	=	0.098	[-]
Res.-Sätt. Öl	$S_{nr,1}$	=	0.0	[-]	Res.-Sätt. Öl	$S_{nr,2}$	=	0.0	[-]
Eindringdruck	$p_{d,1}$	=	370.0	[Pa]	Eindringdruck	$p_{d,2}$	=	1324.0	[Pa]
Vert.-Index	λ_1	=	3.86	[-]	Vert.-Index	λ_2	=	2.49	[-]

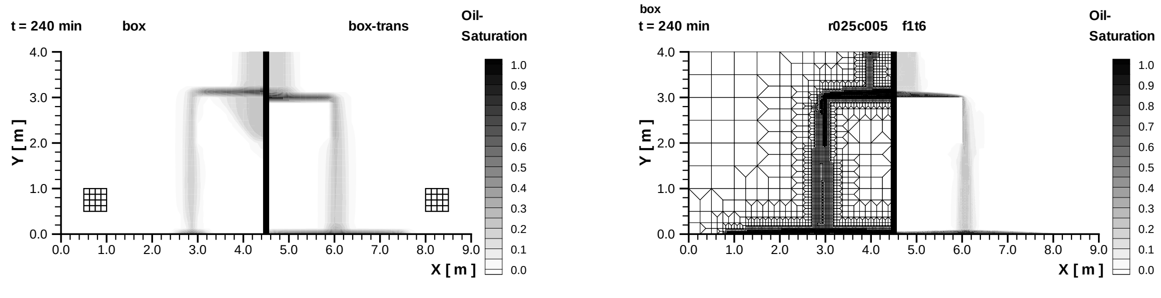


Abb. 8: Links: Vergleich zwischen Box- und Box-Trans-Methode. Rechts: Netz und Öl-Sättigungsverteilung für die adaptive Simulation (Box-Methode)

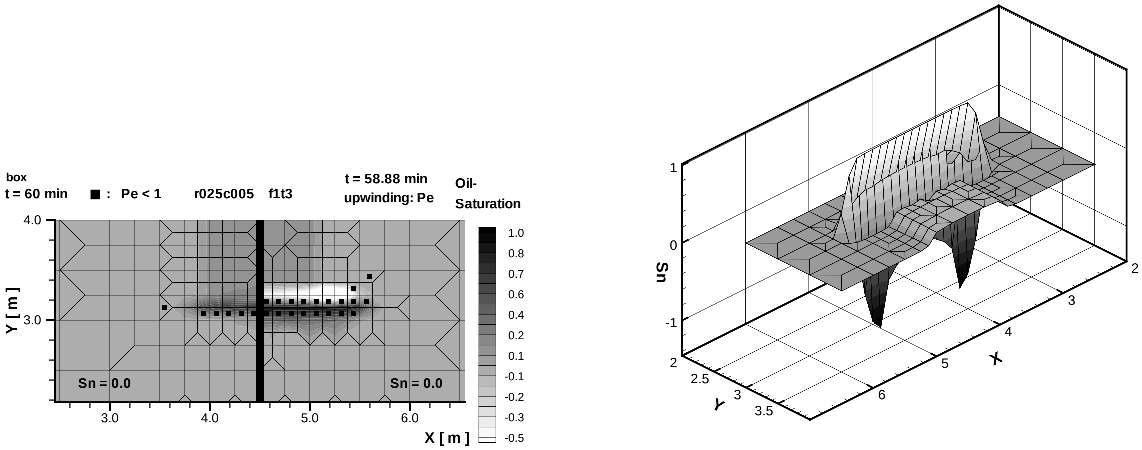


Abb. 9: Links: Vergleich der Öl-Sättigungsverteilung für die adaptive Methode mit 'fully upwinding' und '*Peclet*-upwinding' (Ausschnitt). Rechts: Detail der Öl-Sättigung über der Linse für den Fall '*Peclet*-upwinding'.

Anwendungsbeispiel

Im Folgenden wird die h-adaptive Methode für die Simulation eines Experimentes verwendet, das von der Autorin an der ETH Zürich, Institut für Hydromechanik, durchgeführt wurde. Auf die Anwendung des entwickelten *Peclet*-Kriteriums wird an dieser Stelle verzichtet, da die im Rahmen der Arbeit erarbeiteten Ergebnisse zeigen, dass noch weitere Untersuchungen über den genauen Einsatz dieses Kriteriums notwending sind.

Bei dem Experiment wird das Modell eines Deiches mit einem geneigten Kern in einen *Hele-Shaw*-Aufbau eingebaut. Der Ölstand auf der einen Seite des Modells wird erhöht und somit der Durchfluss durch den Deich nachgebildet. Das Ergebnis des Experimentes (stationärer Zustand) ist in Abb. 10 gezeigt. Das Netz und die Sättigungsverteilung der numerischen Simulation sind in Abb. 11 dargestellt. Es lässt sich erkennen, dass die Simulation das Experiment-Ergebnis sehr gut nachbildet. Die scharfe Front (Trennlinie zwischen Öl und Luft) sowie der Bereich, in dem das Öl aus dem Kern austritt, sind deutlich zu erkennen. Die Darstellung des Netzes zeigt, dass die Sättigungsfront sehr gut mit feinen Elementen aufgelöst wird.

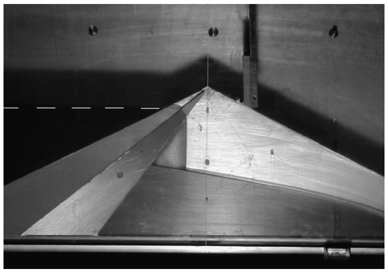


Abb. 10: Experiment im *Hele-Shaw*-Aufbau, stationärer Zustand (nach *Paul et al.* [56])

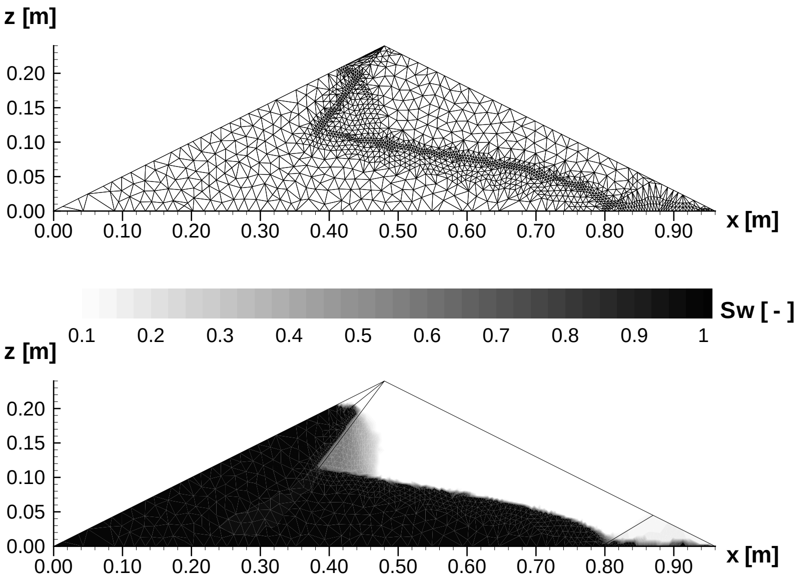


Abb. 11: Netz (oben) und Sättigungsverteilung (unten) der adaptiven Simulation

Zusammenfassung

Für die Zwei-Phasen-Strömung in porösen Medien werden adaptive Methoden hergeleitet und vorgestellt. Diese Methoden beinhalten zum einen eine automatische Anpassung der Elementgröße des Berechnungsnetzes in Bereichen, in denen der Gradient der Sättigung sehr steil ist (h-adaptive Methode mit Gradienten-Indikator). Zum anderen wird eine Methode erarbeitet, mit der es möglich ist, in Bereichen, in denen Advektion dominiert (gekennzeichnet durch eine Zwei-Phasen-*Peclet*-Zahl größer eins), ein 'fully upwinding' Diskretisierungsschema zu verwenden, während in Bereichen, in denen Diffusion vorherrscht (*Pe* kleiner eins) ein 'zentral gewichtetes' Schema zum Einsatz kommt.

Eine Anwendung der erarbeiteten Methoden findet für homogene und heterogene Testfälle statt. Hierbei ist festzustellen, dass die vorgestellte h-adaptive Methode sehr gut die gestellten Anforderungen erfüllt. Es muss sich jedoch bewusst gemacht werden, dass durch (stark) unterschiedliche Elementgrößen im Gebiet die Schwierigkeit der Behandlung der nichtlinearen Effekte der Zwei-Phasen-Gleichungen unter Umständen noch vergrößert wird.

Zu der Anwendung des *Peclet*-Kriteriums auf die Wahl der Diskretisierungsmethode ist zu bemerken, dass noch weitere Untersuchungen notwendig sind, um die vorgestellte Methode effizient einsetzen zu können.