Contributions to the large-scale Simulation of Flow and Transport in Heterogeneous Porous Media

Olaf Ippisch

Habilitationsschrift

vorgelegt an der Fakultät für Mathematik und Informatik der
Ruprecht-Karls-Universität Heidelberg

zur Erlangung der Venia legendi im Fachgebiet

Informatik

To my wife
Vera
and
my children
Ronja and Florian

Contents

List of Figures

List of Tables

List of Algorithms

Nomenclature

Scalar values, functions and sets are denoted by normal letters (e.g. $p_l, k_r, S_i, \ldots$). Vectors are typeset in boldface italic roman symbols (e.g. $\boldsymbol{x}$), whereas tensors are written in boldface sans serif letters (e.g. $\boldsymbol{\mathsf{K}}$).

Uppercase Latin Letters

$\mathcal{A}_{ij}$	Flux contribution for face j of element i
$\mathcal{E}_h$	Partitioning of the domain $\mathit{\Omega}$
$\mathcal{H}$	Hessian
$\mathcal{J}$	Jacobian
$\mathcal{M}_i$	Storage contribution for element i
$\mathcal{Q}_i$	Sink contribution for element i
$\mathcal{T}$	Time domain
$\boldsymbol{\mathsf{D}}_s$	Dispersion coefficient [m^2 s^{-1}]
$\boldsymbol{\mathsf{D}}_{s,m}$	Diffusion coefficient for solute in aqueous solution [m^2 s^{-1}]
$\boldsymbol{\mathsf{K}}$	Permeability tensor [m^2]
$\boldsymbol{\mathsf{K}}_s$	Saturated hydraulic conductivity [m/s]
$\boldsymbol{\mathsf{K}}_{nw}$	[m^2 Pa^{-1} s^{-1}]
$\boldsymbol{F}_i$	Body force per unit volume acting on phase i [N m^{-3}]
C	Average solute concentration [mol m^{-3}]
$D^w_{g,\mathsf{atm}}$	Diffusion coefficient for water vapour in air [m^2 s^{-1}]
D^w_n	Diffusion coefficient for water vapour [m^2 s^{-1}]
$D_{s_{\mathsf{eff}}}$	Effective (molecular) diffusion coefficient in porous media [m^2 s^{-1}]
E_{nw}	Net rate of production of interfaces[m^{-1} s^{-1}]
$J(S_w)$	Leverett J-function [–]

J_s	Numerical flux function for solute transport
$J_s(U_i, U_j)$	Numerical flux function
$J_{s,\mathsf{conv}}$	Numerical flux function for convection
$J_{s,\mathsf{diff}}$	Numerical flux function for diffusion
K	Number of time steps [–]
K_n^w	Equivalent conductivity for water vapour transport [kg m^{-2} s^{-1}]
K_F	Freundlich adsorption coefficient
K_L	Langmuir adsorption coefficient [m^3 mol^{-1}]
K_s	Linear adsorption coefficient [m^3 kg^{-1}]
M_i	Mass exchange of phase i with other phases [kg m^{-3} s^{-1}]
P_d	Pore blockage factor [–]
R	Ideal gas constant [J mol^{-1} K^{-1}]
R	Retardation factor [–]
R_s	Solute sink density [mol m^{-3} s^{-1}]
R_w	Average sink term for element i
R_{ij}	Viscous drag coefficient between phase i and j [kg s^{-1} m^{-3}]
S_{drain}	Main drainage curve [–]
S_{eff}	Effective saturation [–]
S_{imb}	Main imbibition curve [–]
S_{max}	Maximal wetting phase saturation $(1 - S_{n_r})$ [–]
S_{mob}	Mobile fraction of the wetting phase $(1 - S_{n_r} - S_{w_r})$ [–]
S_{ref}	Wetting phase saturation at reversal point [–]
S_i	Saturation of phase i [–]
S_{n_r}	Residual non-wetting phase saturation [–]
S_{w_r}	Residual wetting phase saturation [–]
T	Temperature [K]

T_{pot}	Potential transpiration rate [kg^3 s^{-1}]
U^k	Average matric potential in time step k $[m/Pa/m^2s^{-2}]$
U_i	Cell average of ψ_m on element e_i
V_m^w	Molar volume of liquid water [m^3 mol^{-1}]
W_h	Broken polynomial space of order zero

Lowercase Latin Letters

$\boldsymbol{e_z}$	Unit vector in the vertical direction [–]
$\boldsymbol{f}$	Right-hand side of linear equation system in the Newton scheme
$\boldsymbol{j_i}$	Mass flux density [kg m^{-2} s^{-1}]
$\boldsymbol{j}$	Volumetric flux density of the liquid phase [m s^{-1}]
$\boldsymbol{j}_s$	Molar solute flux density [mol m^{-2} s^{-1}]
$\boldsymbol{j}_{s_{\mathsf{conv}}}$	Molar solute flux density due to convection [mol m^{-2} s^{-1}]
$\boldsymbol{j}_{s_{\mathsf{disp}}}$	Molar solute flux density due to dispersion [mol m^{-2} s^{-1}]
$\boldsymbol{m}_i$	Momentum transfer between phase i and other phases [N m^{-3}]
$\boldsymbol{n}$	Unit outer normal vector
$\boldsymbol{v}$	Pore water velocity [m s^{-1}]
$\boldsymbol{v}^{k,m}$	Correction in step m of the Newton scheme in time step k
$\boldsymbol{v}_i$	velocity of phase i [m s^{-1}]
$\boldsymbol{w}_{nw}$	Average velocity of the fluid-fluid interfaces [m s^{-1}]
$\boldsymbol{x}$	Vector of measurement points (optimisation)
$\boldsymbol{y}$	Vector of measurement values (optimisation)
∂e_i	Boundary of element i
a_{nw}	Specific area of the wetting/non-wetting phase interface [m^{-1}]
c_s	Solute concentration in the liquid phase [mol m^{-3}]
c_{max}	Langmuir maximal adsorbed concentration [m^3 kg^{-1}]
$c_{s_{\mathsf{sorb}}}$	Adsorbed solute concentration [mol kg^{-1}]

e_i	Element i
f_i	Weighting factor in the multiple van Genuchten model [–]
g	Acceleration of gravity [m s^{-2}]
h	Mesh width [m]
k_{eff}	Effective hydraulic conductivity [kg m^{-2} s^{-1}]
k_{r_i}	Relative permeability of phase i [–]
m	Van Genuchten Parameter [–]
n	Freundlich exponent [–]
n	Van Genuchten Parameter [–]
p^w_{sat}	Partial pressure of water vapour over pure liquid [Pa]
p_c	Capillary pressure [Pa]
p_c^{eq}	Capillary pressure in hydraulic equilibrium [Pa]
p_i	Pressure of phase i [Pa]
q_w	Volumetric sink term [–]
r_i	Sink term for phase i [kg m^{-3} s^{-1}]
r_r	Sink term due to root water uptake [m^2 s^{-1}]
r_s	Compensation factor for root water uptake [–]
r_s	Sink density for solute [mol m^{-3} s^{-1}]
$r_{r_{\mathsf{pot}}}$	Potential sink term due to root water uptake [m^2 s^{-1}]
$r_{s_{\mathsf{max}}}$	Maximal compensation factor for root water uptake [–]
t_0	Initial time [s]
t_{end}	Final time [s]
t_k	Discrete point in time [s]
u_{n_p,w_u}	Energy density stored in n_p/w_u phase interface [J m^{-3}]
u_{w_p,n_u}	Energy density stored in w_p/n_u phase interface [J m^{-3}]
w_i	Weighting function i

w_{ij}	Water exchange between flow domains [kg m^{-3} s^{-1}]
x	Cartesian coordinate in a metric system [m]
y	Cartesian coordinate in a metric system [m]
z	Cartesian coordinate in a metric system [m]

Uppercase Greek Letters

$\bar{\mathit{\Omega}}$	Closed domain
$\mathbf{\Sigma}_i$	Stress tensor in phase i [N m^{-2}]
$\mathit{\Omega}$	Open domain
$\partial\mathit{\Omega}$	Boundary of the domain
$\mathit{\Phi}$	Porosity [–]
$\mathit{\Psi}_{ia}$	Material property for water flow due to gradients in a_{nw} [Pa m]
$\mathit{\Psi}_{iS}$	Material property for water flow due to saturation gradients [Pa]
$\mathit{\Psi}_{nw}$	Material property for interface transport due to water saturation gradients [Pa]
$\mathit{\Theta}$	Contact angle [–]

Lowercase Greek Letters

α	Van Genuchten Parameter [m^{-1}]
α_{ij}	Exchange rate for water transfer between flow domains [kg m^{-3} s^{-1} Pa^{-1}]
χ	Length scale in Miller similarity [m]
η	Effective saturation in the Barenblatt model [–]
η	Scaling parameter in Miller similarity [–]
κ	Mualem integral in the multiple van Genuchten model [–]
λ	Brooks-Corey parameter [–]
λ	Levenberg-Marquardt parameter (optimisation)
λ_i	Line search parameter [–]
λ_l	Longitudinal dispersivity [m]
λ_t	Transversal dispersivity [m]

$\boldsymbol{\beta}$	Parameter vector (optimisation)
μ_i	Dynamic/kinematic viscosity $[Pas]$
ν_n^w	Molar density of water vapour in the gas phase [mol m^{-3}]
$\nu_{s_{\mathsf{sorb}}}$	Adsorbed molar density [mol m^{-3}]
ϕ	Base function for Hermite spline [–]
ϕ_N	Normal flux density at Neumann boundary [kg m^{-2} s^{-1}]
ψ	Soil water potential $[m/Pa/m^2s^{-2}]$
ψ_D	Potential at Dirichlet boundary $[m/Pa/m^2s^{-2}]$
ψ_g	Gravity potential $[m/Pa/m^2s^{-2}]$
ψ_m	Matric potential $[m/Pa/m^2s^{-2}]$
ψ_o	Osmotic potential $[m/Pa/m^2s^{-2}]$
ψ_{oo}	Potential at lower end of optimal root water uptake $[m/Pa/m^2s^{-2}]$
ψ_{sat}	Potential at which root water ceases due to lack of oxygen $[m/Pa/m^2s^{-2}]$
ψ_{uo}	Potential at upper end of optimal root water uptake $[m/Pa/m^2s^{-2}]$
ψ_{wilt}	Potential at wilting point $[m/Pa/m^2s^{-2}]$
ψ_{m_0}	Air entry value [Pa]
$\psi_{m_{\mathsf{rev}}}$	Matrix potential at reversal point $[m/Pa/m^2s^{-2}]$
ρ_i	Density of phase i [kg m^{-3}]
ρ_r	Root density [m^{-3}]
ρ_s	Bulk density of the soil [kg m^{-3}]
ρ_{ij}	Correlation coefficient
σ_i^k	Reconstructed slope of solute concentration distribution in element i at time t^k [mol m^{-4}]
σ_w	Surface tension of wetting fluid [N m^{-1}]
σ_{nw}	Macroscale interfacial tension [N m^{-1}]
τ	Relaxation time [s]
τ	Tortuosity [–]

τ^k	Time step in interval $[t_k : t_{k+1}]$ [s]
θ_w	Volumetric water content [–]
$\theta_{w_{\mathsf{eq}}}$	Water content at equilibrium in the Ross and Smettem model [–]
θ_{w_r}	Residual volumetric water content of the wetting phase [–]
θ_{w_s}	Saturated volumetric water content of wetting phase [–]
ξ	Base function for Hermite spline [–]
ξ	Tortuosity coefficient [–]

Subscripts

n	Non-wetting phase
n_p	Percolating non-wetting phase
n_u	Non-percolating non-wetting phase
s	Solid phase
w	Wetting phase
w_p	Percolating wetting phase
w_u	Non-percolating wetting phase

Superscripts

k	Time step index
k, m	Iteration m in time step k

Preface

The simulation of porous media requires an interdisciplinary approach. While the models describing the flow process and the properties of porous media are 50 to 100 years old, there are many open questions and pitfalls. Hysteresis is still widely ignored, which is also a consequence of unsatisfactory hysteresis models. Hydraulic non-equilibrium seems to occur not only in fast processes, but has also recently been shown in rather slow multi-step flux experiments.

On the other hand, flow in porous media is also interesting from a mathematical point of view. Nearly all types of first and second-order partial differential equations occur. Many discretisation schemes have been developed to solve the two-phase flow equations and their advantages and disadvantages are still not fully understood. Richards' equation has for a long time attracted less attention, but this has changed in the last years.

Finally, porous media are also interesting for the computer scientist. Due to their natural heterogeneity and the strong non-linearity of their parameter functions, the use of massively parallel computers is necessary for further progress. The development of simulation software which is user friendly, efficient, scalable and maintainable is not an easy task.

In the last decade I had the honour of working and cooperating with excellent scientists. To list all of them is not possible. Thus I hope that everybody who reads this rather feels himself included instead of left out.

Above all I want to thank my family for their patience during long days and nights I spent at the computer and not with them. I want to thank them for their encouragement and their support and I want to thank my parents for encouraging me to pursue a path to higher education.

Peter Bastian has introduced me into the art of scientific computing and numerical mathematics and at the same time gave me room for my own developments. Kurt Roth has shaped my understanding of the physics of porous media. Hans-Jörg Vogel is not only a creative and passionate scientist with whom I had many lively discussions, he has also became a close friend. Ingrid Hellwig has helped me with many practical problems in the last years. Without them this work would not have been possible.

There are some people who have contributed in particular. Felix Heimann has parallelised $\mu\varphi$ during a software practical. Markus Blatt, the developer of the ISTL library and especially its algebraic multigrid preconditioner, has helped me in the optimal usage of the solver and has optimised the AMG during the scalability tests on JUGENE. Jö Fahlke has implemented the parallel transient solute transport model based on a previous version of Peter Bastian. Hans-Jörg Vogel and Steffen Schlüter wrote the structure generator used for the virtual soil systems. Steffen performed the simulations of the two-dimensional virtual soil systems and Hans-Jörg the simulations of the dynamic systems.

Together with them, as well as with Wolfgang Durner, Jan Vanderborght and Kurt Roth we found after long discussions hydraulic parameter sets for the virtual soil systems. Jan Vanderborght helped in acquiring computation time on JUGENE. Alexandra Herzog implemented the Hermite splines during her diploma thesis and included the singular value decomposition into the parameter estimation code. The Jülich Supercomputing Center and Wolfgang Frings in particular provided excellent support and gave us the opportunity to test the scalability of our code. The Dune Project develops great software and I profited from their parallel communication sub-system, but also suffered under the build system. To all of them my sincere thanks.

I also had the pleasure to act as a so-called “mentor” for some Ph.D. students and I had many interesting discussions with Pavel Hron, Ole Klein, Rebecca Neumann, Sven Marnach, Steffen Müthing, Adrian Ngo, Jurgis Pods, Dan Popovic, Anatja Samouëlian and Klaus Schneider-Zapp. I also want to mention here my former colleagues Christian Engwer, Volker Reichenberger and Uli Weller.

Heidelberg, January 2014 Olaf Ippisch

1. Introduction

1.1. Definitions

A porous medium can be generally defined as a solid material which only fills part of the total space and thus has void spaces included. Furthermore this void spaces (called the pore space) are distributed over the whole medium. For transport to be possible in the porous medium, the pore space has to be interconnected.

If the porous medium is not in vacuum it is filled with a gas phase and/or one or more immiscible liquid phases. Bear and Bachmat 1991 define a phase as a chemically homogeneous portion of a system under consideration which is separated from other such portions by a definite physical boundary. In a single-phase system the whole pore space is filled with a single gas or fluid. Each of the phases can consist of one or more components, i.e. of one or more distinct chemical species. The same components can be present in more than one phase. If atoms or molecules are crossing the boundary between two phases a phase change occurs.

The solid phase itself can be chemically uniform (mostly in technical porous media) or be a composite of different materials. The properties of porous media result from the chemical nature of the solid phase, the geometry of the pore space and the surface properties of the solid phase, which are often different from the properties of the bulk solid phase.

1.2. Relevance of Porous Media Research

Porous media defined in this general sense are of central importance for life on earth. Even the cell itself with its cytoskeleton can be interpreted as a (very complex, high porosity) porous medium. Bones also form a porous structure. However, in this study the focus is more on larger scale applications and in porous media which are not part of living organisms. The theory of porous media flow was developed for applications in hydrology, agriculture and civil and petrol engineering and this is still the main area of research today.

Whith the world population still growing, water is one of the central scarce resources on our planet. It is needed for drinking water and irrigation as well as for industrial production. According to the Federal Institute for Geosciences and Natural Resources (BGR) 2014 more than 70 per cent of the public drinking water production in Germany are from groundwater reservoirs. Agriculture is consuming 70 per cent of the global freshwater production. In the least developed countries it is even up to 90 per cent. At the same time soil salinity is an imminent threat (World Water Assessment Programme

2012). Thus the optimal use of irrigation water is a topic of intense research (e.g. Zhao et al. 2012; McCarthy et al. 2013; Boutraa 2010; Lorite et al. 2012). However, intensive agriculture is also a threat for groundwater quality. Precision agriculture is trying to optimise the use of pesticides and fertilisers not only to obtain maximal output at minimal costs, but also to reduce groundwater pollution (Gebbers and Adamchuk 2010; Inoue et al. 2012).

Soils and groundwater are also under threat of contamination e.g. with oil and other hydrocarbons. Environmental remediation is therefore a major research area including the surfactant enhanced ”pump and treat” approach (Mulligan et al. 2001), soil vapour extraction (Class and Helmig 2002), injection of nano particles (Zhang 2003) or bioremediation (Boopathy 2000). As industrial societies produce huge amounts of (partially toxic) waste, the security assessment of landfills and nuclear waste repositories is important as well (e.g. Tsang et al. 2012).

Porous media are also central to many topics in the field of energy production. Petrol engineering is getting a more and more complicated matter. A recent development is the massive increase of the production of shale gas and shale oil with fracking (Gassiat et al. 2013). But geologic formations are also investigated as possible storage sites for carbon dioxide (Neumann et al. 2013). The domestic use of geothermal energy for heating requires more intensive planning, as the density of these facilities increases. It is also a potential source of electrical energy (Bayer et al. 2013).

Soil atmosphere interactions are highly relevant for global climate (Seneviratne et al. 2010). With models for weather and climate prediction getting better, an improved representation of soils is crucial for further improvement. The large scale and the large grid block size are a big challenge for science. Soils are not only a store for energy and water, but a potential source for greenhouse gases as well.

Finally, porous media have also important technical applications, e.g. in catalysts and fuel cells.

1.3. The Multi-Scale Nature of Geosystems

Geosystems are – as a consequence of their genesis and history – heterogeneous on many different scales. The processes producing these heterogeneous structures are manifold. In magmatic rocks they are mainly a consequence of different magma composition and cooling, in sediments they are due to the sedimentation history and in metamorphic rocks this structure is altered again by pressure and heat. Folding processes, faults, fractures and karst formation, mineral dissolution and precipitation contribute to the complexity of natural porous media. In soils the heterogeneity of the parent material is reshaped by soil forming processes like frost and chemical weathering, cryoturbation, bioturbation by mobile organisms and roots, erosion by wind and water, and last but not least by human impact e.g. by ploughing.

Thus geosystems and especially soils are inherently multi-scale systems with many different levels of heterogeneity. Multi-scale systems are a topic of current research in many different fields, e.g. research on steel (EXASTEEL 2014) and in hydrogeology.

Mathematical theories for multi-scale analysis have been successfully developed and applied e.g. for groundwater flow problems (Jenny et al. 2003). However, there are special aspects which make the situation more difficult in multi-phase flow systems and especially in soils. First of all multi-phase flow in contrast to single-phase flow is described by non-linear partial differential equations. Unfortunately, averages of non-linear functions have different properties than those of linear functions. The relationship between the average of two quantities which are related by a non-linear function, is typically non-unique and shows phenomena like hysteresis. One consequence of non-linear processes is, that there is not necessarily a smoothing of small-scale processes. They still can have significant consequences on much larger scales. Additionally, while in soils the main flow direction is vertical, the different soil types are located besides each other. Thus the processes are at most – if at all – weakly coupled and can operate on very different time scales. Even in the (vertical) main direction of transport soil horizons are very thin and processes often do not reach an asymptotic limit.

1.4. Scales

An important aspect of porous media research is the existence of several spatial (and related temporal) scales which are involved. In this work we want to distinguish four scales: the pore scale, the laboratory scale, the field scale and the landscape scale. Our capabilities to describe the processes on each of these scales with appropriate transport equations differs. The possibility to perform experiments is limited on some scales and therefore our knowledge and uncertainty about the system is scale-dependent.

1.4.1. Pore Scale

At the pore scale (micrometre to centimetre) the spatial distribution of all phases can be resolved. Only one phase may be present at each point in space and time. The physical laws are in principle well known, but they are not always easy to apply. The geometry of the pore space is often complicated. This makes numerical simulations very challenging. Additionally, the solid phase can be composed of different materials. In practice even (micro-)organisms might be important. The pore space geometry today can be measured with very high accuracy using X-ray micro-tomography or neutron tomography. It is even possible to distinguish the fluid and the gas phase. Experiments are difficult to perform and usually are only possible in simplified geometries.

1.4.2. Laboratory Scale

At the laboratory scale (centimetre to metre) it is no longer possible to resolve single pores (except macropores). The main constraints are the maximal sample size in micro-tomography and the extreme high storage requirements of such a representation. Porous media at this scale are described in a continuum representation, where fractions of every phase are present at any point in space and time. Transport laws are rather well known, but there are limitations especially for fast processes. This is the scale of most experiments, where the initial and boundary conditions can be very well controlled and the reaction of the system can be measured with sometimes surprisingly high resolution in space and time.

1.4.3. Field Scale

The main difference between laboratory and field scale (one to one hundred metre) is, that it is much harder to determine subsurface structures, (hydraulic) parameters and state variables, or to control the boundary conditions of the system. Thus the uncertainty about the state and the future of the system is growing significantly. As long as the properties of the porous medium are more or less uniform, a description with the same transport laws as used in the laboratory is often possible. However, there are difficulties to transfer parameters determined in the laboratory to the field. At the field scale many porous media start to become interesting for practical applications (e.g. agriculture, remediation).

1.4.4. Landscape Scale

The length scale of catchments to landscapes or even to the whole earth surface (one kilometre to thousands of kilometres) is the scale of interest for many important applications (groundwater management, flood control, oil production, CO_2 sequestration, weather and climate prediction. . . .). At this scale it is often neither possible to resolve the heterogeneity of the system nor to assume homogeneity with good conscience. Even with the most powerful computers available the grid resolution is too coarse. One example are land surface applications, where each grid cell includes very different soil types. In reservoir modelling it is not possible to resolve fractures or the detailed caprock topography. Thus the uncertainty on this scale is growing drastically. Effective transport equations which could give a correct averaged representation on the large grid blocks are essentially unknown. Multi-phase flow equations or simplified forms like Richards' equation are still used and the parameters are estimated using data assimilation and parameter estimation techniques. Experimental capabilities on this scale have been rather limited in the past. However, remote sensing offers new possibilities for observation.

1.5. Challenges

The different scales described before, the different accessibility of each scale to physical description and experiment, and the typically large scale applications require the use of scientific computing to bridge the scale gap and come to reliable predictions. Simple multi-scale approaches do not seem sufficient due to the non-linearity of the system and the ignorance about small scale features. For the time being it is therefore necessary to simultaneously conduct separate research projects on different scales with their respective problems. Due to the multi-scale heterogeneity it is necessary to simulate systems with complex parameter distributions on any given scale. Thus the use of massively parallel computing is mandatory.

1.6. Scope of this Work

In this work we want to describe some contributions to the description and mainly to the parallel simulation and parameter estimation at the continuum scale. The main focus is on the robust and efficient simulation of (unsaturated) water and solute transport in strongly heterogeneous porous media.

After a review of the physical description of two-phase flow (which is focused on the liquid phase), a robust numerical approach for the solution of Richards' equation will be developed and its implementation as a user-friendly and scalable software tool will be described. As parameters in porous media are often unknown, a framework for parameter estimation will be developed.

The second part of this work concentrates on the massively parallel solution of the convection-diffusion equation with transient fields of volumetric water content and flux density, calculated with the water transport code developed before.

Applications at different scales from the laboratory sample to the landscape scale will be presented in the third part. The first application assesses the possibility to predict solute transport through an undisturbed soil core using small scale hydraulic parameters estimated from multi-step outflow experiments, and the internal structure of the soil sample measured by X-ray tomography. The next two applications investigate the possibility to represent heterogeneous soils with one-dimensional effective models at the field scale. First high-resolution three-dimensional virtual soil systems are created. Then dynamic effects originating from macroscopic heterogeneity are studied.

In the last chapter thorough weak scalability tests of the developed codes are performed using two supercomputers of the BlueGene series. The test does not only test the scalability of the calculations but also investigates massively parallel file transfer. With simulations using up to 917'504 threads the scalability tests ideally combine the two meanings of large-scale simulation in computer science (many parallel processes) and geoscience (large domains).

Part I.

Water Flow in Porous Media

2. Continuum Scale Modelling of Water Flow in a Two-Phase System

2.1. Standard Theory

2.1.1. Two-Phase Flow Equations

The traditional theory of two-phase flow in porous media was originally not derived from pore scale equations. It is mainly based on empirical findings following the ideas of Darcy 1856 on single-phase flow and later considerations by Buckingham 1907, Wyckoff and Botset 1936, Muskat and Meres 1936 and Leverett 1940a. It can be developed starting with mass conservation equations for the wetting and non-wetting phase (neglecting phase change at the moment):

$$\frac{\partial\left(\varPhi S_i \rho_i\right)}{\partial t} + \nabla \cdot \boldsymbol{j_i} + r_i = 0 \qquad i = w, n \tag{2.1}$$

Here $\varPhi$ is the porosity, S_i is the saturation/volumetric fraction of phase i, ρ_i is the density, $\boldsymbol{j_i}$ the mass flux density and r_i is a sink term for phase i.

The flux laws follow a simple first order approach with the phase pressure difference and gravity as driving force, multiplied with a saturation-dependent material coefficient which includes both properties of the fluid and of the porous medium:

$$\boldsymbol{j_i} = -\rho_i \mathsf{K} \frac{k_{r_i}(S_i)}{\mu_i} \; \left(\nabla p_i - \rho_i g \boldsymbol{e_z}\right) \tag{2.2}$$

K is the absolute permeability tensor of the material, $k_{r_i}(S_i)$ the (usually scalar) relative permeability function, μ_i the dynamic/kinematic viscosity, p_i the pressure of phase i, g the constant of gravity and $\boldsymbol{e_z}$ the unit vector in the direction of gravity. The vertical axis points in the direction of gravity.

According to Hilfer 2006b these flux laws can be derived from momentum balance equations assuming that the fluids are incompressible, the stress tensors of the fluids are diagonal, the inertial term is negligible and the body forces are given by gravity. It is further necessary to assume that the momentum transfer between the fluids and the rigid matrix is governed by viscous drag which is proportional to the phase velocities $\boldsymbol{j_i}$ with a proportionality constant given by $\mathsf{K}^{-1}\frac{\mu_i \varPhi^2 S_i^2}{k_{r_i}(S_i)}$. A further (not completely necessary) condition is the macroscopic homogeneity of the porous medium. These are rather strong conditions and are not generally fulfilled. However, the theory has been successfully applied in numerous studies over the last decades. Alternative approaches will be discussed in section 2.2.

The combination of the flux laws and the continuity equations gives two equations for four unknowns (the two phase saturations S_i and the two phase pressures p_i). One unknown can be eliminated by the assumption that the two phases together always fill the full pore space:

$$\sum_{i=w,n} S_i = 1. \tag{2.3}$$

The equation system is then closed by the assumption that there is a unique relationship between the difference of the phase pressures and the saturation called the capillary pressure-saturation curve

$$p_n - p_w = p_c(S_w). \tag{2.4}$$

This assumption is experimentally well supported for static systems and slow flow conditions.

2.1.2. Richards' equation

Under the condition that the non-wetting phase is mobile enough to always be (almost) at the same pressure, the two equations can be decoupled. If only the flow of the wetting phase is of interest, it is then sufficient to solve a scalar equation, the so called Richards' equation[1].

$$\frac{\partial\left(\Phi S_w \rho_w\right)}{\partial t} - \nabla \cdot \left[\rho_w \mathbf{K} \frac{k_{r_w}(S_w)}{\mu_i} \quad (\nabla p_w - \rho_w g \boldsymbol{e_z})\right] + r_w = 0 \tag{2.5}$$

For a system where the wetting phase is water and the non-wetting phase is air, it has been shown by numerical tests that this assumption is justified as soon as both phases exist and are continuous (e.g. Ippisch 2001).

In soil physics, where this equation is generally used, it is common to write the equation in terms of the volumetric water content $\theta_w = \Phi S_w$ and the soil water potential ψ. It is defined as the energy needed to transfer an infinitesimally small amount of water from the porous medium to free pure water at a reference height. The advantage of the soil water potential is that it is well defined and includes more possible driving forces than just the phase pressure. The potential is a sum of different contributions e.g. the osmotic potential ψ_o, the gravity potential $\psi_g = \int_{z_0}^{z} \rho_w(x) g \boldsymbol{e_z} dx$ and the matric potential ψ_m, which takes into account the interactions between water and the porous medium. We will use the matric potential in a (common) wider definition, in which positive values are allowed for ψ_m under saturated conditions, corresponding to positive water pressures.

If the potential is expressed as energy per unit volume, the dimensions of the soil water potential are the dimensions of a pressure. The negative of the matric potential can

[1] named after L. A. Richards who proposed the equation in 1931 (Richards 1931)

then be identified with the difference between atmospheric pressure and water pressure, i.e. the capillary pressure. If we neglect osmotic effects and use that $\nabla\psi_g = -\rho_w g \boldsymbol{e_z}$ we obtain

$$\frac{\partial\left(\theta_w\rho_w\right)}{\partial t} - \nabla\cdot\left[\rho_w \mathbf{K}\frac{k_{r_w}(\theta_w)}{\mu_i} \quad (\nabla\psi_m - \rho_w g \boldsymbol{e_z})\right] + r_w = 0 \tag{2.6}$$

It is common to assume a constant water density and express the potential as energy per unit weight. The potential has then the dimensions of a length, which is the height of a hanging water column producing the suction necessary to extract water from the porous medium. It is often called the *hydraulic head h*. We then obtain the most common form of Richards' equation in the form of a volume conservation equation

$$\frac{\partial\theta_w}{\partial t} - \nabla\cdot\left[\mathbf{K_s} k_{r_w}(\theta_w) \quad (\nabla h - \boldsymbol{e_z})\right] + q_w = 0 \tag{2.7}$$

where $\mathbf{K_s} = \mathbf{K}\frac{\rho_w g}{\mu_w}$ is the saturated hydraulic conductivity tensor with dimension $[L/T]$ and q_w is a volumetric sink term.

Having one equation for the two unknowns θ_w and ψ_m we are again one equation short. The closure is again done by relating the matric potential ψ_m and the volumetric water content by a material function

$$\theta_w = \Phi S_w(\psi_m) = \Phi S_w\left(-h\rho_w g\right). \tag{2.8}$$

The function $S_w(\psi_m)$ is called soil water characteristic and is essentially the inverse of the capillary pressure-saturation curve.

2.2. Alternative Equations

Alternative equations have been proposed mainly to describe two effects: *hysteresis* and the occurrence of so-called *dynamic effects*.

2.2.1. Dynamic Effects

The term *dynamic effects* usually means the rate dependence of hydraulic characteristics in multi-phase flow processes. In a more general approach also effects occurring in transient flow situations are included which cannot be represented with Richards' equation or classic multi-phase flow equations with a time and rate-independent set of hydraulic parameter functions. We want to denote situations where dynamic effects occur with the term *hydraulic non-equilibrium*.

Dynamic effects have been discussed as early as 1960 (Biggar and Taylor 1960) and have been extensively studied especially in the last fifteen years. Dynamic effects occurred in drainage experiments (Topp et al. 1967; Vachaud et al. 1972), imbibition experiments (Biggar and Taylor 1960; Nielsen et al. 1962; Rawlins and Gardner 1963), different kinds of multi-step outflow experiments with different non-wetting phases (Schultze et al. 1999;

Wildenschild et al. 2001; O'Carroll et al. 2005; Vogel et al. 2008; Camps-Roach et al. 2010), in spontaneous imbibition experiments (Le Guen and Kovscek 2006; Schembre and Kovscek 2006; Hassanizadeh et al. 2002 and in a so-called multi-step flux experiment (Weller et al. 2011a). Most experiments were conducted with water as the wetting and air as the non-wetting phase, but other fluids have been used as well. This list of course does not attempt to be complete. In the experiments it was found that there was often a flow-rate-dependent behaviour which could not be explained with the classical equations. Often hydraulic parameter functions obtained from combinations of local measurements had a distinctly different shape than the relationships measured at equilibrium. Most experiments addressed the drainage of water from soil samples and found that the matrix potential was more negative at the same saturation than expected from the equilibrium curves. In imbibition experiments the opposite was found.

In general three different effects have been described

- all types of experiments showed that a curve constructed from combinations of locally measured potential and wetting phase saturation during a dynamic experiment is different from a curve measured under hydraulic equilibrium. The curves are non-unique and depend on the speed of change.
- in multi-step and one-step outflow experiments a rapid outflow at the beginning is followed by a slower secondary outflow even at high water contents. During the secondary outflow there is no measurable change of soil water potential.
- in multi-step flux experiments the water flux at the boundary is changed in a step-wise fashion. A change of the flux rate is followed by a rapid adjustment of wetting phase saturation and potential. After a while the potential change is reversed while the saturation change continues slowly. Potential overshoots have been observed both in imbibition and drainage in different materials from sand to loam.

In all experiments the dynamic effects were most pronounced in well sorted coarse material.

There are different kinds of possible explanations for dynamic effects and it is important to separate them carefully. We want to distinguish between macroscopic effects, which could in principle be resolved with Richards' equation or multi-phase flow equations by a more elaborate analysis of the measurements or by a simulation with higher spatial resolution, and microscopic effects which occur at the transition from pore scale to continuum scale and would require a modification of the transport equations.

2.2.2. Macroscopic Explanations for Dynamic Effects

Possible explanations on the macroscale are:

Averaging If two quantities are related by a non-linear relationship and there is some time-dependent distribution of both quantities which locally obeys this relationship it can easily be shown mathematically that the relationship obtained by averaging each quantity separately over space at each time and making pairs of the averaged quantities at the same time, will in general not be the original local relationship.

Measurement characteristic (temporal and spatial) Each measurement device has always a measurement characteristic determining how it is integrating the signal to be measured over space and time. Instantaneous point measurements are only possible in theory. The measurement characteristic can have a marked influence on the presence of sharp gradients and rapid changes.

Macroscopic Heterogeneity Averaging effects already occur in macroscopically homogeneous porous material. In macroscopically heterogeneous porous media they can be much more pronounced. Different materials have a different time response to external forcing. If e.g. infiltration in a heterogeneous porous medium is to be described with a one-dimensional simulation using effective homogeneous parameters, it is necessary that the infiltration process is slow enough to allow for horizontal equilibration of the different materials. Else preferential flow processes will result in dynamic effects with an extent depending on the flow rate.

2.2.3. Microscopic Explanations for Dynamic Effects

Apart from this macroscopic effects there are explanations which refer to pore scale processes. A nice list of possible effects is given in Wildenschild et al. 2001.

Phase Entrapment In rapid flow processes parts of a phase can get discontinuous. They block the flow of the other phase and can only drain very slowly by dissolution/ evaporation or surface film flow. Wildenschild et al. 2001 distinguished four types of phase entrapment.

- *Water Phase Entrapment* refers to the detachment of water clusters from the percolating water phase at high flow velocities, which could drain during slower drainage processes.
- *Pore Water Blockage* is caused by a rapid drainage at the bottom of the soil sample above the ceramic plate, blocking further outflow of the water phase. Viscous fingering could enhance this process in MSO experiments driven by a air pressure increase at the upper boundary.
- *Air entrapment*: As suggested by Schultze et al. 1999 the limited availability of air due to a not fully continuous air phase can lead to reduced drainage. If entrapped air is present, a rapid outflow compensated by a pressure decrease in the discontinuous air phase can be followed by a retarded outflow when the air pressure in the gas bubbles is equilibrated. Thus this effect should be more pronounced for compressible non-wetting phases and for samples containing entrapped air at the start of a drainage experiment.
- *Air-entry Value Effect* occurs in well sorted porous media with a well-defined air-entry value. If the air entry-value is exceeded in a completely water saturated porous medium in a large step, the outflow can occur in a more piston

like fashion due to the limited air availability than predicted by Richards' equation. However, this could in principle be simulated using multi-phase flow equations.

Dynamic Contact Angle At a moving contact line the contact angle between fluid and pore wall may be different under dynamic compared to steady-state conditions, and probably it may even depend on the phase velocity. O'Carroll et al. 2010 found reduced dynamic effects if the soil was treated to have a contact angle of about 80° compared to a fully water wet sample and explained this by a reduced effect of the dynamic contact angle.

Local Heterogeneity Dynamic effects could also result from soil heterogeneity so that the sensors are in contact with different materials with a different time-dependent behaviour. However, as the most pronounced dynamic effects were observed in homogeneous sand packings this is not a very likely explanation.

Phase Redistribution Weller et al. 2011b proposed in the analysis of multi-step flux experiments an internal redistribution of the phases as a reason for dynamic effects. After a change of the boundary condition the phases would first assume a configuration which can be reached quickly and afterwards redistribute to reach a more stable equilibrium phase configuration.

2.2.4. Hysteresis

Hysteresis is defined as the dependence of the hydraulic characteristics on the flow history and especially on the direction of flow. In particular this means that the relationship between volumetric water content and matric potential is non-unique. It was reported as early as 1927 (Haines 1927; Haines 1930) and also discussed in Richards 1931. The hysteresis of the relative permeability curve depending on the volumetric water content is generally assumed to be negligible (e.g. Topp 1971). Hysteresis is generally explained by phase entrapment and a different filling of individual pores caused by the pore geometry (so called "ink bottle effect") or by a different contact angle in drainage and imbibition.

The most common (and generally accepted) approach to hysteresis is to assert its existence and importance and then state to ignore it for the rest of the work. If hysteresis is taken into account it is mostly done by special forms of the soil water characteristic, which do not only depend on the volumetric water content, but also on history (see section 2.3.3). As entrapped air is dissolved and removed from the system over time, hysteresis is also a non-equilibrium process (though on potentially very long time scales).

2.2.5. Models including Hydraulic Non-Equilibrium

Different mathematical models have been proposed to extend or replace the multi-phase flow equations and deliver a better description of dynamic effects and hysteresis. The models here have been rewritten in a consistent way to allow for a better comparison of the approaches. We distinguish between models, which are based on microscopic explanations for dynamic effects (see section 2.2.3) and models, which attribute the effects mainly to heterogeneity (see section 2.2.2).

Models based on microscale considerations are based on effects which also occur in homogeneous porous media. The most common models of this type are the following four:

2.2.5.1. Dynamic-capillary pressure

A time-dependent relation between actual capillary pressure $p_n - p_w$ and the capillary pressure in equilibrium was first proposed by Stauffer 1978 and afterwards again based on thermodynamic arguments by Hassanizadeh and Gray 1990. The basic idea is that the actual capillary pressure is only equal to the static capillary pressure when the system is in equilibrium. Else there is a difference which depends on the time derivative of saturation and a relaxation time τ:

$$p_n - p_w = p_c^{eq} - \tau \frac{\partial S_w}{\partial t} \tag{2.9}$$

Dynamic capillary pressure has been widely discussed and there exists also some theoretical work (e.g. Schweizer 2012; Van Duijn et al. 2013).

2.2.5.2. Barenblatt

Barenblatt et al. 2003 proposed a slightly different model. They assume that capillary pressure and relative permeabilities do not depend directly on the current saturation, but on an effective saturation η, which depends on the rate of saturation change:

$$k_{r_w} = k_{r_w}(\eta), \;\; k_{r_n} = k_{r_n}(\eta), \;\; p_n - p_w = p_c^{eq}(\eta) \tag{2.10}$$

$$\eta = S_w + \tau \frac{\partial S_w}{\partial t} \tag{2.11}$$

An advantage of the Barenblatt model compared to dynamic-capillary pressure is, that it also produces dynamic effects in relative permeability for which experimental evidence exists in multi-step flux experiments (Weller et al. 2011a).

2.2.5.3. Hassanizadeh and Gray (interfacial area)

Hassanizadeh and Gray 1993 proposed a model which is thermodynamically motivated and tries to incorporate the energy stored in interfaces into the flow equations introducing the specific area (the area per unit volume) a_{nw} of the interface between the wetting and non-wetting phase as new state variable. In the form presented by Joekar-Niasar and

Hassanizadeh 2011 they get the new flux equations:

$$\boldsymbol{j}_i = -\rho_i \mathbf{K} \frac{k_{r_i}(S_i)}{\mu_i} \left[\nabla p_i - \rho_i g \boldsymbol{e}_z - \varPsi_{ia} \nabla a_{nw} - \varPsi_{iS} \nabla S_i\right] \tag{2.12}$$

$$\frac{\partial\, a_{nw}}{\partial\, t} + \nabla \cdot (a_{nw} \boldsymbol{w}_{nw}) = E_{nw} \tag{2.13}$$

$$\boldsymbol{w}_{nw} = -\mathbf{K}_{nw} \cdot \left[\nabla \left(a_{nw} \sigma_{nw}\right) + \varPsi_{nw} \nabla S_w\right] \tag{2.14}$$

$$\tag{2.15}$$

where $\varPsi_{ia}$ and $\varPsi_{iS}$ are material properties, $\boldsymbol{w}_{nw}$ is the average velocity of the fluid-fluid interfaces, E_{nw} is the net rate of production of interfaces between wetting and non-wetting phase, σ_{nw} is the macroscale interfacial tension and $\varPsi_{nw}$ is another material property.

They additionally assume a dynamic relation between the phase pressure difference and the static capillary pressure:

$$p_n - p_w = p_c^{eq}(S_w, a_{nw}) - \tau \frac{\partial\, S_w}{\partial\, t} \tag{2.16}$$

The capillary pressure at equilibrium is now assumed to be a function of interfacial area as well.

2.2.5.4. Hilfer

Hilfer et al. (Hilfer 2006a; Doster et al. 2010) propose a different approach to develop improved equations for multi-phase flow. They start from a set of mass and momentum balances for each phase:

$$\sum_i S_i = 1 \tag{2.17}$$

$$\frac{\partial\left(\varPhi S_i \rho_i\right)}{\partial\, t} + \nabla \cdot \boldsymbol{j}_i = M_i \tag{2.18}$$

$$\boldsymbol{j}_i = \nabla \cdot \left(S_i \varPhi \rho_i \boldsymbol{v}_i\right) \tag{2.19}$$

$$S_i \varPhi \rho_i \frac{D^i}{Dt} \boldsymbol{v}_i - S_i \varPhi \nabla \cdot \boldsymbol{\varSigma}_i - S_i \varPhi \boldsymbol{F}_i = \boldsymbol{m}_i - \boldsymbol{v}_i M_i \tag{2.20}$$

where M_i is mass exchange with other phases, $D^i/Dt = \partial/\partial t + \boldsymbol{v}_i \cdot \nabla$ is the material derivative for phase i, $\boldsymbol{v}_i$ is the velocity of phase i, $\boldsymbol{\Sigma}_i$ is the stress tensor in phase i, $\boldsymbol{F}_i$ is the body force per unit volume acting on phase i and $\boldsymbol{m}_i$ is the momentum transfer to all other phases.

The traditional multi-phase flow equations for two phases ($i = \{n, w\}$) can be obtained by ignoring inertial terms, using the fluid phase pressure gradient for the divergence of the stress $\nabla \cdot \boldsymbol{\Sigma}_i = -\nabla p_i$, restricting the body forces $\boldsymbol{F}_i$ to gravity and using a special choice for the viscous drags $\boldsymbol{m}_i$ introducing relative permeability. Finally a capillary pressure saturation equation is needed as closure.

Hilfer now proposes to improve this equations by dividing the wetting and non-wetting phase in a percolating and a non-percolating phase. Four balance equations are then obtained, one for each of the four phases ($i = \{n_p, n_u, w_p, w_u\}$, with e.g. n_u being the non-percolating non-wetting phase).

The exchange between the percolating and non-percolating phase of the same fluid is assumed to be a function f_i of the two respective saturations and of the time derivative of the total saturation of the fluid:

$$M_{w_p} = -M_{w_u} = \varPhi \rho_w f_w(S_{w_u}, S_{w_p}, \partial_t S_w) \tag{2.21}$$

$$M_{n_p} = -M_{n_u} = \varPhi \rho_n f_n(S_{n_u}, S_{n_p}, \partial_t S_n) \tag{2.22}$$

The (scalar) stress for the percolating phases is the same as in the standard model. However, the stress in the non-percolating phases is quite different. It cannot depend on the pressure in the percolating phase of the same fluid, as there is no contact between both. It therefore is assumed to depend on the pressure in the percolating phase of the other fluid plus an energy density u_{i_p,j_u} stored in the interface between the two phases, which he assumes to depend on the non-percolating phase saturation:

$$\varSigma_{w_p} = -p_{w_p}, \quad \varSigma_{n_p} = -p_{n_p} \tag{2.23}$$

$$\varSigma_{w_u} = -p_{n_p} + u_{n_p w_u}(S_{w_u}), \quad \varSigma_{n_u} = -p_{w_p} + u_{w_p n_u}(S_{n_u}) \tag{2.24}$$

The body force acting on the percolating phases is also produced by gravity, but additional capillary body forces $\boldsymbol{F}_{c_i}$ are acting on the non-percolating fluids keeping them trapped inside the medium.

$$\boldsymbol{F}_{w_p} = \rho_w g \boldsymbol{e}_z, \quad \boldsymbol{F}_{n_p} = \rho_n g \boldsymbol{e}_z \tag{2.25}$$

$$\boldsymbol{F}_{w_u} = \rho_w g \boldsymbol{e}_z + \boldsymbol{F}_{c_w}(S_{w_p}), \quad \boldsymbol{F}_{n_p} = \rho_n g \boldsymbol{e}_z + \boldsymbol{F}_{c_n}(S_{n_p}) \tag{2.26}$$

Finally the viscous drag is assumed to be a linear function of the velocity difference between the phases and the matrix ($i = s$) with coefficient R_{ij} which are material properties. Of course as there is no contact between the percolating and non-percolating phase of the same fluid there is also no viscous drag term for these pairs:

$$\boldsymbol{m}_{w_p} = R_{w_p n_p}(\boldsymbol{v}_{n_p} - \boldsymbol{v}_{w_p}) + R_{w_p n_u}(\boldsymbol{v}_{n_u} - \boldsymbol{v}_{w_p}) - R_{w_p s}\boldsymbol{v}_{w_p} \tag{2.27}$$

$$\boldsymbol{m}_{w_u} = R_{w_u n_p}(\boldsymbol{v}_{n_p} - \boldsymbol{v}_{w_u}) + R_{w_u n_u}(\boldsymbol{v}_{n_u} - \boldsymbol{v}_{w_u}) - R_{w_u s}\boldsymbol{v}_{w_u} \tag{2.28}$$

$$\boldsymbol{m}_{n_p} = R_{n_p w_p}(\boldsymbol{v}_{w_p} - \boldsymbol{v}_{n_p}) + R_{n_p w_u}(\boldsymbol{v}_{w_u} - \boldsymbol{v}_{n_p}) - R_{n_p s}\boldsymbol{v}_{n_p} \tag{2.29}$$

$$\boldsymbol{m}_{n_u} = R_{n_u w_p}(\boldsymbol{v}_{w_p} - \boldsymbol{v}_{n_u}) + R_{n_u w_u}(\boldsymbol{v}_{w_u} - \boldsymbol{v}_{n_u}) - R_{n_u s}\boldsymbol{v}_{n_u} \tag{2.30}$$

Hilfer proposes concrete functions for f_w, f_n, u_{ij}, F_{c_w} and F_{c_n} but their specific forms still need to be investigated. As the assumptions listed above provide only nine equations for ten unknowns, an additional closure condition is needed. Here again different choices are possible.

2.2.5.5. Ross and Smettem

There are two models also addressing to some degree non-equilibrium effects, but using explanations based on macroscopic heterogeneity. One of them was proposed by Ross and Smettem 2000 for an effective one-dimensional description of heterogeneous porous media at the macroscale. Their model is similar to the model proposed by Barenblatt. The basic idea is a partial decoupling of potential and saturation. The change in saturation is lagging behind the equilibrium value calculated from the current matric potential ψ_m and only relaxing to equilibrium values over time:

$$\frac{\partial \theta_w}{\partial t} = \frac{1}{\tau}(\theta_{w_{\mathsf{eq}}}(\psi_m) - \theta_w) \tag{2.31}$$

A peculiarity of the approach of Ross and Smettem is that they obtain a discretised version of this equation by a separation of variables and a subsequent integration. The discrete form is

$$\theta_w^{k+1} = \theta_w^k + \left(\theta_{w_{\mathsf{eq}}}(\psi_m^{k+1}) - \theta^k\right)\left(1 - e^{-\mathit{\Delta} t/\tau}\right) \tag{2.32}$$

with the water contents θ_w^k and θ_w^{k+1} in time step k and $k+1$ and a time step size of $\mathit{\Delta} t$. They argue, that the use of this formula would be more exact than an Implicit Euler scheme. However, to perform the separation of variables they completely ignore the time dependence of the potential and thus of $\theta_{w_{\mathsf{eq}}}$. Thus this statement is rather questionable.

2.2.5.6. Multiple-Domain

The non-equilibrium between materials with different hydraulic properties results in preferential flow processes. Different preferential flow models have been developed as an extension of Richards' equation. One of the most prominent is the dual-permeability model of Gerke and Genuchten 1993, which assumes that macropore flow can be described by two coupled Richards' equations for the macropores and the aggregates, which are coupled by a first order exchange term w_{ij} depending on the potential difference between the two flow domains i and j and an empirical parameter α_{ij} which at least depends on the materials and the flow rate. In general a multi-domain model can be written in the following form:

$$\frac{\partial\,(\rho_w \mathit{\Phi} S_i)}{\partial\, t} + \nabla \cdot \boldsymbol{j}_i + r_i + w_{ij} = 0, \quad i,j \in \{1,\dots,N\} \tag{2.33}$$

$$\boldsymbol{j}_i = -\rho_w \mathbf{K}_i \frac{k_{r_i}(S_i)}{\mu_w}\;(\nabla\psi_i - \rho_w g \boldsymbol{e}_z) \tag{2.34}$$

$$S_i = f(\psi_{c_i}), \quad S_i \in [0, S_{i_{\mathsf{max}}}], \quad \sum_{i=1}^{N} S_i = S_w \tag{2.35}$$

$$w_{ij} = \alpha_{ij}\,(\psi_i - \psi_j)\,, \quad i,j \in 1,\dots,N \tag{2.36}$$

2.3. Hydraulic Functions

In the classical two-phase flow equations, in Richards' equation and in most of the alternative models described in section 2.2 the properties of the material/fluid interactions are represented by three parameter functions: The relative permeability functions $k_{r_i}(S_i)$ and the soil water characteristic $S_w(\psi_m)$ or its inverse, the capillary pressure–saturation curve $p_c(S_w)$. Together they are often called the hydraulic functions of a porous medium.

The functions have to be experimentally determined for each material though there are some physical constraints: The relative permeabilities should be equal to one for full saturation of the respective phase and equal to zero if the phase is absent. The wetting phase saturation should be one if the hydraulic head is lower than the so called air entry value, the potential at which the largest pores present in the system are completely filled with water, and approach zero for very large hydraulic heads. The relative permeability functions and the soil water characteristic should be monotonically increasing and the capillary pressure–saturation curve should be monotonically decreasing.

For the simulation of multi-phase flow in porous media, the soil water characteristic and the relative permeability functions for the two phases are needed in the form of steady functions over the whole range of potentially occurring potentials. Especially the relative permeability functions are quantities which are hard to measure. Therefore it is common to use parametrised functions which reduce the measurement problem to the determination of suitable parameters for the sample. This is often done by solving parameter estimation problems. It should be noted that model errors gravely influence the parameter estimation process and may not always result in a bad agreement between experimental data and simulation results, but rather in unreliable parameter estimates. Thus the selection of appropriate parametrisations of the hydraulic functions is of utmost importance for a meaningful model output.

2.3.1. Parametrisations for the Soil Water Characteristic

As the sum of the two phase saturations in two-phase flow is 1, one function relating saturation and matric potential is sufficient. By convention the wetting phase saturation is used. As the capillary pressure–saturation curve is just the inverse of the soil water characteristic only the latter will be discussed in this work. Matric potential will be used as independent variable, as the term capillary pressure narrows the perspective on capillary forces. Water at low matric potentials is not bound by capillary forces, but by surface interactions between water and the solid phase.

There exists an incredibly large (and still growing) number of parametrisations for the soil water characteristic. In this section some of the most prominent will be discussed. However, there is still some interesting development. Tuller et al. 1999 for example proposed a model based on the assumption of angular pores which should represent better water sitting in corners and films.

To include residual saturations of trapped discontinuous wetting or non-wetting phase

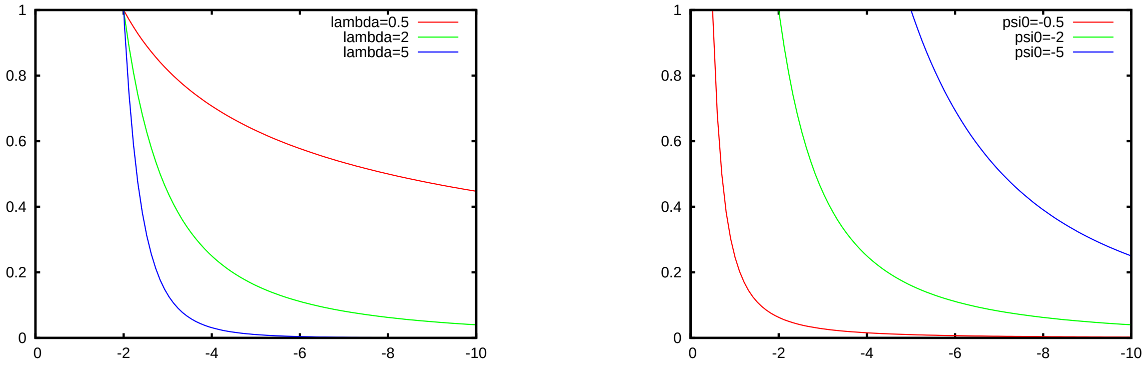

Figure 2.1.: Dependency of the Brooks-Corey model for the soil water characteristic on the parameter λ (left) and the air entry value ψ_{m_0} (right).

it is common to formulate a parametrisation for the so called effective saturation

$$S_{\mathsf{eff}}(\psi_m) = \frac{\theta_w - \theta_{w_r}}{\theta_{w_s} - \theta_{w_r}} = \frac{S_w - S_{w_r}}{1 - S_{n_r} - S_{w_r}},$$

where θ_{w_r} and S_{w_r} are the residual wetting phase content and the residual wetting phase saturation, $\theta_{w_s} = \Phi(1 - S_{n_r})$ is the maximal wetting phase content at saturation and S_{n_r} is the residual non-wetting phase saturation.

2.3.1.1. Brooks-Corey Model

Brooks and Corey 1966 proposed the model

$$S_{\mathsf{eff}}(\psi_m) = \begin{cases} \left(\frac{\psi_m}{\psi_{m_0}}\right)^{-\lambda} & \text{if } \psi_m < \psi_{m_0} \\ 1 & \text{if } \psi_m \geq \psi_{m_0} \end{cases}$$

The model has two parameters:

ψ_{m_0} is called air entry value. It specifies the potential at which the largest pores start to drain. Above this point the soil is completely saturated.

λ specifies the steepness of the soil water retention curve. A very high λ corresponds to pores which all have the same size and thus drain at the same potential.

The disadvantage of the Brooks-Corey model is that its derivative is discontinuous at the air entry point.

2.3.1.2. Van Genuchten Model

The unsteadiness of the derivative of the Brooks-Corey function at the air entry value may cause problems in numerical codes. van Genuchten 1980 proposed an alternative parametrisation, which is differentiable everywhere:

$$S_{\mathsf{eff}}(\psi_m) = [1 + (\alpha|\psi_m|)^n]^{-m}$$

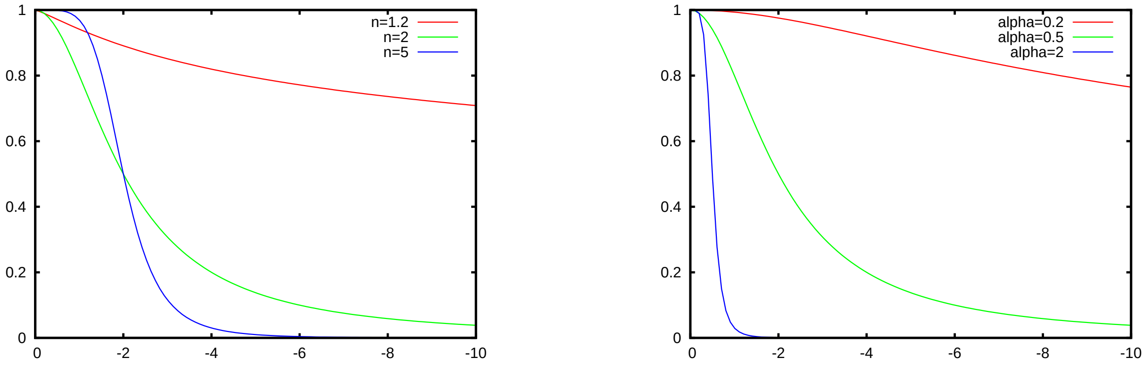

Figure 2.2.: Dependency of the van Genuchten model for the soil water characteristic on the parameters n (left) and α (right).

The model has three parameters:

n is related to the steepness of the function (like the λ in the Brooks-Corey model).

α The inverse of α is the point of inflection of the soil water characteristic. Thus for high n (steep functions) α is related to the position of the air entry value (and is often wrongly called so).

m Due to a restriction coming from the application of the Mualem model (see below) m is not a free parameter but is fixed to $m = i - \frac{1}{n}$ and usually $m = 1 - \frac{1}{n}$ is taken.

2.3.1.3. Multiple van Genuchten model

Many natural soils are a mixture of materials with different grain size, mineralogical properties etc. and also contain large pores (macropores) originating from animals or roots. The soil water characteristic of such soils is more complicated and its derivative has more than one maximum. To get more flexibility in the parametrisation, so-called multiple van Genuchten models were proposed in the early nineties (e.g. Durner 1994). A multiple van Genuchten model is constructed as a weighed sum of ordinary van Genuchten models with different values α_i, n_i and m_i for each sub-material. The weighting factors f_i add to one.

$$S_{\text{eff}}(\psi_m) = \sum_i f_i \left[1 + (\alpha_i |\psi_m|)^{n_i}\right]^{-m_i} \tag{2.37}$$

In contrast to multi-domain models (section 2.2.5.6) an essential assumption of multiple van Genuchten models is that the materials are so intimately mixed that they are always in equilibrium at the time scale of interest.

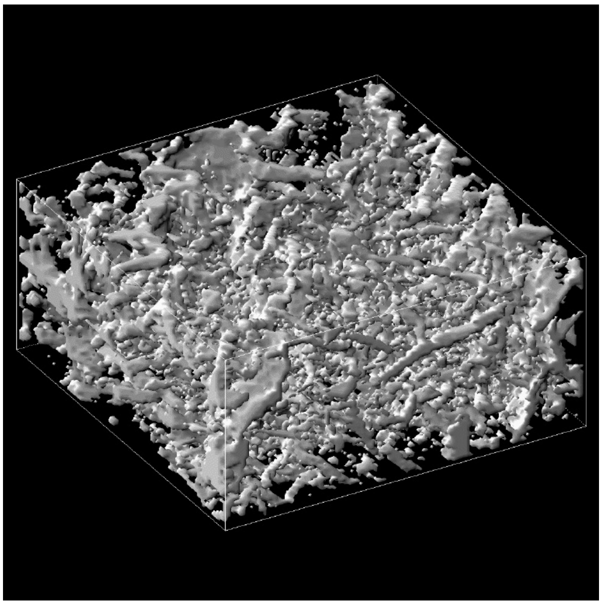

Figure 2.3.: Pore structure of a natural porous medium reconstructed from serial sections.

2.3.2. Parametrisations for the Hydraulic Conductivity Function

As wetting and non-wetting phase conductivities in a porous medium are notoriously difficult to measure, methods have been proposed for the derivation of the relative permeability functions from the soil water characteristic. The basic idea of these approaches is the derivation of an equivalent pore size distribution from the soil water characteristic and the calculation of relative permeabilities through integrals over pore conductivities times weighted with their relative frequency. Originally the models have been motivated by the idea of representing a porous medium as a bundle of cylindrical capillaries. This is obviously not true. The real pore structure of porous media is much more complicated (figure 2.3).

Especially at low water content and when the water phase loses its connectivity, water flow is dominated by liquid films and pendular rings. Tuller and Or 2001 thus derived relative permeability from the assumption of angular pores. However, the traditional models work surprisingly well. This might be due to the fact that larger pores dominate the overall flow behaviour and that due to capillarity effects there is really a relation between a cylindrical approximation of the conductivity at the narrowest portion of a flow path and the potential at which it desaturates.

The same principles can be applied to calculate wetting and non-wetting phase relative permeabilities. The only difference is the range over which the integration has to be conducted. For the non-wetting phase one has to integrate from full wetting phase saturation to the saturation of interest (as the largest pores are always filled with the non-wetting phase), for the wetting phase from zero wetting phase saturation to the saturation of interest (as the smallest pores are occupied by the wetting phase). The residual saturations are assumed to not contribute to the conductivity, thus the models are again formulated as functions of $S_{\textsf{eff}}$.

As this work is mainly concerned with the simulation of unsaturated water flow with Richards' equation, only the wetting phase relative permeabilities will be given.

2.3.2.1. Burdine Model

If we assume that the conductivity of a pore can be calculated from its radius with the law of Hagen-Poiseuille it scales with the square of its radius. If the pore size distribution is known, the relative permeability can be calculated from

$$k_{r_w}(S_{\textsf{eff}}) = S_{\textsf{eff}}^{\tau} \frac{\int_0^{S_{\textsf{eff}}} \frac{1}{\psi_m(S)^2} dS}{\int_0^1 \frac{1}{\psi_m(S)^2} dS}$$

This model was first proposed by Burdine 1953. The additional term $S_{\textsf{eff}}^{\tau}$ is intended to take the tortuosity of the pores into account (i.e. that the flow path gets more tortuous if their water content drops). τ is a dimensionless fitting parameter.

The Burdine model is often used together with the Brooks-Corey model. Using the Young-Laplace-Equation (Bear 1972) to relate matric potential and effective pore radius r:

$$\psi_m = -\frac{2\sigma_w \cos(\Theta)}{r}, \tag{2.38}$$

where σ_w is the surface tension of the wetting fluid and Θ is the contact angle, the integrals can be directly evaluated and one obtains:

$$k_{r_w}(S_{\textsf{eff}}) = S_{\textsf{eff}}^{\tau+1+2/\lambda}$$

where usually $\tau = 2$ is used.

2.3.2.2. Mualem Model

Mualem 1976 proposed a slightly different model assuming that the length of a pore is proportional to its radius and that the pores are randomly connected. He argued (not completely convincingly) that the square could then be taken out of the integral to obtain

$$k_{r_w}(S_{\textsf{eff}}) = S_{\textsf{eff}}^{\tau} \left[\frac{\int_0^{S_{\textsf{eff}}} \frac{1}{h(S)} dS}{\int_0^1 \frac{1}{h(S)} dS} \right]^2$$

For the Brooks-Corey model Mualem obtained

$$k_{r_w}(S_{\textsf{eff}}) = S_{\textsf{eff}}^{\tau+2+2/\lambda}$$

Mualem argued that there is a good agreement between experimental values and model results for many soils with $\tau = 0.5$. This value is used in many studies (probably also as

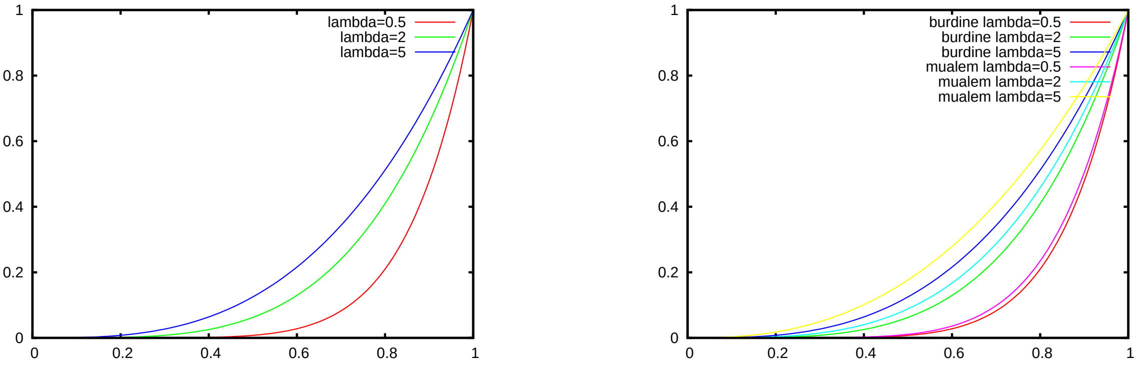

Figure 2.4.: Dependency of the relative permeability calculated by the Brooks-Corey/Burdine model on λ (left) and comparison of the Brooks-Corey/Burdine and the Brooks-Corey/Mualem model (right).

the influence of τ on the shape of the relative permeability function is rather small). The only difference between the Brooks-Corey/Mualem and the Brooks-Corey/Burdine model is the interpretation of τ. With the standard values of τ for both relative permeability models one obtains an exponent of $2.5 + 2/\lambda$ for the Brooks-Corey/Mualem model in contrast to $3 + 2/\lambda$ obtained by the Brooks-Corey/Burdine model.

2.3.2.3. van Genuchten/Mualem

Van van Genuchten 1980 was able to show that for his model the integral

$$\int_{0}^{S_{\text{eff}}} \frac{1}{|\psi_m(S)|} dS. \tag{2.39}$$

can be solved analytically assuming that $m = i - 1/n$ (where usually $i = 1$ is used) with the result:

$$\int_{0}^{S_{\text{eff}}} \frac{1}{|\psi_m(S)|} dS = 1 - (1 - S_{\text{eff}}^{1/m})^m \quad (m = 1 - 1/n). \tag{2.40}$$

The relative permeability is then given by

$$k_{r_w}(S_{\text{eff}}) = S_{\text{eff}}^{\tau} \cdot \left[1 - \left(1 - S_{\text{eff}}^{1/m}\right)^m\right]^2. \tag{2.41}$$

However, as has been shown by Ippisch et al. 2006, the van Genuchten/Mualem model can produce very steep relative permeability curves for $S_{\text{eff}} \to 1$, which are non-physical, if $n < 2$ or α too large. This is caused by the missing air entry value in the van Genuchten model. As the permeability is proportional to the square of the pore radius large pores will have a dominant influence on the relative permeability. As each porous medium

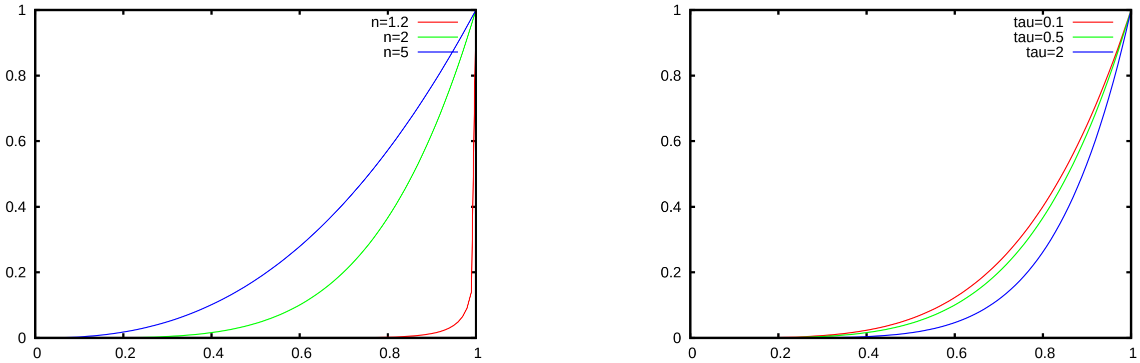

Figure 2.5.: Dependency of the van Genuchten/Mualem model for the relative wetting phase relative permeability on the parameter λ (left) and τ (right).

has a maximal effective pore radius, there is also a minimal hydraulic head (i.e. an air entry value), which can be calculated by the Young-Laplace-Equation (equation 2.38). In the van Genuchten model pores with an arbitrary large radius are only excluded if the derivative $\left|\frac{dS}{d\psi_m}\right|$ goes faster to zero than $1/\psi_m$. As

$$\left|\frac{dS}{d\psi_m}\right| = \alpha mn \left(\alpha|\psi_m|\right)^{n-1} \left[1 + \left(\alpha|\psi_m|\right)^n\right]^{-(m+1)}, \tag{2.42}$$

this is only the case if $n > 2$.

The maximum of $\left|\frac{dS}{d\psi_m}\right|$ is located at α^{-1}. This should at least be larger than the air entry value corresponding to the largest pores present in the soil. This results in the additional condition

$$\alpha^{-1} \gg |\psi_{m_0}| \Leftrightarrow \alpha|\psi_{m_0}| \ll 1 \tag{2.43}$$

2.3.2.4. Modified van Genuchten/Mualem Model

The straightforward introduction of an air-entry value ψ_{m_0} in the van Genuchten model is difficult, as there is no known solution for the resulting Mualem integral. However, Vogel et al. 2001 proposed a modification, which is a truncation and rescaling of the original van Genuchten model for which the solution to the integral is well known. This approach has been simplified by Ippisch et al. 2006 to

$$S_{\mathsf{eff}}(\psi_{m_0}) = \begin{cases} \frac{1}{S^*_{\mathsf{eff}}} \cdot [1+(\alpha|\psi_m|)^n]^{-m} & \psi_m < \psi_{m_0} \\ 1 & \psi_m \geq \psi_{m_0} \end{cases} \quad , \quad S^*_{\mathsf{eff}} = [1+(\alpha|\psi_{m_0}|)^n]^{-m}$$

$$k_{r_w}(S_{\mathsf{eff}}) = \begin{cases} S^{\tau}_{\mathsf{eff}} \cdot \left[\frac{1-\left(1-\left(S_{\mathsf{eff}} S^*_{\mathsf{eff}}\right)^{1/m}\right)^m}{1-\left(1-{S^*_{\mathsf{eff}}}^{1/m}\right)^m} \right]^2 & S_{\mathsf{eff}} < S^*_{\mathsf{eff}} \\ 1 & S_{\mathsf{eff}} \geq S^*_{\mathsf{eff}} \end{cases} \tag{2.44}$$

For $\psi_{m_0} = 0$ one obtains the standard van Genuchten/Mualem model and for $\alpha|\psi_{m_0}| \gg 1$ this model converges to the Brooks-Corey model as then

$$[1+(\alpha|\psi_m|)^n]^{-m} \approx (\alpha|\psi_m|)^{-mn} \quad \mathsf{for} \quad \psi_m < \psi_{m_0} \tag{2.45}$$

and

$$S_{\mathsf{eff}}(\psi_m) \approx \frac{(\alpha|\psi_m|)^{-mn}}{(\alpha|\psi_{m_0}|)^{-mn}} = \left(\frac{\psi_m}{\psi_{m_0}}\right)^{-mn} . \tag{2.46}$$

2.3.2.5. Multiple van Genuchten/Mualem model

A closed form solution for the relative permeability function for a multiple van Genuchten/Mualem model was published by Priesack and Durner 2006:

$$\kappa(\psi_m) = \frac{\sum\limits_i f_i \alpha_i \left\{1-(\alpha_i|\psi_m|)^{n_i-1}\left[1+(\alpha_i|\psi_m|)^{n_i}\right]^{-m_i}\right\}}{\sum\limits_i f_i \alpha_i} \tag{2.47}$$

$$k_{r_w}(\psi_m) = S_{\mathrm{eff}}(\psi_m)^{\tau} \kappa(\psi_m)^2 \tag{2.48}$$

2.3.2.6. Modified multiple van Genuchten/Mualem model

Of course the same technique used to produce a van Genuchten model with an air entry value can also be applied to the multiple van Genuchten model to obtain the modified multiple van Genuchten model:

$$S_{\mathsf{eff}}(\psi_m) = \begin{cases} \frac{S_{\mathrm{eff}}(\psi_m)}{S_{\mathrm{eff}}(\psi_{m_0})} & \psi_m < \psi_{m_0} \\ 1 & \psi_m \geq \psi_{m_0} \end{cases} \tag{2.49}$$

$$k_{r_w}(S_{\mathsf{eff}}) = \begin{cases} S_{\mathrm{eff}}(\psi_m)^{\tau} \left(\frac{\kappa(\psi_m)}{\kappa(\psi_{m_0})}\right)^2 & \psi_m < \psi_{m_0} \\ 1 & \psi_m \geq \psi_{m_0} \end{cases} \tag{2.50}$$

2.3.2.7. Splines

Splines are an attractive alternative for the specification of the soil water characteristic and the relative permeability curve as they require (nearly) no assumptions on the shape of the hydraulic function. Chardaire and Liu 1992 used linear splines to estimate the capillary pressure-saturation curve and the relative permeability curve. Kastanek and Nielsen 2001 used cubic splines for the description of the soil water characteristic and Bitterlich et al. 2004 estimated the coefficients of quadratic B-splines and piecewise cubic Hermite splines to determine the hydraulic functions. Piecewise cubic Hermite splines are defined by

$$s(x) = \sum_{i=0}^{n} \left(y_i \phi_i(x) + y_i' \xi_i(x) \right) \tag{2.51}$$

where ϕ_i and ξ_i are cubic base functions which have non-zero values only on the interval $]x_{i-1} : x_{i+1}[$ with $\phi_i(x_i) = \xi(x_i) = 1$ and y_i and y_i' are the value and the derivative of the function to be interpolated at point x_i.

As Fritsch and Carlson 1980 showed, a piecewise cubic Hermite spline is monotone if the data is monotone, i.e.

$$y_0 \leq y_1 \leq \cdots \leq y_n \text{ for } x_0 \leq x_1 \leq \cdots \leq x_n, \tag{2.52}$$

and if the derivatives are in a monotonicity region. This can be either achieved by ex post checking and modification of given derivatives or by approximation of the derivatives from the difference quotients of the adjoining intervals. Brodlie 1980 proposed to use

$$y_i' \approx \begin{cases} \frac{\Delta_{i-1}\Delta_i}{\alpha\Delta_{i-1}+(1-\alpha)\Delta_i} & \text{if } \Delta_{i-1}\Delta_i > 0 \\ 0 & \text{else} \end{cases} \tag{2.53}$$

$$\text{with } \Delta_i = \frac{y_{i+1} - y_i}{x_{i+1} - x_i}$$

$$\text{and } \alpha = {}^1\!/_3 \left(1 + \frac{x_{i+1} - x_i}{x_{i+1} x_{i-1}} \right).$$

Fritsch and Butland 1984 recommended

$$y_i' \approx \begin{cases} 2\frac{\Delta_{i-1}\Delta_i}{\Delta_{i-1}+\Delta_i} & \text{if } \Delta_{i-1}\Delta_i > 0 \\ 0 & \text{else} \end{cases}. \tag{2.54}$$

As the latter produces an interpolated function which is still close to a piecewise linear interpolation (Herzog 2005) but is differentiable, it seems preferable if no further knowledge about the function is available.

Monotone piecewise cubic Hermite splines can be used for both the soil water characteristic and the relative permeability function. Of course it would also be possible to derive a relative permeability function from the interpolated soil water characteristic with the Mualem or Burdine model.

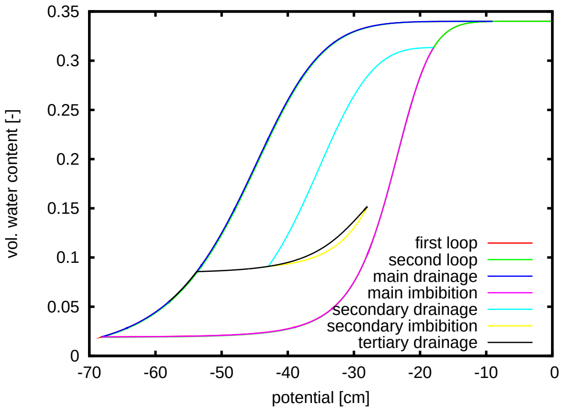

Figure 2.6.: Main hysteresis loops and scanning curves in a porous medium produced with the modified dependent domain model by Mualem.

2.3.3. Hysteretic Hydraulic Functions

As explained in section2.2.4 hysteresis is mostly a consequence of phase entrapment. An alternative to treating the entrapped non-wetting phase explicitly like in the model of Hilfer (section 2.2.5.4) is the usage of hysteresis models. Here functions for the water character are used, which do not only depend on the matric potential, but also on the wetting history of the system. If completely saturated porous media [2] are drained completely, the primary drainage curve is obtained. The subsequent re-wetting to the maximal achievable saturation yields the main imbibition curve, the following drainage the main drainage curve. Curves obtained by reverting the direction of flow at points in between are called scanning loops (figure 2.6). Typically hysteresis models use the main drainage and main imbibition curve to derive intermediate curves by some scaling laws.

Mualem derived a variety of physically motivated hysteresis models (Mualem 1974; Mualem and Dagan 1975; Mualem 1977; Mualem and Morel-Seytoux 1978; Mualem 1984) allowing for a different degree of flexibility and requiring different amounts of information (only the main imbibition and drainage curves or additionally some scanning loops). Kool et al. 1987 proposed a simpler model, which is rather popular. However, there is an important difference. If the direction of saturation change is reversed at some potential and then reversed again, the saturation at the initial potential should be the same due to thermodynamic reasons. While Mualem's models produce such closed hysteresis loops, the Kool and Parker model doesn't. As open hysteresis loops result in energy production or dissipation this is hard to accept. The Kool and Parker model is also restricted to a van Genuchten parametrisation, whereas the Mualem models are in principle much more flexible. Parker and Lenhard proposed a hysteresis model for three-phase flow (Parker and Lenhard 1987) and even developed a theory for the change of the conductivity due to phase entrapment (Lenhard and Parker 1987).

In this study the modified dependent domain model of Mualem was used.

[2] this can only achieved after a very long time or with special techniques like saturation under vacuum or a gas phase of CO_2

2.3.3.1. Modified Dependent Domain Model

For a shorter notation of the equations we define the maximal wetting phase saturation $S_{\mathsf{max}} = 1 - S_{n_r}$ and the extractable part of the wetting phase $S_{\mathsf{mob}} = 1 - S_{n_r} - S_{w_r}$.

The modified dependent domain model calculates the saturation from the main drainage curve $S_{\mathsf{drain}}(\psi_m)$ and the main imbibition curve $S_{\mathsf{imb}}(\psi_m)$. Additionally the saturation S_{rev} and the matric potential $\psi_{m_{\mathsf{rev}}}$ where the direction of saturation change has reversed last is needed. The model can be written in the following simplified form (compared to Mualem's original version).

If the current potential is more negative than $\psi_{m_{\mathsf{rev}}}$ (drainage) the saturation is calculated from

$$S(\psi_m) = \max\left(S_{\mathsf{drain}}(\psi_m), S_{\mathsf{rev}} - \frac{P_d(S(\psi_m))}{S_{\mathsf{mob}}}\left[S_{\mathsf{max}} - S_{\mathsf{imb}}(\psi_m)\right]\left[S_{\mathsf{rev}} - S_{\mathsf{imb}}(\psi_m)\right]\right) \tag{2.55}$$

If the current potential is less negative than $\psi_{m_{\mathsf{rev}}}$ (imbibition) the saturation is calculated from

$$S(\psi_m) = \min\left(S_{\mathsf{imb}}(\psi_m), S_{\mathsf{rev}} + \frac{P_d(S_{\mathsf{rev}})}{S_{\mathsf{mob}}}\left[S_{\mathsf{max}} - S_{\mathsf{imb}}(\psi_{m_{\mathsf{rev}}})\right]\left[S_{\mathsf{imb}}(\psi_m) - S_{\mathsf{imb}}(\psi_{m_{\mathsf{rev}}})\right]\right) \tag{2.56}$$

The so-called pore block factor $P_d(S)$ accounts for the fact that during imbibition a part of the pore space might be filled with a disconnected non-wetting phase.

$$P_d(S) = S_{\mathsf{mob}} \cdot \frac{S_{\mathsf{max}} - S}{S_{\mathsf{max}} - S_{\mathsf{imb}}(S^{-1}_{\mathsf{drain}}(S))} \tag{2.57}$$

Equation 2.55 is an implicit definition of the saturation (with the saturation occurring also on the right side of the equation) and thus the root of a non-linear equation has to be computed to obtain the solution.

In principle any drainage and imbibition curves with arbitrary shape can be used as long as it is guaranteed that $S_{\mathsf{imb}}(\psi_m) \leq S_{\mathsf{drain}}(\psi_m)$ at any potential ψ_m. For the Brooks-Corey model this is always fulfilled if the same λ is used for both curves and $\psi_{m_{0\mathsf{imb}}} \geq \psi_{m_{0\mathsf{drain}}}$, for the van Genuchten model if n and m are the same for imbibition and drying and $\alpha_{\mathsf{imb}} \leq \alpha_{\mathsf{drain}}$. Different values for λ or n and m should be possible if the hysteresis model is only applied in a range where the above condition holds.

2.3.4. Miller Similarity Scaling

The effect of fine scale heterogeneity can be represented using the assumption of Miller similarity. [3] The concept of Miller similarity (Miller and Miller 1956) implies that at

[3] This paragraph is a nearly literal quote from Ippisch 2003. For the sake of readability the quotation marks are omitted.

different locations the geometry of the pore space is exactly the same and only the size of the elements which compose the pore space is different. If we can associate a scaling parameter with a typical length scale of the pore space (e.g. average or maximal pore diameter), the hydraulic parameters measured at a certain scale χ^* can be used at a different scale χ, if capillary pressure and permeability are scaled using the factor $\eta = \chi/\chi^*$ by the relation:

$$S_w(\psi_m) = S_w^*\left(\eta \cdot \psi_m\right) \tag{2.58}$$

$$k_{r_w}(\psi_m) = k_{r_w}^*\left(\eta \cdot \psi_m\right) \tag{2.59}$$

$$K_s = \eta^2 \cdot K_s^* \tag{2.60}$$

By definition, porosity is constant in Miller similar media.

Miller similarity can also be seen as a special case of the Leverett J-function (Leverett 1940b), which states that

$$\frac{\psi_m(S_w)\sqrt{K/\Phi}}{\sigma_w \cos(\Theta)} = J(S_w) \tag{2.61}$$

where Θ is the contact angle at the fluid solid interface, K is the permeability of the porous medium and $J(S_w)$ is a generalised function for a certain type of material. Obviously Miller similarity is obtained if contact angle and porosity are kept constant.

2.4. Water Vapour Transport

In dry soils the wetting phase can get disconnected and the wetting phase relative permeability thus can approach zero very quickly, especially in sandy material. Here we want to consider a water/air system. Water vapour transport is then the main transport mechanism for the wetting phase and water transport is gravely underestimated if vapour transport is completely neglected. Due to the high energy associated with the phase change, the transport of water vapour is, of course, in principle a non-isothermal process. However, if the porous medium is dry enough to let the relative humidity decrease noticeably (which is true when vapour transport starts to become the dominant flow process inside a porous medium), the evaporation rates are low enough to keep the temperature approximately constant. The secondary phase of evaporation from a soil surface, where the evaporation rate is controlled by soil moisture and starts to decrease, can be described well with an isothermal model (Schneider-Zapp and Ippisch 2010).

To model isothermal vapour transport in a porous medium, we assume local thermodynamic equilibrium. The most natural notation is obtained using the molar density ν_n^w (amount of substance of water vapour per volume) to quantify the amount of water vapour in the gas phase. If we assume that the ideal gas law is valid, ν_n^w can be calculated from (Rawlins and Campell 1986)

$$\nu_n^w = \frac{p_{\mathsf{sat}}^w(T)}{RT} \exp\left(\frac{\psi_m V_m^w}{RT}\right), \tag{2.62}$$

where V_m^w is the molar volume of liquid water, R is the ideal gas constant, T is the temperature of the porous medium and $p_{\mathsf{sat}}^w(T)$ the partial pressure of water vapour over pure liquid water at temperature T. It can be described with Magnus' formula (Murray 1967) as

$$p_{\mathsf{sat}}^w(T) = 610.78\,Pa\exp\left(\frac{17.2694(T-273.16\,K)}{T-35.86\,K}\right). \tag{2.63}$$

Of course by inversion of this function it is also possible to calculate the equilibrium matric potential from the partial pressure of water vapour.

The diffusive water vapour flux $\boldsymbol{j}_n^w$ is given by

$$\boldsymbol{j}_n^w = -V_m^w D_n^w \nabla \nu_n^w, \tag{2.64}$$

where D_n^w is the diffusion coefficient for water vapour in the porous medium, which can be reformulated, using the chain rule, as

$$\boldsymbol{j}_n^w = -V_m^w D_n^w \left(\frac{\partial \nu_n^w}{\partial T}\nabla T + \frac{\partial \nu_n^w}{\partial \psi_m}\nabla \psi_m\right). \tag{2.65}$$

If only isothermal water vapour transport is considered, the first term in the brackets can be dropped. If equation 2.63 is used to calculate the molar density of the water vapour, we obtain:

$$\boldsymbol{j}_n^w = -D_n^w \frac{p_s^w(T) V_m^{w^2} \exp\left[\frac{\psi_m V_m^w}{RT}\right]}{(RT)^2}\nabla\psi_m. \tag{2.66}$$

A comparison of different models for the dependency of diffusion coefficients in the gas phase on soil moisture by Jin and Jury 1996 showed that one of the models proposed by Millington and Quirk (Millington 1959; Millington and Quirk 1961

$$D_w^n(\theta_w) = \frac{\theta_n^2}{\varPhi^{2/3}}\cdot D_{g,\mathsf{atm}}^w = \frac{(\varPhi-\theta_w)^2}{\varPhi^{2/3}}\cdot D_{g,\mathsf{atm}}^w \tag{2.67}$$

gave the best agreement with experimental data.

As isothermal water transport is only an important transport process under dry conditions, we assume for the calculation of the diffusion coefficient that $\theta_w = 0$ (i.e. $\theta_n = \varPhi$) and obtain

$$D_w^n \approx \varPhi^{4/3} D_{g,\mathsf{atm}}^w. \tag{2.68}$$

Thus the diffusive flux is given by

$$\boldsymbol{j}_n^w = -D_{g,\mathsf{atm}}^w \frac{\varPhi^{4/3}\cdot p_{\mathsf{sat}}^w(T)\cdot (V_m^w)^2 \cdot \exp\left(\frac{\psi_m V_m^w}{RT}\right)}{(RT)^2}\nabla\psi_m, \tag{2.69}$$

which can be conveniently included in Richards' equation by introducing an equivalent conductivity for water transport in the gas phase

$$K_n^w(\psi_m) = D_{g,\mathsf{atm}}^w \frac{\Phi^{4/3} \cdot p_{\mathsf{sat}}^w(T) \cdot (V_m^w)^2 \cdot \exp\left(\frac{\psi_m V_m^w}{RT}\right)}{(RT)^2} \tag{2.70}$$

.

The transport equation then becomes

$$\frac{\partial\left(\theta_w \rho_w\right)}{\partial t} + \nabla \cdot \left[K_n^w(\psi_m) \cdot \nabla\psi_m + \rho_w K \frac{k_{r_w}(S_w)}{\mu_i} \cdot (\nabla\psi_m - \rho_w g \boldsymbol{e_z})\right] + r_w = 0 \tag{2.71}$$

Under very dry conditions the gradient of matric potential will be the dominant driving force compared to gravity while under wet conditions water vapour transport itself will be negligible. Therefore we finally obtain, by neglecting the missing gravity term in the vapour transport part:

$$\frac{\partial\left(\theta_w \rho_w\right)}{\partial t} + \nabla \cdot \left[\left(K_n^w(\psi_m) + \rho_w K \frac{k_{r_w}(S_w)}{\mu_i}\right) \cdot (\nabla\psi_m - \rho_w g \boldsymbol{e_z})\right] + r_w = 0. \tag{2.72}$$

Isothermal water vapour transport thus can be considered by adding an effective water vapour conductivity to the wetting phase conductivity in the mass balance equation for the water component.

2.5. Root Water Uptake

In soils root water uptake by transpirating plants is probably the most important water extraction process. In contrast to bare soil evaporation it is not located at the soil surface but distributed over the whole root zone. Therefore it cannot be modelled by a boundary condition but rather by a sink term. While the potential root water uptake depends mainly on subsurface plant features (growth stadium, leaf area ...) and weather conditions (temperature, humidity, solar radiation ...), the actual root water uptake depends on the availability of water in the root zone. Two effects have to be considered: The ability of a root to locally extract water from the surrounding soil, and the ability of a plant to compensate dry conditions in one part of the root system by higher water uptake in wetter parts of the domain.

We assume for the following that the potential transpiration T_{pot} is already known, e.g. from a plant model. The potential local root water uptake $r_{r_{\mathsf{pot}}}(\boldsymbol{x})$ can then be calculated by multiplication with the local root density $\rho_r(\boldsymbol{x})$, which is normalised to give a partition of unity over the whole domain:

$$r_{r_{\mathsf{pot}}}(\boldsymbol{x}) = T_{\mathsf{pot}} \cdot \frac{\rho_r(\boldsymbol{x})}{\int\limits_{\Omega} \rho_r(\boldsymbol{x})\, dV} \tag{2.73}$$

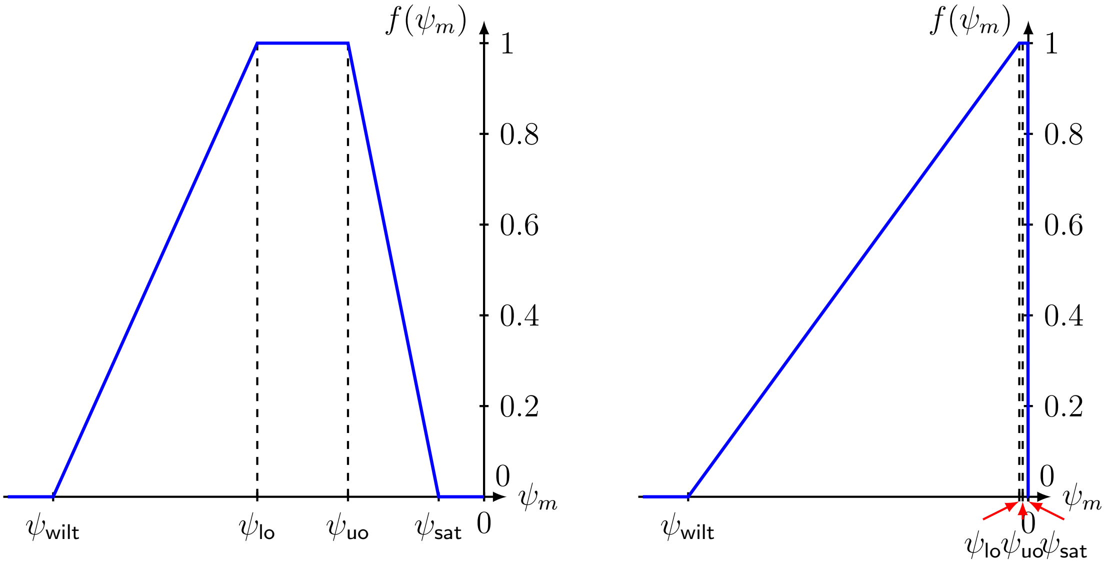

Figure 2.7.: Stress function used in the root water uptake model. Schematic text book representation (left) and with realistic values for the potentials (right).

2.5.1. Water Stress

The actual local root water uptake $r_r(\boldsymbol{x})$ is calculated as a product of the potential local root water uptake $r_{r_{\mathsf{pot}}}(\boldsymbol{x})$ and a stress function $f(\psi_m(\boldsymbol{x}))$ depending on the current local matric potential.

$$r_r(\boldsymbol{x}) = r_{r_{\mathsf{pot}}}(\boldsymbol{x}) \cdot f(\psi_m(\boldsymbol{x})) \tag{2.74}$$

This model for root water uptake follows an approach from Feddes et al. 1976, which has been used in many studies since, mostly in one-dimensional simulations. It has been used for two-dimensional heterogeneous systems by Kuhlmann et al. 2012.

The shape of the stress function $f(\psi_m) : \psi_m \in \mathbb{R} \to [0,1]$ is unknown and plant-specific. A simple piecewise linear function is normally used (figure2.7). Root water uptake is assumed to be zero above a potential ψ_{sat} due to limited oxygen availability for the roots, and below a potential ψ_{wilt} called the wilting point. It is assumed to be optimal in a potential range $[\psi_{\mathsf{lo}} : \psi_{\mathsf{uo}}]$. The four potentials are free (plant-specific) parameters of the model. The potential at the lower end of the optimal range ψ_{lo} can also depend on the transpiration rate.

2.5.2. Compensation

If the water uptake of a plant is reduced by stress in some part of the root system, this can be compensated by a higher uptake in unstressed parts of the root system. This is called compensation. The importance of compensation and which models are appropriate to represent it, is still under intensive discussion (Jarvis 1989; Li et al. 2001; Guswa

2005; Varado et al. 2006; Lier et al. 2008; Kuhlmann et al. 2012). Variants between full compensation (full transpiration as long as there is still water accessible to roots somewhere) and no compensation exist.

A simple way to implement full compensation is the scaling of the potential root water uptake with the ratio between potential and actual transpiration r_s:

$$r_r(\boldsymbol{x}) = r_{r_{\mathsf{pot}}}(\boldsymbol{x}) \cdot f(\psi_m(\boldsymbol{x})) \cdot r_s \tag{2.75}$$

$$r_s = \max\left(1, \frac{T_{\mathsf{pot}}}{\int\limits_{\Omega} r_r(\boldsymbol{x})\, dV}\right) \tag{2.76}$$

It makes sense to introduce an upper limit $r_{s_{\mathsf{max}}}$ for r_s. On the one hand an increase of r_s might not further increase root water uptake if the soil is dry everywhere and uptake limited by the stress function, on the other hand the potential of roots to increase water uptake for compensation is limited. According to Guswa 2005 the capability of roots for compensation “can range from one to more than two” depending on the plant. We thus finally obtain

$$r_r(\boldsymbol{x}) = r_{r_{\mathsf{pot}}}(\boldsymbol{x}) \cdot f(\psi_m(\boldsymbol{x})) \cdot \min\left(r_{s_{\mathsf{max}}}, r_s\right). \tag{2.77}$$

The implicit equation 2.75 can be solved either iteratively or explicitly by using values from a prior time step.

2.6. Model Selection

In this work we want to present an efficient and massively parallel numerical model for the simulation of water transport in strongly heterogeneous soils. The model will be based on Richards' equation (section 2.1.2) as the non-wetting phase is formed by air for which the basic assumption of Richards' equation is mostly well fulfilled. Isothermal water vapour transport will be (optionally) included as an increase of the hydraulic conductivity (section 2.4), hysteresis in the form of the modified dependent domain model of Mualem (section 2.3.3.1). Root water uptake is simulated with the approach described in section 2.5. Dynamic effects will be introduced to study the effect of macroscopic heterogeneity on water transport. This will be done with the model of Ross and Smettem (section 2.2.5.5), as it is easy to implement yet allows an impact assessment of dynamic effects.

3. Numerical Solution of Richards' Equation

The mathematical model described in section 2.6 essentially requires the numerical solution of Richards' equation. Hysteresis and the dynamic effects of Ross and Smettem only modify the evaluation of the time derivative. Isothermal water vapour transport affects the conductivity function and root water uptake the sink term. We thus want to concentrate in this chapter on a solution of Richards' equation in the form

$$\frac{\partial(\rho_w \theta_w(\psi_m))}{\partial t} - \nabla \cdot \left[\frac{\rho_w \mathsf{K} k_r(\psi_m)}{\mu_w} \left(\nabla \psi_m - \rho_w g \boldsymbol{e}_z \right) \right] + r_w = 0. \tag{3.1}$$

with K being at most a diagonal tensor.

For all problems to be solved here, the PDE is posed on a finite domain Ω with boundary $\partial\Omega$ and for a finite time interval $\mathcal{T} = [t_0 : t_{\mathsf{end}}]$. We will use ψ_m as the independent variable.

3.1. Requirement Specification

All numerical approximations have their individual strengths and weaknesses and there is no general best or worst choice. Before the selection of a numerical approach for the solution of Richards' equation, it is thus necessary to analyse the requirements of the resulting model.

The program to be developed in this work is intended to be used for teaching and for research applications. The intended application area is the simulation of structured and highly heterogeneous porous media at the scale of laboratory samples (cm to dm) up to the field scale (1-100 m). At least a two-scale heterogeneity should be simulated. The structures will have deterministic (soil horizons) and stochastic components (sub-scale heterogeneity). Interest is often not only in water transport but in the transport of a dissolved solute.

From the intended application area the following requirements can be derived:

- A flexible input and management for the soil structure and the corresponding hydraulic properties is necessary.
- It is hardly possible to manage the necessary high spatial resolution with an unstructured grid, and as the structure will be obtained from measurements (e.g. georadar, digital images, X-ray tomography) or generators using a structured grid, it makes sense to use a structured grid for the discretisation.

- High gradients of matric potential can occur at the surface during evaporation, thus a tensor-product grid is necessary.
- High-spatial resolution with larger domains requires the use of massively parallel computing.
- As the heterogeneity can lead to sharp contrasts of the hydraulic functions, a very stable numerical scheme is necessary. On the other hand as the permeability field is piecewise constant with jumps at element boundaries and thus the solution is expected to have a low regularity, the use of higher-order methods is not necessary.
- As the flow field calculated from the water transport equation will be used for solute transport, a locally mass-conservative numerical scheme is necessary.
- The complex results require the use of powerful visualisation tools. It makes sense to use existing programs like Data Explorer or ParaView for this task and thus to add appropriate output formats to the model.
- An easy specification of model parameters and features is necessary to make the program useful for students and geoscientists.
- Hydraulic parameters are often unknown and have to be estimated. The program thus must be easily usable in an optimisation context.

3.2. Classification of the Equation

The basis for the selection of an appropriate numerical solution scheme is the correct classification of the partial differential equation. Richards' equation obviously is a non-linear partial differential equation of second order. For a more detailed classification we need to bring it in a form where only the independent variable occurs in the derivatives.

For the two-dimensional case a linear second-order PDE has the general form:

$$\begin{aligned} a(x,y)\frac{\partial^2\psi_m}{\partial x^2}(x,y) + 2b(x,y)\frac{\partial^2\psi_m}{\partial x\partial y}(x,y) + c(x,y)\frac{\partial^2\psi_m}{\partial y^2}(x,y) \\ + d(x,y)\frac{\partial\psi_m}{\partial x}(x,y) + e(x,y)\frac{\partial\psi_m}{\partial y}(x,y) + f(x,y)\psi_m(x,y) + g(x,y) = 0. \end{aligned} \tag{3.2}$$

At a point (x,y) the PDE can be classified according to the first three terms (main part) into

elliptic if $\det\left(\begin{smallmatrix} a & b \\ b & c\end{smallmatrix}\right) = a(x,y)c(x,y) - b^2(x,y) > 0$

hyperbolic if $\det\left(\begin{smallmatrix} a & b \\ b & c\end{smallmatrix}\right) = a(x,y)c(x,y) - b^2(x,y) < 0$

parabolic if $\det\left(\begin{smallmatrix} a & b \\ b & c\end{smallmatrix}\right) = a(x,y)c(x,y) - b^2(x,y) = 0$ and

$$\text{Rank}\begin{bmatrix} a & b & d \\ b & c & e \end{bmatrix} = 2 \text{ in } (x,y)$$

For higher dimensional problems the general form is

$$\sum_{i,j=1}^{n} a_{ij}(\boldsymbol{x}) \frac{\partial^2 \psi_m}{\partial x_i \partial x_j} + \sum_{i=1}^{n} a_i(\boldsymbol{x}) \frac{\partial \psi_m}{\partial x_i} + a_0(\boldsymbol{x}) \psi_m - f(\boldsymbol{x}) = 0 \quad \text{in } \varOmega. \tag{3.3}$$

and the classification is done according to the eigenvalues of the matrix formed by the coefficients $a_{ij}(\boldsymbol{x})$ of the second order derivatives.

For the sake of simplicity we start with the one-dimensional Richards' equation without gravity:

$$\frac{\partial(\rho_w \theta_w(\psi_m))}{\partial t} - \frac{\partial}{\partial x}\left[\frac{\rho_w K k_r(\psi_m)}{\mu_w} \frac{\partial \psi_m}{\partial x}\right] + r_w = 0 \tag{3.4}$$

Assuming a constant water density ρ_w and applying the product rule we obtain

$$\underbrace{\frac{\partial \theta_w}{\partial \psi_m}}_{C(\psi_m)} \frac{\partial \psi_m}{\partial t} - \underbrace{\frac{K k_r(\psi_m)}{\mu_w}}_{D(\psi_m)} \frac{\partial^2 \psi_m}{\partial x^2} + \underbrace{\left[-\frac{\partial}{\partial x}\left(\frac{K k_r(\psi_m)}{\mu_w}\right)\right]}_{v(\psi_m)} \cdot \left(\frac{\partial \psi_m}{\partial x}\right) + \frac{r_w}{\rho_w} = 0. \tag{3.5}$$

This is a non-linear convection-diffusion equation with the diffusion coefficient $D(\psi_m) = \frac{K k_r(\psi_m)}{\mu_w}$, the velocity $v(\psi_m) = -\frac{\partial}{\partial x}\left(\frac{K k_r(\psi_m)}{\mu_w}\right)$ and the specific soil water capacity $C(\psi_m)$.

If gravity is considered, by applying the chain rule to the gravity term we obtain

$$\begin{aligned} &\underbrace{\frac{\partial \theta_w}{\partial \psi_m}}_{C(\psi_m)} \frac{\partial \psi_m}{\partial t} - \underbrace{\frac{K k_r(\psi_m)}{\mu_w}}_{D(\psi_m)} \frac{\partial^2 \psi_m}{\partial z^2} \\ &+ \underbrace{\left[-\frac{\partial}{\partial z}\left(\frac{K k_r(\psi_m)}{\mu_w}\right) + \rho_w g \frac{\partial}{\partial \psi_m}\left(\frac{K k_r(\psi_m)}{\mu_w}\right)\right]}_{v(\psi_m)} \cdot \left(\frac{\partial \psi_m}{\partial z}\right) + \frac{r_w}{\rho_w} = 0. \end{aligned} \tag{3.6}$$

This only adds the term $\rho_w g \frac{\partial}{\partial \psi_m}\left(\frac{K k_r(\psi_m)}{\mu_w}\right)$ to the velocity.

A classification is now possible. We get a zero determinant

$$\det \begin{pmatrix} -\frac{K k_r(\psi_m)}{\mu_w} & 0 \\ 0 & 0 \end{pmatrix} = 0 \tag{3.7}$$

and the additional rank condition

$$\text{Rank} \begin{bmatrix} -\frac{K k_r(\psi_m)}{\mu_w} & 0 & -\frac{\partial}{\partial z}\left(\frac{K k_r(\psi_m)}{\mu_w}\right) + \rho_w g \frac{\partial}{\partial \psi_m}\left(\frac{K k_r(\psi_m)}{\mu_w}\right) \\ 0 & 0 & \frac{\partial \theta_w}{\partial \psi_m} \end{bmatrix} = 2. \tag{3.8}$$

The coefficients depend on space and time. Thus the type of Richards' equation can be different at different points in space or time. Richards' equation is a formally parabolic PDE. However, if the porous medium is saturated, the derivative of water content vanishes and Richards' equation becomes an ordinary differential equation (in one-dimension) or an elliptic PDE (in two- or three space dimensions).

In its discretised form Richards' equation can also become effectively hyperbolic. For a grid with grid spacing h Richards' equation will become effectively convection-dominated if the (non-linear) grid Peclet number $\frac{hv}{2D}$ is larger than 1:

$$\frac{h\left|\frac{\partial}{\partial z}\left(\frac{Kk_r(\psi_m)}{\mu_w}\right)+\rho_w g\frac{\partial}{\partial\psi_m}\left(\frac{Kk_r(\psi_m)}{\mu_w}\right)\right|}{2\frac{Kk_r(\psi_m)}{\mu_w}}>1 \tag{3.9}$$

For a homogeneous material (i.e. if K and μ_w do not depend on z) this condition can be simplified. We can then apply the chain rule again to the first part of the velocity and get:

$$\frac{\left|\frac{K}{\mu_w}\frac{\partial k_r(\psi_m)}{\partial\psi_m}\cdot\left(\frac{\partial\psi_m}{\partial z}-\rho_w g\right)\right|}{2K\frac{k_r(\psi_m)}{\mu_w}}>\frac{2}{h} \tag{3.10}$$

$$\Leftrightarrow\left|\frac{1}{k_r(\psi_m)}\frac{\partial k_r(\psi_m)}{\partial\psi_m}\cdot\left(\frac{\partial\psi_m}{\partial z}-\rho_w g\right)\right|>\frac{2}{h} \tag{3.11}$$

$$\Leftrightarrow\left|\frac{\partial\ln\left(k_r(\psi_m)\right)}{\partial\psi_m}\cdot\left(\frac{\partial\psi_m}{\partial z}-\rho_w g\right)\right|>\frac{2}{h} \tag{3.12}$$

Thus convection will be the dominant process if gravity and potential gradient point into the same direction (which is the case during imbibition from above) and if the relative permeability curve is very steep. Thus it should occur most frequently in well graded and coarse textured materials, like coarse sands[1].

Richards' equation is a potentially degenerate non-linear second-order partial differential equation, which can become elliptic, parabolic and effectively hyperbolic at the same time in different parts of the domain (this is also true for problems with two and three space dimensions). This needs to be taken into account in the choice of the numerical scheme.

3.3. Discretisation

Based on the requirements specified in section 3.1 a method of lines approach with a cell-centred Finite-Volumes discretisation in space and an implicit Euler scheme in time was chosen.

[1] All quantities in condition 3.11, apart from the pressure derivative of the relative permeability curve have to be calculated anyhow to solve the equation. Thus this might be a cheap estimator for the necessity for regularisation of Richards' equation e.g. by upwinding.

3.3.1. Spatial Discretisation

A weak form of Richards' equation is obtained by integration of the PDE times weighting functions w_i over the domain Ω

$$\int\limits_{\Omega} w_i \cdot \left(\frac{\partial(\rho_w \theta_w(\psi_m))}{\partial t} + \nabla \cdot \boldsymbol{j}_w + r_w \right) dV = 0$$

$$\text{with } \boldsymbol{j}_w = -\frac{\rho_w \mathbf{K} k_r(\psi_m)}{\mu_w} \left(\nabla\psi_m - \rho_w g \boldsymbol{e}_z \right) \tag{3.13}$$

Let $\mathcal{E}_h = \{e_i\},\ i = 0, \ldots, N$ be a partitioning of the domain Ω into elements e_i (lines (1D), rectangles (2D) or cuboids (3D)) with $\bar{\Omega} = \bigcup\limits_{i=0}^{N} \bar{e}_i$ and $e_i \cap e_j = \emptyset\ \forall i \neq j$. h is an indicator for the mesh width. For a tensor-product grid let $h = \min(\boldsymbol{h_i})$ with $\boldsymbol{h_i}, i = x, y, z$ being the vector of element width in x, y and z direction. We now choose the weighting functions to be piecewise constant functions which are one on element e_i and zero elsewhere. We then obtain n equations one for each element:

$$\int\limits_{e_i} \left(\frac{\partial(\rho_w \theta_w(\psi_m))}{\partial t} + \nabla \cdot \boldsymbol{j_w} + r_w \right) dV = 0. \tag{3.14}$$

We split the integral in three parts and apply the Satz of Gauß to the divergence

$$\int\limits_{e_i} \frac{\partial(\rho_w \theta_w(\psi_m))}{\partial t}\, dV + \int\limits_{\partial e_i} \boldsymbol{j_w} \cdot \boldsymbol{n}\, dA + \int\limits_{e_i} r_w\, dV = 0. \tag{3.15}$$

where ∂e_i is the boundary of element e_i and $\boldsymbol{n}$ is the unit outer normal. We can exchange the time derivative and the integration and obtain an equation for the time evolution of the averages. For the chosen tensor product grid each element e_i has $2d$ boundary faces f_{ij}, we thus can also split the boundary integral into $2d$ face integrals:

$$\frac{\partial}{\partial t} \int\limits_{e_i} \rho_w \theta_w(\psi_m)\, dV + \sum_{j=0}^{2d} \int\limits_{f_{ij}} \boldsymbol{j_w} \cdot \boldsymbol{n}\, dA + \int\limits_{e_i} r_w\, dV = 0. \tag{3.16}$$

Please note that the index j in f_{ij} is a local number of the face and *not* the index of the adjacent element.

We now want to find a solution in the broken polynomial space of order zero (piecewise constants)

$$W_h(\Omega, \mathcal{E}_h) = \left\{ U \in C^0(\Omega) : U|_e = \mathsf{const}\ \ \forall e \in \mathcal{E}_h \right\}. \tag{3.17}$$

where U_i represents the cell average of ψ_m on element e_i. We can obtain an approximation of the weak form by quadrature using the midpoint rule

$$\int_{e_i} \rho_w \theta_w(\psi_{m_i})\, dV \approx \rho_w \cdot \theta_w(U_i) \cdot V_i = \mathcal{M}_i(\boldsymbol{\psi}_m) \tag{3.18}$$

$$\int_{f_{ij}} \boldsymbol{j_w} \cdot \boldsymbol{n}\, dA \approx J_w(U_i, U_j) \cdot n_j \cdot A_{ij} = \mathcal{A}_{ij}(\boldsymbol{\psi}_m) \tag{3.19}$$

$$\int_{e_i} r_w\, dV = R_w(U_i) \cdot V_i = \mathcal{Q}_i(\boldsymbol{\psi}_m). \tag{3.20}$$

$J_w(U_i, U_{k_j})$ is the numerical flux function, which still has to be determined and k_j is the index of the element sharing the element face f_{ij}. In our two point flux approximation $J_w(U_i, U_{k_j})$ only depends on the cell averages of the two cells with the common face f_{ij}. As a consequence of the tensor product grid, $J_w(U_i, U_j)$ is a scalar function and $n_j \in \{-1, 1\}$.

As equations 3.18, 3.19 and 3.20 can be interpreted as applications of a midpoint rule to the integrals they yield second order accurate approximations. We obtain semi-discrete equations for the cell averages:

$$\frac{\partial \left(\rho_w \theta_w(U_i) V_i\right)}{\partial t} + \sum_{j=0}^{2d} \left(J_w(U_i, U_j) \cdot n_j \cdot A_{ij}\right) + R_w \cdot V_i = 0, \tag{3.21}$$

or in abbreviated form

$$\frac{\partial \mathcal{M}_i(U)}{\partial t} + \sum_{j=0}^{2d} \mathcal{A}_{ij}(U) + \mathcal{Q}_i(U) = 0. \tag{3.22}$$

In fact, equation 3.21 is the equation for elements with interior faces only. The domain boundary will be treated in section 3.3.4 and will lead to different values of $\mathcal{A}_{ij}(U)$.

3.3.2. Numerical Flux Function

A good numerical flux function can be obtained by the analysis of the one-dimensional problem in a domain $\Omega =]0, \ell[$ with the Dirichlet boundary conditions $\psi_m(0) = \psi_0$ and $\psi_m(\ell) = \psi_\ell$. The one-dimensional (horizontal) Richards' equation is

$$\frac{\partial(\rho_w \theta_w(\psi_m))}{\partial t} - \frac{\partial}{\partial x}\left[\frac{\rho_w K k_r(\psi_m)}{\mu_w}\frac{\partial \psi_m}{\partial x}\right] + r_w = 0. \tag{3.23}$$

If there are no sources or sinks, if we neglect the time derivative and introduce the conductivity $k(\psi_m, x) = \rho_w K k_r(\psi_m)/\mu_w$

$$\frac{\partial j_w}{\partial x} = 0 \tag{3.24}$$

$$j_w = -k(\psi_m, x)\frac{\partial \psi_m}{\partial x} \tag{3.25}$$

As the spatial derivative of j_w is zero, the flux has to be a constant j_0.

$$j_0 = -k(\psi_m, x)\frac{\partial \psi_m}{\partial x} \quad \Leftrightarrow \quad \frac{\partial \psi_m}{\partial x} = -\frac{j_0}{k(\psi_m, x)} \tag{3.26}$$

Integration of both sides over the domain yields

$$\int_0^\ell \frac{\partial \psi_m}{\partial x}\,dx = [\psi_m(x)]_0^\ell = \psi_\ell - \psi_0 = -j_0 \int_0^\ell \frac{1}{k(\psi_m, x)}\,dx \tag{3.27}$$

$$j_0 = -\underbrace{\frac{\ell}{\int_0^\ell \frac{1}{k(\psi_m,x)}\,dx}}_{\text{eff.conductivity}} \cdot \underbrace{\frac{\psi_\ell - \psi_0}{\ell}}_{\text{approx.gradient}} \tag{3.28}$$

If we assume that the conductivity distribution is piecewise constant on $\ell_l < \ell$

$$k(x) = \begin{cases} k_l & x \leq \ell_l \\ K_r & x > \ell_l \end{cases} \tag{3.29}$$

(this is the case when the absolute permeability K is different and the effect of relative permeability is negligible), we can calculate the effective conductivity k_{eff}:

$$k_{\mathsf{eff}} = \frac{\ell}{\int_0^\ell \frac{1}{k(x)}\,dx} = \frac{\ell}{\ell_l \frac{1}{k_l} + (\ell - \ell_l)\frac{1}{k_r}} = \frac{\ell k_l k_r}{\ell_l k_r + (\ell - \ell_l) k_l} \tag{3.30}$$

and obtain a weighted harmonic mean. For saturated conditions (where all the assumptions hold) this effective permeability yields the exact solution (in 1D) if the spatial derivative is replaced by a difference quotient. For other distributions of $k(x)$ we would obtain other effective conductivities. If K is homogeneous and the difference only results from $k_r(\psi_m)$, a geometric mean of the conductivities might give a better approximation.

From the analysis of the one-dimensional problem we thus get the following numerical flux function for the flux over face f_{ij} between the elements e_i and e_j with conductivities k_i and k_j and element width h_i and h_j:

$$J_w(U_i, U_j) = -k_{\mathsf{eff}}(U_i, U_j)\frac{U_i - U_j}{\frac{(h_i + h_j)}{2}} \tag{3.31}$$

$$k_{\mathsf{eff}} = \frac{(h_i + h_j) k_i k_j}{h_j k_i + h_i k_j}. \tag{3.32}$$

3.3.3. Upwinding

If Richards' equation becomes effectively hyperbolic, a central difference approximation will not produce a stable and correct solution and some stabilisation is necessary. In two-phase flow an upwinding of the mobility $\lambda_w = \frac{k_r(\psi_m)}{\mu_w}$ is an often used approach (Helmig and Huber 1998). However, this will not work for elements with different relative permeability functions. Bastian 1999 presented a formulation with interface conditions for a vertex-centred Finite-Volume method.

Effective hyperbolicity is linked to the passing of sharp fronts through the domain. Upwinding can be viewed as assuming that the front has already passed and using the values behind the front. While saturation and relative permeability might be discontinuous over element interfaces, potentials will be continuous. We can thus perform an upwinding of matric potential, calculate saturations and relative permeabilities from the element functions and calculate the effective conductivity as a harmonic mean of the element conductivities. In the case of homogeneous material this is equivalent to an upwinding of mobilities. The direction of flow can be easily determined from the sign of the gradient of total potential $\nabla\psi_m - \rho_w g$.

With the upwind operator (here presented only for the vertical direction[2] with $z_i < z_j$)

$$\mathsf{upwind}(U_i, U_j) = \begin{cases} U_i \text{ if } \frac{U_i - U_j}{\ell} - \rho_w g \geq 0 \\ U_j \text{ if } \frac{U_i - U_j}{\ell} - \rho_w g < 0 \end{cases} \tag{3.33}$$

the effective permeability between two elements e_i and e_j with average matric potential U_i and U_j and conductivity functions $k_i(\psi_m)$ and $k_j(\psi_m)$ can be calculated from

$$k_{\mathsf{eff,upw}}(U_i, U_j) = \frac{h_i + h_j}{h_i \frac{1}{k_i(\mathsf{upwind}(U_i, U_i))} + h_j \frac{1}{k_j(\mathsf{upwind}(U_i, U_j))}}. \tag{3.34}$$

Thus we can finally define our numerical flux function for heterogeneous porous media with potential hyperbolicity to be

$$J_w(U_i, U_j) = -k_{\mathsf{eff,upw}}(U_i, U_j) \frac{U_i - U_j}{\frac{(h_i + h_j)}{2}}. \tag{3.35}$$

3.3.4. Boundary Conditions

For a parabolic partial difference equation initial conditions and boundary conditions on all space boundaries are needed to get a well-posed problem. As there is always a diffusive contribution, this remains true even if Richards' equation gets effectively hyperbolic. For an elliptic problem it is further necessary to have Dirichlet boundary conditions at least at one point (else the solution is only fixed up to a constant). Besides Neumann and Dirichlet boundary conditions some additional variants are common for porous media flow.

Boundary conditions for the cell-centred Finite-Volume scheme are specified face-wise. In contrast to the standard Finite-Element or the vertex-centred Finite-Volume scheme, the choice of the boundary conditions does not influence the number of unknowns or the position of the boundary and the solution is still obtained in an unconstrained space.

[2] Please remember that the vertical axis is pointing in the direction of gravity

3.3.4.1. Neumann Boundary Condition

At Neumann boundaries the normal flux ϕ_N is prescribed. In the cell-centred Finite-Volume scheme this is easily incorporated by approximating the integral 3.19 at the boundary face f_{ij} by

$$\int\limits_{f_{ij}} \boldsymbol{j_w} \cdot \boldsymbol{n}\, dA \approx \phi_N \cdot A_{ij} = \mathcal{A}_{ij_N}. \tag{3.36}$$

3.3.4.2. Dirichlet Boundary Condition

At Dirichlet boundaries the value of the potential ψ_D is prescribed. In the approximation of the flux it has to be taken into account, that the distance is only $^h/_2$

$$\mathcal{A}_{ij_D} = J_{w_D}(U_i, \psi_D) \cdot n_j \cdot A_{ij} = -k_{\mathsf{eff,upw}}(U_i, \psi_D) \frac{U_i - \psi_D}{\frac{h_i}{2}} \cdot n_j \cdot A_{ij}. \tag{3.37}$$

3.3.4.3. Limited-Flux Boundary Condition

For the simulation of the upper boundary of soils or certain lysimeters it is convenient to pose a boundary condition with a combined constraint on the potential and the flux. It can be understood as a kind of Signorini problem and is used when a flux is to be prescribed, but a pure Neumann boundary condition can result in non-physical results. Typical application cases for the limited inflow boundary condition are:

- Precipitation rates are known but pure Neumann conditions can lead to very high pressures at the surface if the precipitation rate is higher than the conductivity of the soil. If surface run-off is not explicitly simulated instantaneous run-off can be modelled by limiting the flux into the domain to not exceed the flux obtained with a matric potential of zero at the upper boundary.
- Free-drainage lysimeters are containers filled with soil material which are open at the lower boundary (or placed on coarse gravel, which is nearly the same). Outflow can then only occur if the pressure at the lower boundary is positive. This can be modelled by limiting the flux into the domain to zero but allow fluxes out of the domain calculated with a prescribed matric potential of zero. This is often also called a seepage boundary condition.

A typical application case for the limited outflow boundary condition is:

- Potential evaporation is known, but cannot be sustained if the soil is already too dry. Posing a Neumann boundary condition can lead to a break-down of the simulation. Actual evaporation can be modelled by limiting the flux out of the domain to not exceed the flux obtained with a minimal prescribed matric potential at the upper boundary.

This kind of boundary conditions is often realised with a switch of the boundary condition type from Dirichlet to Neumann and vice versa. However, it is easier to formulate the limited inflow boundary condition as

$$\mathcal{A}_{ij_{\mathsf{LIF}}} = J_{w_{\mathsf{LIF}}}(U_i, \psi_D, \phi_N) \cdot n_j \cdot A_{ij} = \max\left(-k_{\mathsf{eff,upw}}(U_i, \psi_D)\frac{U_i - \psi_D}{\frac{h_i}{2}} n_j, \phi_N\right) \cdot n_j \cdot A_{ij}. \tag{3.38}$$

and the limited outflow boundary condition as

$$\mathcal{A}_{ij_{\mathsf{LOF}}} = J_{w_{\mathsf{LOF}}}(U_i, \psi_D, \phi_N) \cdot n_j \cdot A_{ij} = \min\left(-k_{\mathsf{eff,upw}}(U_i, \psi_D)\frac{U_i - \psi_D}{\frac{h_i}{2}} n_j, \phi_N\right) \cdot n_j \cdot A_{ij}. \tag{3.39}$$

This is of course also a switch. However, both boundary conditions can be implemented as in the form of a flux. This would also be beneficial for vertex-centred Finite-Volume or Finite-Element schemes as there is no need to include the Dirichlet boundary conditions as essential boundary conditions in the Ansatz space.

3.3.4.4. Surface-Storage

The limited influx boundary condition assumes instant run-off of ponding water. This is not always realistic. Water might just be stored in (unresolved) bumps at the surface. An alternative to the limited influx boundary condition is the introduction of surface storage. If the matric potential at the soil surface gets positive, the ponding water is properly accounted for in the storage term of the elements at the surface and can infiltrate later. For the surface elements we get

$$\theta_w = S_w(\psi_m)\varPhi + \max\left(0, \frac{\psi_m}{h\rho_w g} - {}^1\!/_2\right) \tag{3.40}$$

where h is the element height of the uppermost element. In contrast to full consideration of surface run-off, there is no horizontal flow of the surface water. As high precipitation rates usually do not last long, surface storage in combination with Neumann boundary conditions yields a realistic approximation for many practical cases.

3.3.4.5. Gravity-Flow Boundary Condition

In soils the potential gradients are strongly damped in deeper parts of the profile. If the lower boundary of the simulation domain is still above groundwater level but there is always a net percolation, it is possible to neglect the potential gradient and assume that gravity is the only driving force:

$$\mathcal{A}_{ij_G} = J_{w_G}(U_i) \cdot n_j \cdot A_{ij} = k(U_i)\rho_w g \cdot n_j \cdot A_{ij}. \tag{3.41}$$

3.3.5. Initial Conditions

Initial conditions are boundary conditions in the time direction. Essentially just the value of the independent variable has to be specified, in this case the matric potential. Simple as it seems there still are different choices of initial conditions depending on the situation to simulate. If no measurement values are available, a constant potential everywhere is often a good initial condition for steady-state simulations. For time-dependent simulations a given matric potential at the lower boundary or the matric potential corresponding to a given saturation, and hydraulic equilibrium above

$$\psi_{m_{\mathsf{initial}}} = \psi_0 + \rho_w g(z_0 - z) \tag{3.42}$$

is a sensible choice. For sandy materials it makes sense to use hydraulic equilibrium in the wetter part and a constant potential above, to avoid too dry initial conditions. For hysteretic material also the hysteresis state (drainage or imbibition) and the last reversal point must be given (if the initial condition is not on one of the main branches).

3.4. Time Discretisation

With the spatial discretisation and appropriate boundary conditions we obtain a system of non-linear ordinary differential equations for the time evolution of the potentials.

We introduce the partitioning of $\mathcal{T}$ into the discrete time sequence $t_k,\ k = 0, \dots, K$ with $\mathit{\Delta} t_k = t_{k+1} - t_k > 0$ and $t_{\mathsf{end}} = t_0 + \sum_{k=0}^{K-1} \mathit{\Delta} t_k$. The solution U^0 at t_0 is given by the initial condition. We obtain the solution U^{k+1} at t_{k+1} from U^k at t_k by integration of the semi-discrete equation 3.22 over the interval $\tau_k = (t_{k+1} - t_k)$:

$$\int\limits_{t_k}^{t_{k+1}} \left(\frac{\partial \mathcal{M}_i(U)}{\partial t} \right) dt = - \int\limits_{t_k}^{t_{k+1}} \left[\sum_{j=0}^{2d} \mathcal{A}_{ij}(U) + \mathcal{Q}_i(U) \right] dt. \tag{3.43}$$

The integral on the left side of the equation can be directly evaluated. For the integral on the right side we have to use an integration rule. The easiest integration is obtained by a one-step $\mathit{\Theta}$ method

$$\begin{aligned} \mathcal{M}_i(U^{k+1}) - \mathcal{M}_i(U^k) = & - \mathit{\Theta} \cdot \tau^k \cdot \left[\sum_{j=0}^{2d} \mathcal{A}_{ij}(U^{k+1}) + \mathcal{Q}_i(U^{k+1}) \right] \\ & - (1 - \mathit{\Theta}) \cdot \tau^k \cdot \left[\sum_{j=0}^{2d} \mathcal{A}_{ij}(U^k) + \mathcal{Q}_i(U^k) \right] \end{aligned} \tag{3.44}$$

With $\mathit{\Theta} = 0$ we obtain the explicit Euler scheme, with $\mathit{\Theta} = 1$ the implicit Euler scheme. Both are first order accurate in time, but while the implicit Euler scheme is unconditionally stable in the infinity norm and the L_2 norm, the explicit Euler scheme is only stable when a time step restriction is obeyed. The Crank-Nicholson scheme with $\mathit{\Theta} = 1/2$ is second order but is only unconditionally stable in the L_2 norm.

As we are interested in maximal stability, the implicit Euler scheme is used in this work. As the discretised equations result from a local mass balance, the scheme is locally and thus also globally mass conservative.

3.5. Linearisation

The discretised equations form a non-linear equation system

$$\boldsymbol{f}(\boldsymbol{U}^{k+1}) = 0 \tag{3.45}$$

with

$$f_i(\boldsymbol{U}^{k+1}) = \mathcal{M}_i(U^{k+1}) - \mathcal{M}_i(U^k) + \tau^k \sum_{j=0}^{2d} \mathcal{A}_{ij}(U^{k+1}) + \tau^k \mathcal{Q}_i(U^{k+1}). \tag{3.46}$$

and the vector of unknowns $\boldsymbol{U}^{k+1} \in \mathbb{R}^N$.

Farthing and Miller 2003 compared Newton's method and fix-point iteration (also called Picard iteration) for the non-linear equation systems obtained from a cell-centred Finite Differences discretisation of Richards' equation with steady-state test cases, trying to obtain directly the steady-state solution or using time dependent solver to converge to the steady-state solution. They found that "*. . . Newton's method with a line search is the most efficient approach for obtaining steady-state solutions to Richards' Equation*" and that "*if Newton's method fails or performs poorly for a given steady-state problem, it is worth examining a range of linear solver and line-search parameters before abandoning a Newton approach*".

The non-linear equation system in this work is thus solved with Newton's method following the procedure of Bastian 1999, where in each step of an iterative scheme a correction to the current approximation of the solution is obtained. Starting with an initial guess $\boldsymbol{U}^{k+1,0}$ (the solution from the last time step or the initial condition) in each step m the correction $\boldsymbol{v}^{k+1,m}$ is calculated by solution of the linear equation system

$$\boldsymbol{\mathcal{J}}(\boldsymbol{U}^{k+1,m}) \cdot \boldsymbol{v}^{k+1,m} = \boldsymbol{f}(\boldsymbol{U}^{k+1,m}), \tag{3.47}$$

where $\boldsymbol{\mathcal{J}}(\boldsymbol{U}^{k+1,m})$ is the Jacobian with

$$\mathcal{J}_{ij}(\boldsymbol{U}^{k+1,m}) = \left.\frac{\partial f_i}{\partial U_j}\right|_{\boldsymbol{U}^{k+1,m}} \tag{3.48}$$

The Jacobian is assembled by numerical differentiation

$$\mathcal{J}_{ij}(\boldsymbol{U}^{k+1,m}) \approx \frac{f_i(\boldsymbol{U}^{k+1,m} + \mathit{\Delta} U_j \boldsymbol{e}_j) - f_i(\boldsymbol{U}^{k+1,m})}{\mathit{\Delta} U_j} \tag{3.49}$$

with

$$\mathit{\Delta} U_j = \sqrt{\epsilon} \cdot (1 + |\boldsymbol{U}_j^{k+1,m}|) \tag{3.50}$$

if ϵ is the precision of the floating point data type used to store U.

Convergence is determined by a sufficient reduction of the non-linear defect

$$d = ||\boldsymbol{f}(\boldsymbol{U}^{k+1,m})||_2 \leq \epsilon_{\mathsf{nl}} \cdot ||\boldsymbol{f}(\boldsymbol{U}^{k+1,0})||_2 = \epsilon_{\mathsf{nl}} \cdot d_0 \tag{3.51}$$

Newton's method only converges if the initial guess is sufficiently close to the solution. To increase the area of convergence, a line search can be used, where the correction is multiplied by a damping factor λ^k, $\lambda \in]0:1[$ in the kth step of the line search until the approximation by the linearisation is good enough to get an improvement. The required reduction is linked to the size of the damping factor (Bastian 1999; Braess 1992) by

$$||\boldsymbol{f}\left(\boldsymbol{U}^{k+1,m} + \alpha \boldsymbol{v}^{k+1,m}\right)||_2 < (1 - {}^1\!/\!{}_4 \cdot \lambda^k) \cdot d_{m-1}. \tag{3.52}$$

The solution is sped up by using an inexact Newton's method. As the initial guess is usually far away from the correct solution, it is not necessary to solve the linear equation system exactly. The required reduction is the minimum of a prescribed reduction and the square of the defect reduction in the last iteration (based on the idea that Newton's method has a quadratic convergence). A further reduction of the computational burden is obtained by reassembling the Jacobian only if the convergence rate in the last iteration is below a certain threshold.

Algorithm 3.1 gives the full algorithm for the inexact Newton method in pseudo code.

3.6. Solution of linear Equations

In each step of Newton's method it is necessary to solve a large sparse linear equation system:

$$\mathbf{A}\boldsymbol{x} = \boldsymbol{b} \tag{3.53}$$

where $\mathbf{A}$ is the $N \times N$ Jacobian matrix with only up to $2d + 1$ non-zero entries per line. With $\mathbf{A}$ originating from a cell-centred Finite-Volume discretisation of Richards' equation with upwinding, it is generally not symmetric. Especially for the elliptic case, where the matrix is only weakly diagonally dominant, the condition of the matrix gets worse with the number of grid cells. In the parabolic case the matrix is strictly diagonally dominant, but the condition of the matrix can still be bad due to highly varying coefficients. Thus a robust linear solver is required. Because of the fill-in and the corresponding memory and computing time requirements and as their scalable implementation on large scale supercomputers is difficult, direct linear solvers are not an option.

A class of iterative linear solvers which can be implemented in a scalable and memory-efficient fashion are preconditioned Krylov-Space methods like conjugate gradients. As conjugate gradient solvers are limited to symmetric matrices they are not applicable for our problem. BiCGstab (Vorst 1992) is an established variation of the conjugate gradient scheme for non-symmetric matrices. However, the convergence rate of Krylov-Space methods depends on the distribution of the eigenvalues of the matrix and thus an appropriate preconditioner is needed.

Algorithm 3.1: Inexact Newton with line-search (after Bastian 1999)

```
function NEWTON($\boldsymbol{U}_0, \epsilon_{\mathsf{nl}}, \epsilon_{\mathsf{reass}}, \epsilon_{\mathsf{linmin}}, \mathsf{maxit}, \mathsf{maxline}$)
    $\boldsymbol{U} \leftarrow \boldsymbol{U}_0$                                    ▷ initialise variables
    $m \leftarrow 0$
    $d_{\mathsf{old}} \leftarrow d \leftarrow d_0 \leftarrow ||\boldsymbol{f}(\boldsymbol{U})||_2$
    while $(d > \epsilon_{\mathsf{nl}} \cdot d_0) \wedge (m < \mathsf{maxit})$ do
        $\epsilon_{\mathsf{lin}} = \min\left(\epsilon_{\mathsf{linmin}}, ({}^{d}/_{d_{old}})^2\right)$
        if $(m = 0) \vee ({}^{d}/_{d_{\mathsf{old}}} < \epsilon_{\mathsf{reass}})$ then
            assemble $\mathbf{J} = \boldsymbol{\mathcal{J}}(\boldsymbol{U})$      ▷ assemble matrix (by numerical differentiation)
        end if
        solve $d + \mathbf{J} \cdot \boldsymbol{v} = 0$ to precision $\epsilon_{\mathsf{lin}} \cdot d$     ▷ solve linear equation system
        $d_{\mathsf{new}} \leftarrow d$                                   ▷ initialise variables
        $\lambda \leftarrow 1$
        $\alpha \leftarrow 1$
        $k \leftarrow 0$
        loop                                                 ▷ line search
            $d_{new} = ||\boldsymbol{f}\,(\boldsymbol{U} + \alpha \boldsymbol{v})\,||_2$
            if $d_{\mathsf{new}} > \alpha \cdot d$ then
                if $k = \mathsf{maxline}$ then
                    throw exception
                else
                    $\lambda \leftarrow {}^{1}/_{2} \cdot \lambda$                   ▷ damping strategy after Braess 1992
                    $\alpha \leftarrow 1 - {}^{1}/_{4} \cdot \lambda$
                    $k \leftarrow k + 1$
                end if
            end if
        end loop
        $\boldsymbol{U} \leftarrow \boldsymbol{U} + \alpha \cdot \boldsymbol{v}$                             ▷ update variables
        $d_{\mathsf{old}} \leftarrow d$
        $d \leftarrow d_{\mathsf{new}}$
        $m \leftarrow m + 1$
    end while
    if $m = \mathsf{maxit}$ then
        throw exception
    else
        return $U$                        ▷ algorithm converged, return result
    end if
end function
```

Another class of efficient iterative linear solvers suitable for parallel computing are multigrid schemes. They operate on a hierarchy of coarser and finer discretisations of the problem and thus can eliminate low frequencies error components efficiently on the coarser grids and high frequency error components on the finer grids. Geometric multigrid schemes have been shown to yield optimal convergence rates for elliptic problems. They are memory-efficient and have been shown to scale in a matrix-free implementation up to 10^{12} unknowns and $294'912$ processes (Gmeiner et al. 2012). However, for an optimal convergence it is necessary to resolve the heterogeneity of the system on the coarse grid (or to have very good upscaling techniques).

As it is the intention of this work to develop a program for strongly heterogeneous porous media, algebraic multigrid (AMG) schemes are much more attractive. Algebraic multigrid schemes construct a hierarchy of coarser matrices from an analysis of the matrix resulting from a fine grid discretisation using heuristic strategies. A variety of AMG implementations exist. They differ for example in the work spent to construct on the coarsening, in the iteration schemes (e.g. V-cycle, W-cycle) and in the resulting change of the number of iterations with the number of unknowns.

In this work we use the algebraic multigrid solver from the Iterative Solver Template Library (DUNE-ISTL) of the software framework DUNE (Distributed and Unified Numerics Environment, Bastian et al. 2008a; Bastian et al. 2008b). ISTL provides a matrix and vector interface for a flexible storage of sparse matrices and several solvers and preconditioners (Blatt and Bastian 2007). ISTL also includes parallel solvers and tools (e.g. parallel index sets) for dealing with parallel data from a domain decomposition (Blatt and Bastian 2008).

The ISTL-AMG is meant to be used as a preconditioner. It is based on ideas of Braess 1995. A sequential version was already used in Bastian 1999. The parallel implementation was done by Markus Blatt. The ISTL-AMG is of agglomeration type. A detailed description is given in Blatt et al. 2013. Here we will give only an outline of the algorithm.

Two unknowns of the fine level matrix $\mathbf{A}$ can be merged into an aggregate, if they are strongly connected. The strength of a connection is measured by

$$\frac{w(i,j)\, w(j,i)}{a_{ii}\, a_{jj}} \tag{3.54}$$

where the weights are defined as

$$w(i,j) = \begin{cases} 0 \text{ if } a_{ij} \geq 0 \\ |a_{ij}| \text{ if } a_{ij} < 0 \end{cases} . \tag{3.55}$$

Two unknowns are strongly connected if

$$\frac{w(i,j)\, w(j,i)}{a_{ii}\, a_{jj}} > \delta \, \min(\eta_i, \eta_j) \tag{3.56}$$

where the criterion is defined as a given threshold $0 < \delta < 1$ times the maximal strength of connection to other unknowns

$$\eta_i = \max_k \frac{w(i,k)\, w(k,i)}{a_{ii}\, a_{kk}}. \tag{3.57}$$

Nodes with η_i below a certain threshold are considered as isolated and are not aggregated. The advantage of this definition is, that there is no aggregation over strong discontinuities in the coefficients of the PDE.

The strongly connected unknowns are clustered with a greedy algorithm, which also tries to build round aggregates and merge small clusters to preserve the sparsity pattern also on the coarser level and thus minimise the fill-in. The coarsening is repeated recursively with the coarser matrix until the number of unknowns is small enough or the coarsening rate falls below a certain threshold.

In the solution process the AMG is used as a preconditioner for a BiCGstab-solver. A V-cycle is performed on the multigrid hierarchy with piecewise constant restriction and prolongation between the grid levels. The coarse grid correction is scaled by a factor $\omega \approx 2$. Symmetric Gauss-Seidel is used as smoother, and the coarse grid problem is solved directly with SuperLU (Demmel et al. 1999; Li et al. 1999). This is crucial for the convergence with matrices from discretisation of elliptic PDE.

The combination of AMG and BiCGstab is expected to have a $\mathcal{O}(N \log(N))$ complexity. Napov and Notay 2012 have presented a theory with which they can guarantee the optimal convergence for a similar agglomeration type AMG. However, they use a different multigrid cycle (AMLI-γ-cycle with $\gamma > 1$) which is more costly. In practice the ISTL AMG has been found to be quite robust for elliptic problems with heterogeneous coefficients.

The coarsening and solution phases of ISTL are easy to run in parallel. Communication is combined into as little messages as possible. Even if there is only one unknown per processor left, the problem size P can still be considerable on the largest supercomputers (e.g. up to 1.8 million on the BlueGene/Q type JUQUEEN). A direct solution is not possible on such large systems and is not efficient in parallel anyhow, an iterative solution may converge too slowly. If the number of unknowns per process is below a certain threshold, the ISTL AMG therefore performs a merge to fewer processors. Three different approaches have been tested:

- accumulate data from all processes to one process at once.
- recursive repartitioning to a smaller number of processes using ParMetis to calculate a suitable partitioning on the full connectivity graph.
- recursive repartitioning to a smaller number of processes. To calculate the partitioning accumulate a reduced connectivity graph on one process and use Metis.

The first approach obviously only works for a small number of processes. In the second approach ParMetis tried to allocate a $P \times P$ matrix, which for large process numbers is not feasible. The third approach works, but obviously introduces a P to one communication, which is a bottleneck.

In a comparison of different algebraic multigrid schemes Mueller and Scheichl 2013 found the ISTL AMG to perform very well.

3.7. Time Step Control

The reasonable time step size can change considerably during a simulation based on Richards' equation. If the boundary conditions change, high non-linearity can require very small time steps. After a while, as a consequence of the smoothing property of parabolic partial differential equations, there is hardly any change during one time step, and an increase is sensible. To optimise the balance between computational speed and precision an automatic time step control is necessary. Different approaches between purely heuristic methods and methods based on accurate estimations of the time discretisation error are possible.

3.7.1. Heuristic Approach

In the simple heuristic approach a time step is reduced, if the Newton algorithm for the solution of the non-linear systems or the solver for the linear systems in a Newton step do not converge. A time step is increased if the defect reduction in the first Newton step is below a certain threshold or if the number of Newton iterations is below a threshold. This approach is very simple to implement, but not linked to any criteria regarding numerical precision.

3.7.2. Indicator-based Approach

A time step control should in the optimal case be based on a-posteriori error estimators. One possibility is the use of embedded methods of different order (e.g. embedded Runge-Kutta schemes) to estimate the step width (if the solution is regular enough) or with methods with different step width (extrapolation schemes).

To avoid the overhead of these approaches simple error indicators can be used. For the Richards' equation one possibility is to take the maximal change of the saturation ΔS_w in the domain as an indicator for the time step size. If the maximal saturation change is below a factor β_0 times the threshold $\Delta S_{w_{\mathsf{max}}}$ it is increased in the next time step. If it is above the threshold $\Delta S_{w_{\mathsf{max}}}$, the time step is recomputed with a step width $\beta_1 \cdot {}^{\Delta S_w}/_{\Delta S_{w_{\mathsf{max}}}}$ $(0 < \beta_0 \leq \beta_1 \leq 1)$.

3.8. Hysteresis

If hysteresis is modelled using the modified dependent domain model of Mualem (section 2.3.3.1), the saturation at the new time step $\boldsymbol{S}_w^{k+1}$ depends on the direction of the saturation change, which is given by the sign of $\boldsymbol{\psi}_m^{k+1} - \boldsymbol{\psi}_m^k$ and on the stored last position $(\boldsymbol{\psi}_{m_{\mathsf{rev}}}, \boldsymbol{S}_{w_{\mathsf{rev}}})$ where a direction of saturation change was reversed (from imbibition to drainage or vice versa) in each cells. As a change of the hysteresis branch corresponds to a non-differentiable switch, problems may occur, if these changes are integrated into the Newton iteration itself. Thus the test for a possible reversal of the direction of saturation change is performed after convergence of the Newton scheme. If a reversal occurred in a set of cells j, the position $(\psi_{m_{\mathsf{rev},j}}, S_{w_{\mathsf{rev},j}})$ is set to $(\psi^k_{m_j}, S_w(\psi^k_{m_j}))$ for all affected cells j and the time step is recalculated. This is iterated if necessary. The iteration converges usually very fast.

3.9. Dynamic Processes

The model of Ross and Smettem for dynamic processes (section 2.2.5.5) decouples saturation and potential. Thus it is necessary to store the saturations at the last time step $\boldsymbol{S}_{w_{\mathsf{old}}}$ and to solve an ordinary differential equation

$$\frac{\partial S_w}{\partial t} = \frac{1}{\tau}(S_{w_{\mathsf{eq}}}(\psi_m) - S_w) \tag{3.58}$$

for each cell to calculate the new saturation. Following the suggestion of Ross and Smettem 2000 the new saturation is calculated as

$$S^{k+1}_{w_i} = S^k_{w_i} + \left(S_{w_{\mathsf{eq}}}(\psi_{m_i}) - S^k_{w_{\mathsf{old}},i}\right) e^{-\Delta t/\tau}. \tag{3.59}$$

After the time step is accepted, $\boldsymbol{S}_{w_{\mathsf{old}}}$ is overwritten with the new saturation $\boldsymbol{S}^{k+1}_w$.

3.10. Interpolation of the Flow Field

For the simulation of solute transport with a finer grid resolution or with particle tracking it is necessary to provide a flux field at every point of the domain. However, the cell-centred Finite-Volume scheme only yields the normal fluxes at every face. Raviart-Thomas elements of order zero (RT_0) are used to interpolate the flux field. The RT_0-space is $H(\mathrm{div})$ conforming, i.e. if the normal fluxes for the element are divergence-free, the interpolated flux field is also divergence-free at every point in space. The RT_0 space is defined as (Raviart and Thomas 1977)

$$\mathrm{RT}_0 = \left\{\boldsymbol{v} \in [L_2(\Omega)]^d : \boldsymbol{v}_{|e} \in \mathrm{RT}_0(e) \;\; \forall e \in \mathcal{E}_h\right\} \tag{3.60}$$

composed of piecewise linear polynomials for each spatial dimension with

$$\mathrm{RT}_0(e) = \{a_i + b_i \cdot x_i, \quad 0 \le i < d, \quad \boldsymbol{x} \in e\}\,. \tag{3.61}$$

The normal fluxes for each face give $2d$ conditions for the $2d$ values of the coefficient vectors $\boldsymbol{a}$ and $\boldsymbol{b}$ can be easily calculated by solving a simple linear equation system for each element. The flux field is continuous on each face in the normal direction (though there can be jumps in the transversal components), which is regular enough for most solute transport models. Even at faces were the conductivity in one element is zero, a sensible solution is obtained.

4. Implementation of an Efficient and Scalable Solver for Richards' Equation

The numerical approach described in the last chapter was implemented in the simulation code $\mu\varphi$ (muPhi). C++ is used as programming language, as it combines modern programming techniques with very good compiler optimisation. A sequential version of $\mu\varphi$ was developed first, which was later parallelised (section 4.4). The code is keeping rather strictly to the standard and thus runs on a variety of platforms. It has been tested on Android, MacOS X, Windows (using Cygwin) and on Linux-systems, from ordinary Desktop computers up to supercomputers like BlueGene/P and BlueGene/Q. It compiles on iOS, but as there is no possibility to start command line programs on iOS, a graphical user interface needs to be developed before it can be used.

4.1. General Software Design

$\mu\varphi$ is more than a single simulation program. It consists of two parts:

- a library containing functionality for discretisation in space and time, l, input and output, grid handling and a framework for parameter management including a default implementation of the parameter handler.
- application programs for specific applications sometimes using specialised parameter classes. However, many tasks can be handled with an already very powerful and flexible default program.

The library defines different types of functionality:

- Discretisation of a PDE in space and time (time control, solver for non-linear equation systems, assembly of the linear equation systems)
- Grid management (domain definition, iterators, element geometry, connectivity information)
- Linear Algebra (efficient solution of the linear equation system, handling of vectors and matrices)
- Parameter Management (input of configuration files, management of hydraulic functions, storage of material distribution and sub-scale heterogeneity)

- Time Control (automatic time-step adjustment, management of points of time which have to be met exactly (for output or boundary condition changes)
- Input/Output (read and write backup files, input of structural parameter fields, output of data in different file formats)

Throughout the library abstract base classes are used to define interfaces, which are then realised in derived classes. This usage of virtual functions allows a choice of parametrisations, grid type, linear solver etc. at run-time. This is important as $\mu\varphi$ was designed as a software for scientific *and* teaching applications and it asks too much from the average user to directly modify the source code.

As the author of this work has previously developed a model for freezing soils based on a vertex-centred Finite-Volume discretisation (Ippisch 2003) and implemented it with the toolbox UG (Bastian et al. 1997), the discretisation and non-linear solver part of $\mu\varphi$ are very much inspired by the numerical procedures part of UG. However, there are important differences. $\mu\varphi$ is implemented in C++ instead of C and thus can exploit object-oriented programming techniques more easily. The linear algebra part is completely different, especially in the parallel version and, most important of all, the discretisation is based on a new efficient structured grid. The idea of having an own scripting language was dropped completely and output in the format of popular visualisation tools is used instead of a proprietary visualisation.

4.2. Grid Implementation

The computational grid is a central part of the program. A grid is understood as a set of elements which fill a given domain. Its implementation is light weight and tailored for cell-centred Finite-Volume schemes. The grid provides information about the total number of elements, about the size of the domain and it also has iterator-like constructs to traverse all elements of the grid. The element plays a central role. It has not only a volume and a size, it also provides information about the number of nodes related to the element (the first node is always the node in the centre of the element and the other nodes are the nodes in the centre of the adjacent elements), the control volume faces, which relate two nodes via an interface, and the boundary faces, where boundary conditions have to be provided. A control volume face has an area and provides the distance between the cell centres adjacent to the interface, the fraction of this interface in each of the neighbouring elements and the element local index of the element. Each node has a globally unique and consecutive index, over which the degrees of freedom are related to the grid. The class structure of the grid implementation is shown in figure 4.1.

A grid stores only very general information about the grid geometry, the total number of elements and the number of elements in each direction of Cartesian space. The geometrical and topological information for every element is calculated on the fly and is updated during the traversal of the grid with an iterator. The current element to which the iterator points is stored inside the grid class. Thus the design is not thread-safe and only one iterator may be used with each grid object simultaneously. The iterator itself

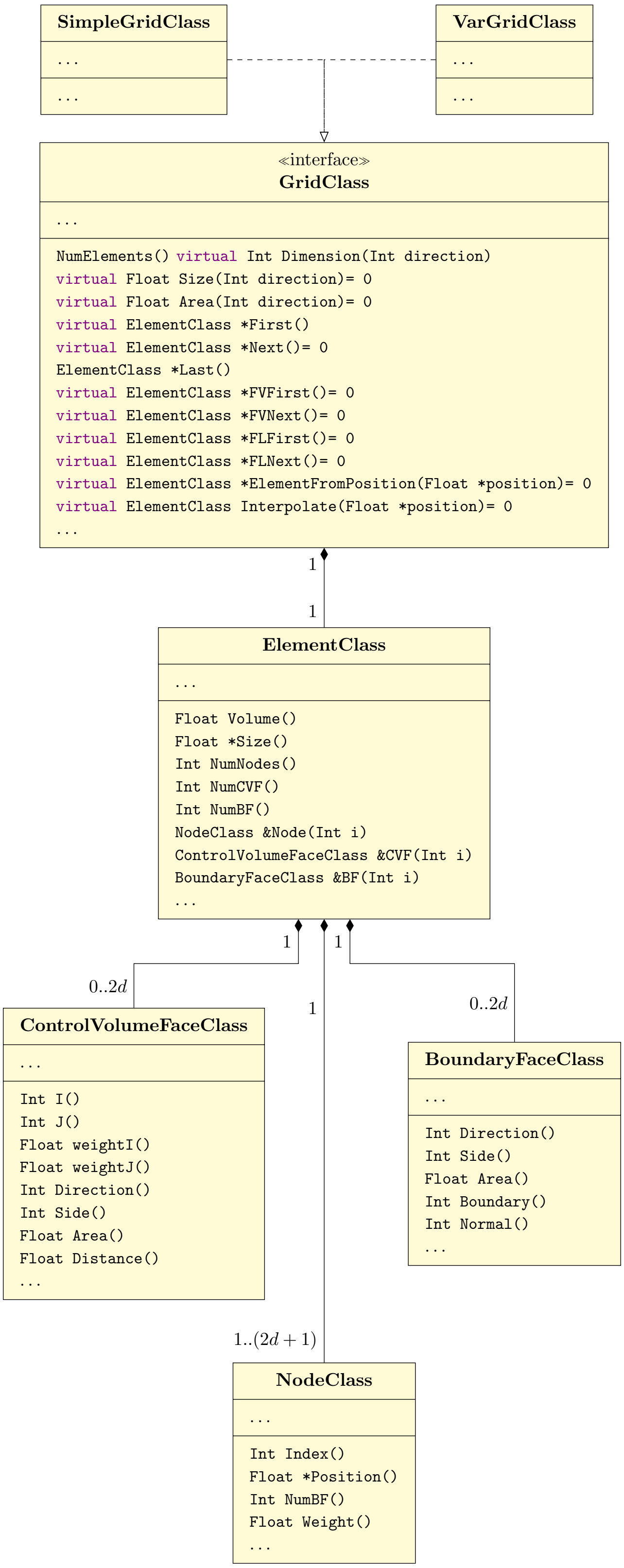

Figure 4.1.: UML-diagram of the classes involved in a grid.

is a pointer returned by the grid method `First` and incremented by `Next`. In the current implementation it just points to the element object held by the grid, but that could be easily changed to obtain multiple (and thread-safe) iterators. To save computation time, the grid provides different iterators which generate different amounts of information. The standard iterator (`First`, `Next`) just set the index and position of the centre node, while the full iterator (`FLFirst`, `FLNext`) sets full geometry and topological information for the element. As the fluxes are continuous at each face, it is sufficient to handle each face exactly once and the grid also provides an iterator (`FVFirst`, `FVNext`), which only handles the left, back and upper side of each element plus all the boundary faces. The handled faces have a consecutive numbering. With methods returning the number of control volume faces (or interior faces) and boundary faces, a simple (and cheap) `for` loop can be used to iterate over the faces and handle the different face types.

Grids which are equidistant in each Cartesian direction are handled by the `SimpleGridClass`, tensor-product grids by the `VarGridClass`. During iterator traversal only the information which changes between the elements is updated and in the simple grid, most of the information remains constant. Thus the traversal of tensor-product grids is more expensive.

4.3. Discretisation Part

Figure 4.2 gives an UML-representation of the discretisation and non-linear solver part of $\mu\varphi$. The inexact Newton scheme with line search (algorithm 3.1) for the solution of non-linear equations is a realisation of a general non-linear solver. During the Newton iterations a linear equation system has to be assembled. This is done by the one-step theta time discretisation (section 3.4), which is implemented as one realisation of a non-linear assembler, and uses itself as a realisation of a linear assembler to calculate the contributions originating from the spatial discretisation for the two steps of the one-step theta scheme. For an implicit Euler scheme this is the storage part for the old time and the storage, flux and source part for the new time (equation 3.22). For a new spatial discretisation or the simulation of other processes only this linear assembler part needs to be modified. Besides the Richards' solver discussed in this work, also a two-phase flow and a non-isothermal two-phase flow realisation of the `TAssembleClass` interface have been implemented. The class also provides methods which are executed at the beginning and at the end of each time step and a method to initialise the matrix and vector storage at simulation start-up.

The components of the Jacobian are calculated by numerical differentiation for the flux, storage and source/sink components of an element and are afterwards accumulated to the global matrix.

In the sequential version of $\mu\varphi$ the vectors and matrices are stored as `AMG_VECTOR` and `AMG_MATRIX` (compressed row storage) data structures defined by the algebraic multigrid implementation of Peter Bastian. The simple direct solver for the coarse grid problem included in the AMG implementation was replaced by a call to SuperLU (Demmel et al. 1999) and a symmetric Gauss-Seidel preconditioner was added to the AMG imple-

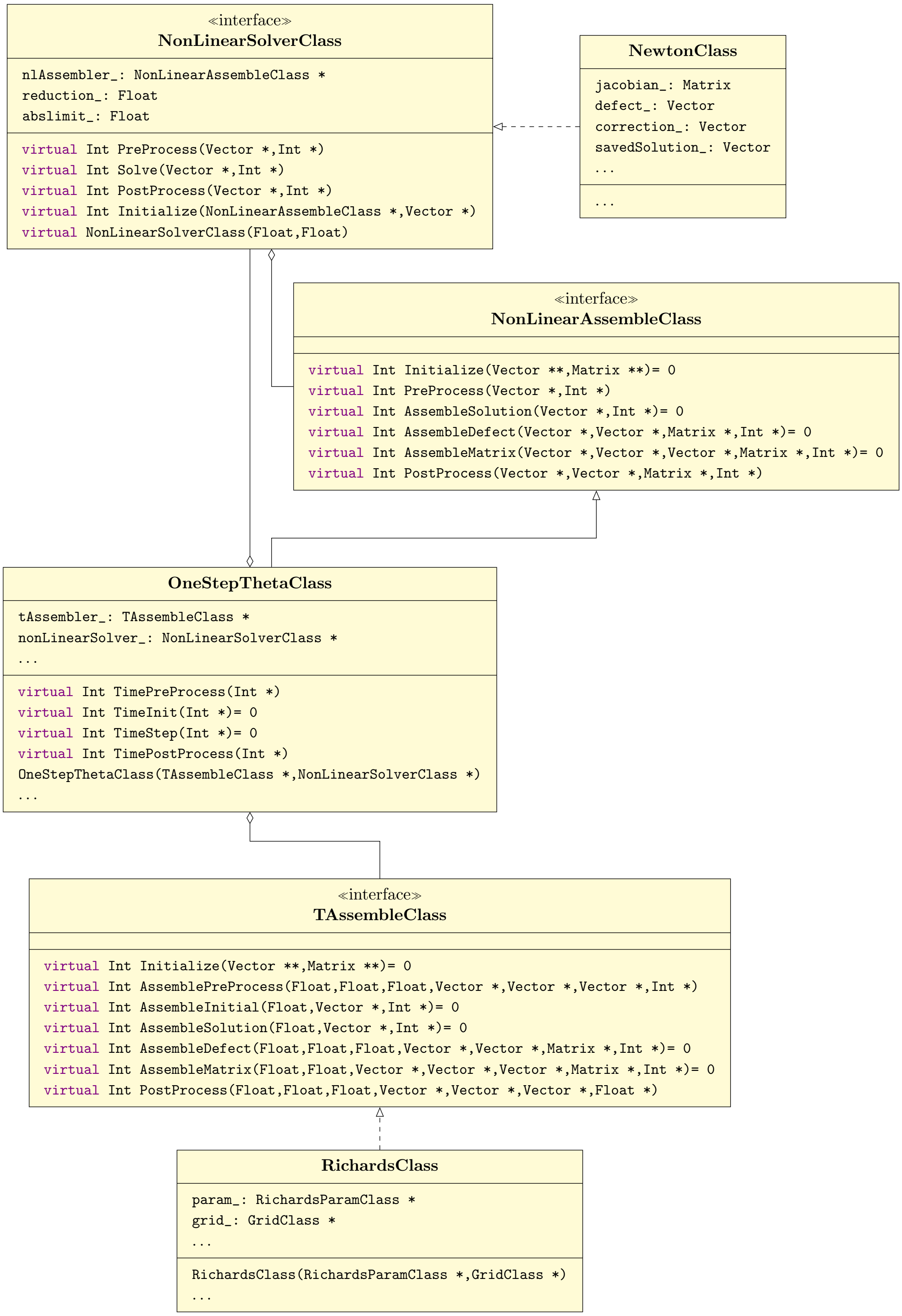
«interface»
NonLinearSolverClass
nlAssembler_: NonLinearAssembleClass *
reduction_: Float
abslimit_: Float
virtual Int PreProcess(Vector *,Int *)
virtual Int Solve(Vector *,Int *)
virtual Int PostProcess(Vector *,Int *)
virtual Int Initialize(NonLinearAssembleClass *,Vector *)
virtual NonLinearSolverClass(Float,Float)
NewtonClass
jacobian_: Matrix
defect_: Vector
correction_: Vector
savedSolution_: Vector
...
...
«interface»
NonLinearAssembleClass
virtual Int Initialize(Vector **,Matrix **)= 0
virtual Int PreProcess(Vector *,Int *)
virtual Int AssembleSolution(Vector *,Int *)= 0
virtual Int AssembleDefect(Vector *,Vector *,Matrix *,Int *)= 0
virtual Int AssembleMatrix(Vector *,Vector *,Vector *,Matrix *,Int *)= 0
virtual Int PostProcess(Vector *,Vector *,Matrix *,Int *)
OneStepThetaClass
tAssembler_: TAssembleClass *
nonLinearSolver_: NonLinearSolverClass *
...
virtual Int TimePreProcess(Int *)
virtual Int TimeInit(Int *)= 0
virtual Int TimeStep(Int *)= 0
virtual Int TimePostProcess(Int *)
OneStepThetaClass(TAssembleClass *,NonLinearSolverClass *)
...
«interface»
TAssembleClass
virtual Int Initialize(Vector **,Matrix **)= 0
virtual Int AssemblePreProcess(Float,Float,Float,Vector *,Vector *,Vector *,Int *)
virtual Int AssembleInitial(Float,Vector *,Int *)= 0
virtual Int AssembleSolution(Float,Vector *,Int *)= 0
virtual Int AssembleDefect(Float,Float,Float,Vector *,Vector *,Matrix *,Int *)= 0
virtual Int AssembleMatrix(Float,Float,Vector *,Vector *,Vector *,Matrix *,Int *)= 0
virtual Int PostProcess(Float,Float,Float,Vector *,Vector *,Vector *,Float *)
RichardsClass
param_: RichardsParamClass *
grid_: GridClass *
...
RichardsClass(RichardsParamClass *,GridClass *)
...

Figure 4.2.: Class structure of the parameter objects of $\mu\varphi$.

mentation. Additionally it is now possible to solve the linear equation system directly with SuperLU, which is more efficient for smaller 2D problems. For 1D problems direct elimination with the Thomas-Algorithm is used for the tridiagonal systems.

4.4. Parallelisation

The consequent usage of an abstract grid interface, of element iterators and function-wrapped access to vectors and matrices made the parallelisation of the code so easy that it could be realised in a software practical by Felix Heimann. The implementation of a parallel grid and the switch to a linear algebra subsystem which can handle distributed vectors and matrices were the two most important things which had to be changed. Of course there was also a need for parallel file I/O (section 4.5) and for some global reduction operations to calculate global sums and global maxima, but these were minor issues. Most of the higher level constructs (non-linear solver, onestep-theta scheme, parameter handling, main program) could remain virtually unchanged.

4.4.1. Parallel Grid

The parallelisation uses a domain decomposition approach. Each process is responsible for a rectangular subdomain. To be able to assemble whole matrix lines, each process needs to store one layer of overlap, which corresponds to all elements with a common face with the elements it owns. A node of the parallel grid thus has, besides the globally unique index, also a local index which is consecutive over all elements of each process, and a partition type, which is either `PT_INTERIOR` if the process owns the element to which the node belongs, `PT_COPY` if the process has a copy in the overlap and `PT_FOREIGN` if the node is not available on the process. The spatial coordinates of the node are always global coordinates. Finally the grid also provides a consecutive index for the interior elements, which is needed in file operations. Information about the global, local and interior domain size and number of elements (total and in each space direction) can also be accessed via the grid interface. The element iterator implementation was adjusted to traverse only the local elements.

4.4.2. Dune-ISTL

For the parallel linear algebra DUNE-ISTL is used in $\mu\varphi$. It is open source, developed in the work group in which also the author works, and provides distributed vectors and matrices as well as powerful iterative linear solvers, in particular a parallel algebraic multigrid preconditioner, which is the successor of the AMG used in the sequential version (section 3.6). ISTL provides different matrix types. For this work a matrix of type `BCRSMatrix<Dune::FieldMatrix<Float,1,1>>` is used to store the Jacobian and vectors of type `Dune::BlockVector<Dune::FieldVector<Float,1>>` for the unknowns and the right-hand side of the linear equation system.

A parallel linear solver needs to exchange information to update the copied values in the overlap and to calculate vector norms and dot products. For this purpose ISTL provides communication objects. An object of type `Dune::OwnerOverlapCopyCommunication<long long,long long>` is used to exchange data between processes with common degrees. The data type `long long` is used for indices to allow for the huge amount of unknowns possible with modern super computers. Access to the communication object is provided by the grid interface so that other parts of $\mu\varphi$ can use the communication infrastructure to exchange data (e.g. concerning structure) for the overlap.

There is also a wrapper for MPI, defined in DUNE-Common, providing standard communications like send, receive, reduce and broadcast. Thus parallel communication can be performed without touching standard MPI functions.

4.5. Parameter Management

The simulation of water flow in heterogeneous soils requires the handling of a number of mostly space and sometimes also time-dependent parameters, like soil water characteristics, relative permeability function, porosity etc. All parameter functions are encapsulated in an object of a parameter class. The methods are defined via an abstract base class, and different parameter functions and distributions can thus be easily implemented and chosen at run-time. As many parameter functions are expensive to evaluate anyhow, this is not a big performance burden. The class structure is given in figure 4.3. A default implementation is provided in the library. New parameter classes can be derived from the default implementation. Thus it is enough to just redefine methods, which need to be changed.

4.5.1. Hydraulic Properties

Heterogeneous materials are realised via two spatially distributed fields. One field stores an index for a field with hydraulic parameter function objects, the other a scaling value for Miller similarity scaling (section 2.3.4). The resolution for each field can be different and does not have to match the grid resolution as scaling value and material index are sampled from the respective field at the element centre. The input of the material and parameter field is described in section 4.6.4. At run-time the material is either accessed via the coordinates (sequential version) or via the element index (parallel version).

The hydraulic functions themselves are objects derived from the base class `HydrParamClass` (figure 4.4). The base class already handles the entry pressure and the calculation of the actual saturation from the effective saturation (section 2.3.1). The derived classes only need to implement virtual functions to calculate effective saturation and relative permeability from matric potential. There exists a variety of implementations (some versions implemented for single research projects are not listed here):

- `VanGenuchtenParamClass`: van Genuchten/Mualem model)
- `ModVanGenuchtenParamClass`: modified van Genuchten/Mualem model

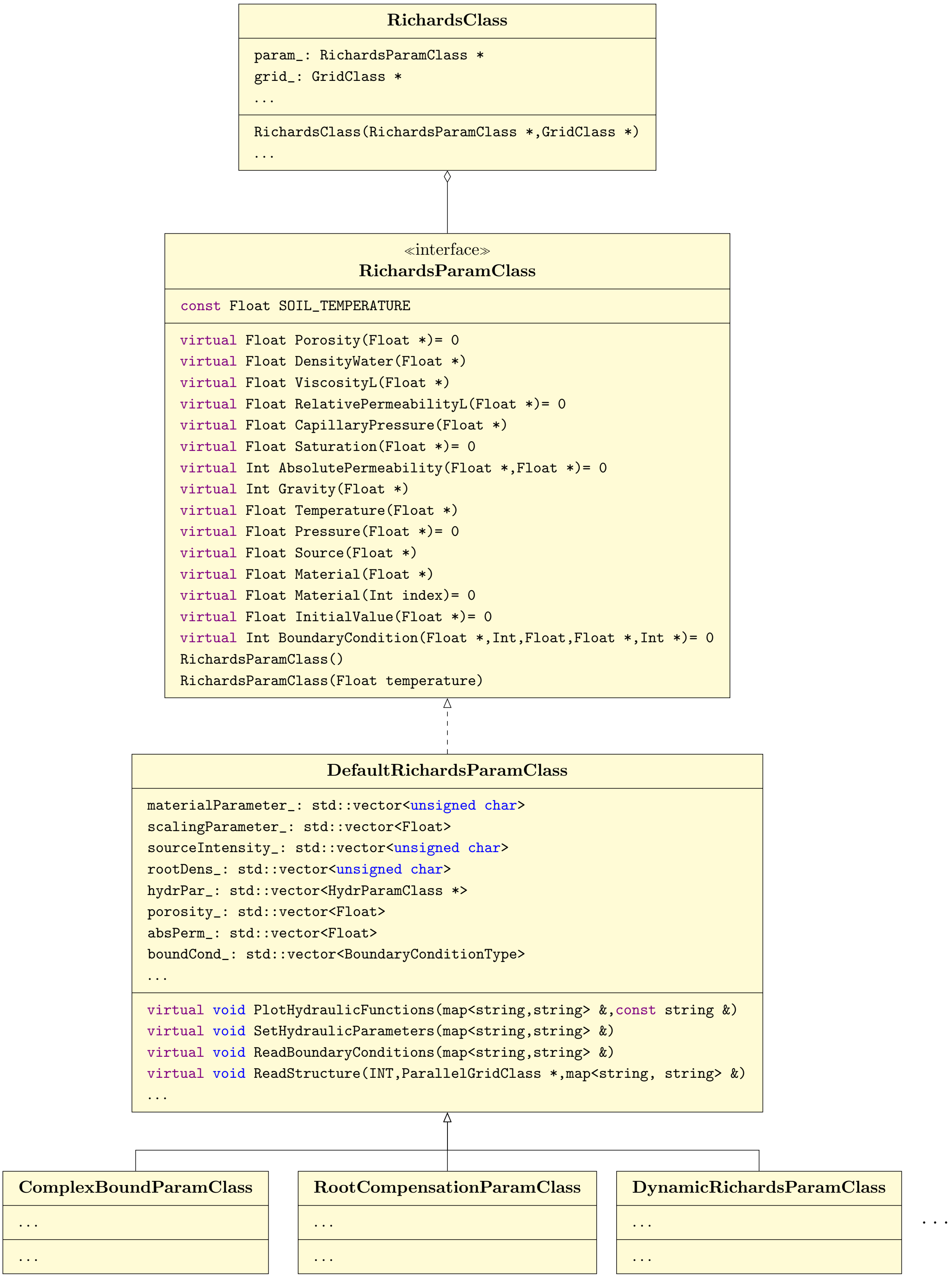
RichardsClass
param_: RichardsParamClass *
grid_: GridClass *
...
RichardsClass(RichardsParamClass *,GridClass *)
...
«interface»
RichardsParamClass
const Float SOIL_TEMPERATURE
virtual Float Porosity(Float *)= 0
virtual Float DensityWater(Float *)
virtual Float ViscosityL(Float *)
virtual Float RelativePermeabilityL(Float *)= 0
virtual Float CapillaryPressure(Float *)
virtual Float Saturation(Float *)= 0
virtual Int AbsolutePermeability(Float *,Float *)= 0
virtual Int Gravity(Float *)
virtual Float Temperature(Float *)
virtual Float Pressure(Float *)= 0
virtual Float Source(Float *)
virtual Float Material(Float *)
virtual Float Material(Int index)= 0
virtual Float InitialValue(Float *)= 0
virtual Int BoundaryCondition(Float *,Int,Float,Float *,Int *)= 0
RichardsParamClass()
RichardsParamClass(Float temperature)
DefaultRichardsParamClass
materialParameter_: std::vector<unsigned char>
scalingParameter_: std::vector<Float>
sourceIntensity_: std::vector<unsigned char>
rootDens_: std::vector<unsigned char>
hydrPar_: std::vector<HydrParamClass *>
porosity_: std::vector<Float>
absPerm_: std::vector<Float>
boundCond_: std::vector<BoundaryConditionType>
...
virtual void PlotHydraulicFunctions(map<string,string> &,const string &)
virtual void SetHydraulicParameters(map<string,string> &)
virtual void ReadBoundaryConditions(map<string,string> &)
virtual void ReadStructure(INT,ParallelGridClass *,map<string, string> &)
...
ComplexBoundParamClass
...
...
RootCompensationParamClass
...
...
DynamicRichardsParamClass
...
...
...

Figure 4.3.: Class structure of the parameter management of $\mu\varphi$.

- `RegModVanGenuchtenParamClass` a modified van Genuchten/Mualem model implementation where the non-differentiable edge is smoothed
- `MultVanGenuchtenParamClass`: multiple van Genuchten/Mualem model
- `ModMultVanGenuchtenParamClass`: modified multiple van Genuchten/Mualem model
- `BrooksCoreyParamClass`: Brooks-Corey/Mualem model
- `RegBrooksCoreyParamClass` Brooks-Corey/Mualem model implementation where the non-differentiable edge is smoothed
- `HydrSplineClass`: cubic Hermite-Splines for both the soil water characteristic and the relative permeability curve

4.5.2. Interpolation Tables

As the evaluation of some of the parametrisation can be quite costly, it makes sense to sacrifice some memory and use interpolation tables. Interpolation is available with cubic Hermite splines and with linear splines. To obtain the same precision of the approximation, linear splines need more memory for the interpolation tables, but are also faster to evaluate. Thus if enough memory is available the use of linear splines is recommended and can improve the total speed of calculation up to a factor of two.

Interpolation is fastest with equidistant intervals, as the coefficient index in the interpolation table can then be calculated directly. The functions which need to be evaluated most frequently are $S_w(\psi_m)$ and $k_r(\psi_m)$. As the curves can be highly non-linear, the choice of an appropriate interval $\Delta\psi_m$ for the setup of the tables is crucial for precision. It is chosen as

$$\Delta\psi = \min_{2 \le i < (\frac{1-\Delta S_w}{\Delta S_w})} \left(\psi_m(i \cdot \Delta S_w) - \psi_m((i-1) \cdot \Delta S_w)\right) \tag{4.1}$$

The limits of the interpolation interval $[\psi_{\mathsf{low}} : \psi_{\mathsf{up}}]$ and the number of interpolation points n are given by

$$\begin{aligned} \psi_{\mathsf{up}} &= \max\left(0, \psi_{m_0}\right) \\ \psi_{\mathsf{low}} &= \max\left(\psi_{\mathsf{min}}, \psi_{\mathsf{up}} - n_{\mathsf{max}} \cdot \Delta\psi\right) \\ n &= \min\left(n_{\mathsf{max}}, \frac{\psi_{\mathsf{up}} - \psi_{\mathsf{low}}}{\Delta\psi}\right), \end{aligned} \tag{4.2}$$

where ψ_{m_0} is the air entry value of the material. The required saturation resolution ΔS_w, the maximal number of interpolation intervals n_{max} and the minimal interpolation potential ψ_{min} are user defined parameters. Below ψ_{low} the hydraulic functions are still evaluated directly. Compared to approaches where the user directly specifies a potential interval $\Delta\psi$, like used for example in Hydrus (Simunek et al. 1997), this provides a more reliable error control. At the moment this is only used for the setup of linear interpolation

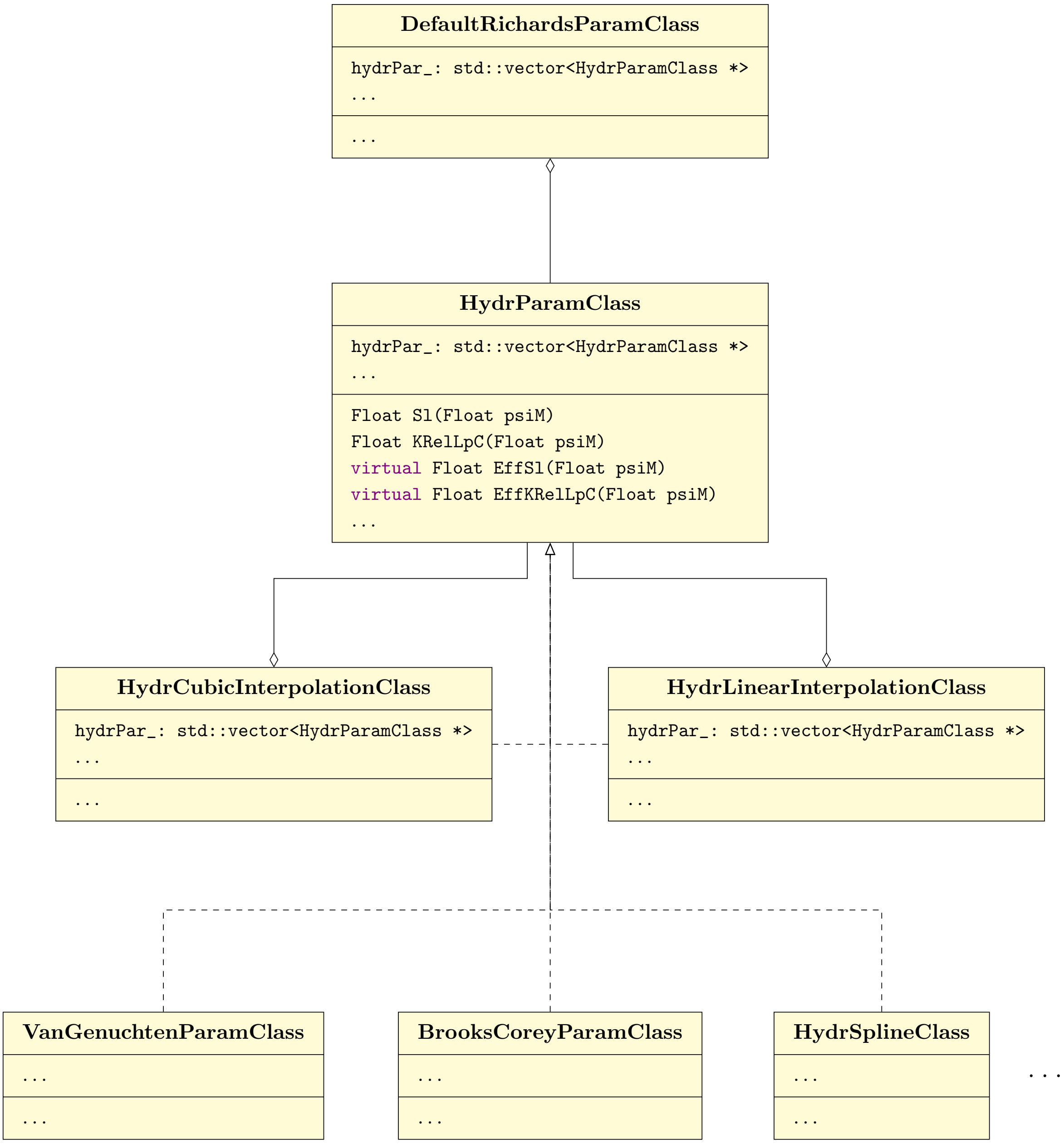
DefaultRichardsParamClass
hydrPar_: std::vector<HydrParamClass *>
...
...
HydrParamClass
hydrPar_: std::vector<HydrParamClass *>
...
Float Sl(Float psiM)
Float KRelLpC(Float psiM)
virtual Float EffSl(Float psiM)
virtual Float EffKRelLpC(Float psiM)
...
HydrCubicInterpolationClass
hydrPar_: std::vector<HydrParamClass *>
...
...
HydrLinearInterpolationClass
hydrPar_: std::vector<HydrParamClass *>
...
...
VanGenuchtenParamClass
...
...
BrooksCoreyParamClass
...
...
HydrSplineClass
...
...
...

Figure 4.4.: Class structure of the hydraulic functions objects.

tables. For cubic interpolation tables the interval $[\psi_{\mathsf{up}} : \psi_{\mathsf{min}}]$ is just divided into $(n_{\mathsf{max}}$ intervals.

To use the interpolation tables it is enough to pass an initialised hydraulic parameter object to the interpolation object. The interpolation table takes care of the hydraulic parameter object as it is still needed below ψ_{low}. Interpolation tables are a realisation of `HydrParamClass` (figure 4.4) and thus can be directly used as hydraulic parameter object.

4.6. File I/O

Files are used to control the program's behaviour, to provide structural information and material parameters and to store results. They also are the main user interface to a scientific simulation code like $\mu\varphi$. Thus they are a critical part of the software design.

4.6.1. Parameter files

A complex numerical model like $\mu\varphi$ has a lot of control parameters like the domain size, the element width (which may be vectorial in case of tensor product grids), convergence limits and limits for the maximal amount of non-linear iterations or time steps, parameters for time step control, parameters describing hydraulic functions and structural patterns, root water uptake etc. The implementation of new functionality always also comes with new parameters. Thus the file format needs to be flexible and easy to comprehend.

While XML-based formats are easy to read by machines, they are less obvious and a bit tedious for human users. For $\mu\varphi$ thus a file format based on a keyword value principle was selected. The first entry in every line is always either a keyword identifying the type of the entry or the hash sign # marking a comment. The rest of the line is a string which is parsed when the entry is used. All key value pairs are stored in a map of type `std::map<std::string,std::string>`. There is a helper function `ReadParameterFile` and a function template `GetValue<T>` to extract parameters of different data types (string, bool, double, int, vectors). This also ensures that a type-safe parsing is performed. The adding of new parameters is trivial, as this only requires the addition of a matching call to `GetValue` to extract the value from the parameter map.

For the storage of boundary conditions and for the element width vectors of tensor product grids there are separate files as both might be used in combination with several parameter files. The latter has a very simple format. In the first line just the d numbers of elements in each space direction is given, in the next d lines follow a matching number of element width. Comment lines are allowed and are again marked by a hash mark. It is checked by the constructor of the tensor-product grid that the domain size resulting from the sum of the element widths is equal to the total domain size. The name of the grid file is specified given in the main input file (of course in the keyword/value format). For parallel computations the content of the main input file, grid file and boundary condition file is just read by the rank zero process and broadcasted to all other processes.

	`sp_units`	`sih_units`	`si_units`
time	h	s	s
potential	cm	m	Pa
length	cm	m	m
flux rates	cm/h	m/s	kg / m^2 s
conductivity	cm/h	m/s	m/s
temperature	K	K	K

Table 4.1.: Different unit systems available in $\mu\varphi$.

4.6.2. Boundary Conditions

There are different types of input formats for boundary conditions dependent on the application. In the ordinary boundary condition file the first line contains the number of boundary conditions and the following lines contain a space separated list of time, lower boundary condition type and values (if necessary) and upper boundary condition type and values. For the parallel version at the moment only this type of input format is supported. At all other boundaries no-flow boundary conditions are used.

For the sequential version and for 2D simulations only there is a more flexible input format, where for all sides of the domain different boundary conditions can be specified for arbitrary sub-intervals. This variety was easily implemented by a parameter class, derived from the default implementation, where the (virtual) method for the input of boundary conditions and the method which is called by the discretisation to obtain the boundary conditions are overloaded.

4.6.3. Unit System

Many parameters have units attached. Unfortunately in petrol engineering, civil engineering, soil physics and hydrology different unit systems are common. While in soil physics matric potential is measured in centimetre, in civil engineering capillary pressure is given in Pascal. Saturated hydraulic conductivity is given in metres per second or centimetres per hour ...

To handle this problem a switchable unit system was introduced. Internally $\mu\varphi$ always uses SI-units. All parameters, boundary condition values and results are multiplied on input or output by conversion factors provided by a global object `UNITS` of class `Units`. The unit system to be used can be chosen by a parameter in the main input file. A different unit system thus can be created by adding a few lines to the `Units` class. Three different unit systems are predefined (table 4.1).

4.6.4. Material Files

For 2D simulations one of the simplest ways to specify a heterogeneous structure is by a grey-scale image, where every grey value corresponds to a different material. In $\mu\varphi$

TIFF-images are used for this purpose, and the corresponding material properties are given in the main parameter file for a stated number of materials (up to 256). For each material a different parametrisation type and the necessary parameters can be given as well as the grey value for which the hydraulic function should be used. For sub-scale heterogeneity, which is represented by Miller similarity, a second image is used and the scaling parameter η is calculated from the grey value k and a given standard deviation σ as

$$\eta = \exp\left(\sigma \cdot \frac{k - 128}{128}\right). \tag{4.3}$$

For 3D simulations the `ddd` file format of the Quantim library (Vogel 2008) is used, which itself is a 3D generalisation of the TIFF format. Quantim is a free library for scientific image processing and specialised on post-processing of porous media data. Thus it is often already known to potential users.

The structures for large-scale parallel simulations can get much larger than the memory of a single compute node, especially with low-memory platforms like the BlueGene/P and /Q. Thus parallel file input is necessary for the material and scaling parameter files. One rather obvious candidate is parallel HDF5. It is a common library, available on many different operating systems and supports the parallel input of so-called hyperslabs. Each process just reads the data block relevant for the part of the domain it owns. The ISTL communication object is then used to exchange the data for the overlap. In the parallel version, the scaling parameter is stored directly as float and not as grey value and converted. Additionally parallel output with the SIONlib (Frings et al. 2009; Frings 2009–2013) has been implemented. A comparison of the obtained transfer rates during input is given in section 12.1.1.5.

4.6.5. Results

Some results produced by the simulation are simple scalar values, like the flux or the cumulative flow over the boundaries. Other results are complete spatial fields of matric potential, volumetric water content or flux density vector and can be very large (up to several Terabyte). The sequential version of $\mu\varphi$ supports output in a format suitable for the Open Visualization Data Explorer of IBM, the vtr-format of ParaView and an own (simple) binary file format. The parallel version supports output based on HDF5 and SIONlib. Two different versions of the HDF5-output have been implemented:

- Output using the hyperslab concept to write a three-dimensional structure, which can be directly used by other programs like ParaView. This requires a reordering of the data from all processes during the write operation. This is suitable for small to mid-size clusters.
- Output into a single flat vector, where each process writes its data as a large contiguous block. Reordering can be performed afterwards on high-memory sequential computers. This is faster on large machines.

Parallel file output can slow down considerably if more than one process tries to access the same file system block at the same time. This problem increases with the number of processes and the size of file system blocks. IBM General Parallel File System (GPFS) for example uses file system blocks of up to 4 Megabyte. The SIONlib library makes sure that whole file system blocks are allocated for each process and thus is faster on large supercomputers. Information about the grid is written into each result file only by the rank zero process. Utilities to convert the output of the flat HDF5 file format and from SIONlib files to ordinary HDF5 or parallel vtr file format are part of the $\mu\varphi$ distribution. A comparison of the obtained transfer rates during output is also given in section 12.1.1.5.

4.6.6. Backup and Restart

The maximal allowed run-time for a single job is usually limited to some hours up to one day on large parallel computers. For problems which require a longer total run-time a backup and restore option is necessary. This is also advantageous as the probability of a system breakdown increases with the number of cores. $\mu\varphi$ can read matric potential or saturation output files and use them as initial condition for continuing the calculation.

4.7. Documentation

Good documentation is crucial for the usability and maintainability of software. The best documentation is a good and logical software design and the use of speaking variable names throughout the code. Comments are added were necessary. A user guide (Ippisch 2013) is documenting all input files and the keys and values for the program (concentrating on the sequential version).

4.8. Code Verification

Testing is important but increasingly difficult for complex simulation software.

4.8.1. Global Mass Balance

As the cell-centred Finite-Volume scheme is inherently locally mass conservative, it should also be globally mass conservative. This can be exploited as an indicator for the quality of the solution. The total mass is calculated by

$$M_{\mathsf{tot}}(t^k) = \sum_{e_i \in \mathcal{E}} \mathcal{M}_i(U^k), \tag{4.4}$$

the total cumulative source by

$$Q_{\mathsf{cum}}(t^k) = \sum_{0<l<k} \left[\Delta t_l \cdot \sum_{e_i \in \mathcal{E}} \mathcal{Q}_i(U^k) \right], \tag{4.5}$$

material	θ_r [–]	θ_s [–]	α [cm^{-1}]	n [–]	K_s [cm/h]	τ [–]
Sand	0.045	0.43	0.15	3.0	41.67	0.5
Loam	0.08	0.43	0.04	1.6	2.083	0.5

Table 4.2.: Material parameters used for the comparison with the analytical solutions (from Vanderborght et al. 2005).

and the total cumulative flow over the boundary of the domain by

$$A_{\mathsf{cum}}(t^k) = \sum_{0<l<k} \left[\Delta t_l \cdot \sum_{f_{ij} \in \partial \otimes} \mathcal{A}_{ij}(U^k) \right] \tag{4.6}$$

The global mass balance error can then be calculated as

$$\Delta M(t^k) = M_{\mathsf{tot}}(t^k) + A_{\mathsf{cum}}(t^k) + Q_{\mathsf{cum}}(t^k) - M_{\mathsf{tot}}(t^0) \tag{4.7}$$

or for the last time step

$$\Delta M_{\mathsf{step}}(t^k) = M_{\mathsf{tot}}(t^k) + A_{\mathsf{cum}}(t^k) + Q_{\mathsf{cum}}(t^k) - M_{\mathsf{tot}}(t^{k-1}). \tag{4.8}$$

The acceptable mass balance error increases with the number of elements and the size of the domain. However, if the linear and non-linear equation systems are solved very precisely, the relative error compared to the largest of the components of the global mass balance equation should be close enough to machine precision.

4.8.2. Comparison with Analytical Solutions

Problems for which analytical solutions exist can be used to test the validity of a simulation program. A set of test cases with analytical solutions was published by Vanderborght et al. 2005 and applied to compare different existing solvers for Richards' equation and the Convection-Dispersion equation. Two of the test cases have been used to test $\mu\varphi$. A special class to calculate the two analytical solutions by numerical integration for arbitrary hydraulic properties was written. For an easy comparison with the results published for other models in the paper of Vanderborght et al. 2005 the same parameters (table 4.2), initial and boundary conditions have been used. The clay was not included in the simulations as it has a van Genuchten n of 1.1, which produces non-physically steep relative permeability functions (section 2.3.2.3).

The first test case is based on a Kirchhoff transform of the Buckingham-Darcy equation. For steady-state condition the equation can be integrated over a range of matric potentials, resulting in the spatial distance between the matric potentials used as upper and lower integration boundaries:

$$\Delta x = x_1 - x_0 = \int_{\psi_1}^{\psi_0} \frac{1}{\left(\frac{j_l}{K(\psi)} - \rho_w g \right)} d\psi. \tag{4.9}$$

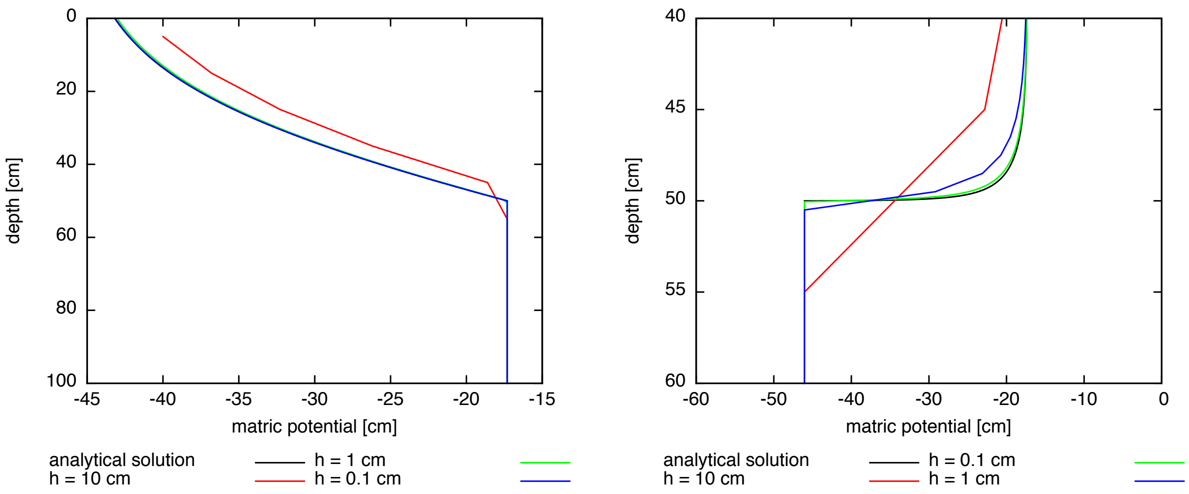

Figure 4.5.: Analytical solution for the matric potential profile during steady-state flux into a layered soil and values calculated with $\mu\varphi$ for different discretisation width with loam above sand (left) and sand above loam (right).

This can be used to calculate profiles of matric potential in layered-soil.

Simulations with an element size of 0.1, 1 and 10 centimetres have been run to study grid convergence with a flux rate of 0.02833 cm/h. For both cases the numerical solutions converge to the analytical solution (figure 4.5). The deviation for an element size of 1 cm is comparable to the models tested by Vanderborght et al. 2005.

A semi-analytical solution of Richards' equation for infiltration in homogeneous soil has been published by Philip 1969 based on a travelling wave approximation and also was used by Vanderborght et al. 2005. In an infinite domain the initial and lower boundary condition is given by a constant water content $\theta(x,0) = \theta_i$ and the upper boundary condition is given by an (also constant) water content $\theta(0,t) = \theta_{\mathsf{sur}}$. A transformed coordinate η is introduced by

$$\eta = x - \frac{K(\theta_{\mathsf{sur}}) - K(\theta_i)}{\theta_{\mathsf{sur}} - \theta_i}. \tag{4.10}$$

If the derivative of θ to η is assumed to be zero at both ±∞ the following solution can be obtained by integration:

$$\begin{aligned} \varDelta\eta &= \eta(\theta) - \eta(\theta_a) \\ &= \int\limits_{\theta}^{\theta_a} \frac{[\theta_{\mathsf{sur}} - \theta_i]\, D_w(\theta)}{[K(\theta_{\mathsf{sur}}) - K(\theta_i)]\,[\theta - \theta_i] - [K(\theta) - K(\theta_i)]\,[\theta_{\mathsf{sur}} - \theta_i]} d\theta \end{aligned}$$

where θ_a is a reference water content. According to this solution, the shape of water content profiles at the infiltration front should not change over time, only their position. If the profiles are plotted relative to a reference water content θ_a all profiles should lie on top of each other and match the analytical solution.

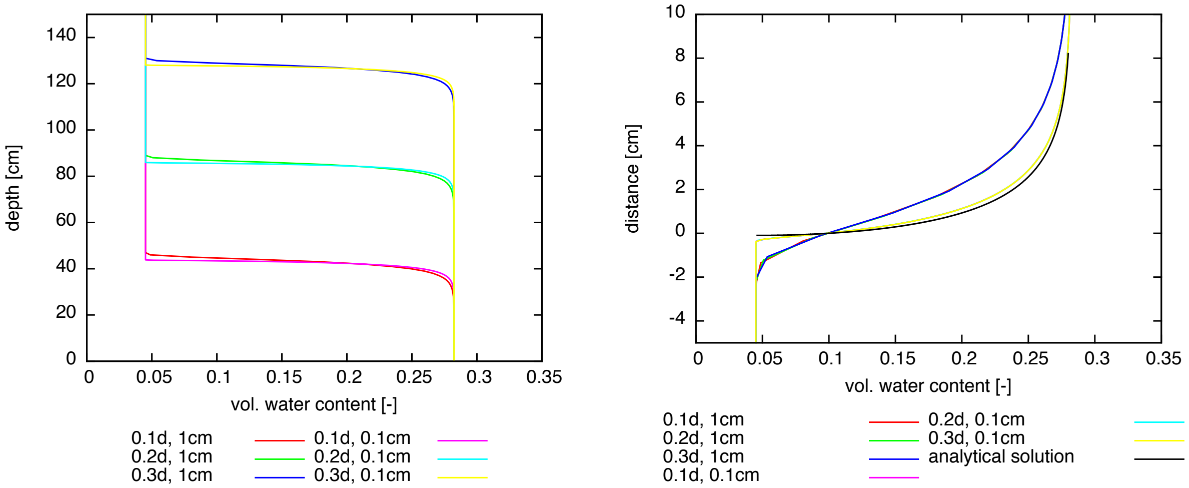

Figure 4.6.: Profiles of volumetric water content during infiltration in a homogeneous sand with a constant flux rate after 2.4, 4.8 and 7.2 hours for two different element width (left). Water content profiles plotted relative to the position where the volumetric water content is equal to 0.1 (right) together with the semi-analytical solution of Philip.

A flux of 4.167 cm/h was applied to a dry sand with an initial matric potential of -400 cm. To approximate a semi-infinite profile, gravity flow was used as lower boundary condition. Profiles after 2.4, 4.8 and 7.2 h calculated with a discretisation width of 0.1 and 1 cm are shown on the left of figure 4.6. A mapping to a reference water content of 0.1 shows a quick convergence to the semi-analytical solution (figure 4.6, right). All profiles of a single simulation with given element width overlap perfectly. The results are in very good agreement with the results published in Vanderborght et al. 2005.

5. Parameter Estimation

The hydraulic functions of porous media are often hard to determine from direct measurements. Parameter estimation is a very useful tool to obtain information about the hydraulic functions fast and reliably from transient measurements. The experiment is simulated with a model of the system, and the distance between simulated and measured data is measured by a so-called objective function. The parameters of the model are then chosen to minimise the objective function. If the model is a non-linear function of the parameters this is not a trivial problem. So-called global optimisation algorithms (Vrugt et al. 2004) try to sample the whole parameter space in an intelligent way. Other algorithms make assumptions over the shape of the hyperplane formed by the objective function and use approximations to make a quick descent (Bock 1987). While the latter have a tendency to end up in global minima, especially for ill-conditioned problems, the former have a complexity which is exponential in the number of parameters to be optimised. This is not feasible if the model is a highly resolved discretisation of a non-linear partial differential equation. In this study we will thus use iterative algorithms which use approximations of the Hessian of the objective function. The two most prominent of these are the Gauss-Newton and the Levenberg-Marquardt algorithm.

5.1. Objective Function

The most common choice for the objective function $F(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\beta})$ is:

$$F(\boldsymbol{x}, \boldsymbol{y}) = \frac{1}{2} \left(\boldsymbol{y} - \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{\beta})\right)^T \mathbf{C_{YY}} \left(\boldsymbol{y} - \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{\beta})\right), \tag{5.1}$$

where $\boldsymbol{x}$ is the vector of simulated values, $\boldsymbol{y}$ is the vector of measurements, $\boldsymbol{\beta}$ is the vector of parameters, $\boldsymbol{f}$ the vector of measurement functions and $\mathbf{C_{YY}}$ is the covariance matrix of the measurements. In our application the evaluation of $\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{\beta})$ requires one run of a forward model, i.e. one transient water transport simulation. For uncorrelated measurements C_{YY} is given by

$$C_{YY,ij} = \begin{cases} \sigma_i^{-2} & \text{if } i = j \\ 0 & \text{else} \end{cases} \tag{5.2}$$

where σ_i^2 is the variance of measurement i. The objective function then becomes

$$F(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\beta}) = \frac{1}{2} \sum_{i=0}^{N} \left(\frac{y_i - f_i(x_i, \boldsymbol{\beta})}{\sigma_i} \right)^2, \tag{5.3}$$

the squared sum of weighted residuals.

5.2. Iterative Methods for Non-linear Least-Squares Problems

At the minimum of the objective function with respect to the parameters we have

$$\nabla_\beta F(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\beta}) = 0. \tag{5.4}$$

This is a system of non-linear equations with

$$\frac{\partial F(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\beta})}{\partial \beta_m} = \sum_{i=0}^{N} \frac{(y_i - f_i(x_i, \boldsymbol{\beta}))}{\sigma_i^2} \frac{\partial f_i(x_i, \boldsymbol{\beta})}{\partial \beta_m}. \tag{5.5}$$

To find the root with Newton's method we have to solve in each step the linear equation system

$$\mathcal{H}^k \mathit{\Delta} \boldsymbol{\beta}^k = \nabla_\beta F(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\beta}^k), \tag{5.6}$$

where $\mathcal{H}$ is the Hessian of the objective function with

$$\mathcal{H}_{lm}^k = \frac{\partial^2 F(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\beta}^k)}{\partial \beta_l \partial \beta_m} = \frac{\partial}{\partial \beta_l} \left[\sum_{i=0}^{N} \frac{\left(y_i - f_i(x_i, \boldsymbol{\beta}^k)\right)}{\sigma_i^2} \frac{\partial f_i(x_i, \boldsymbol{\beta}^k)}{\partial \beta_m} \right] \tag{5.7}$$

$$= \sum_{i=0}^{N} -\frac{1}{\sigma_i^2} \frac{\partial f_i(x_i, \boldsymbol{\beta}^k)}{\partial \beta_l} \frac{\partial f_i(x_i, \boldsymbol{\beta}^k)}{\partial \beta_m} + \frac{\left(y_i - f_i(x_i, \boldsymbol{\beta}^k)\right)}{\sigma_i^2} \frac{\partial^2 f_i(x_i, \boldsymbol{\beta}^k)}{\partial \beta_l \partial \beta_m}. \tag{5.8}$$

5.2.1. Gauss-Newton Algorithm

In the Gauss-Newton scheme the Hessian is approximated by dropping the last term in equation 5.8. Thus in each iteration step the linear equation system

$$\mathbf{A}^k \boldsymbol{\nu}^k = \boldsymbol{b}^k \tag{5.9}$$

is solved with

$$A_{lm}^k = -\sum_{i=0}^{N} \frac{1}{\sigma_i^2} \frac{\partial f_i(x_i, \boldsymbol{\beta}^k)}{\partial \beta_l} \frac{\partial f_i(x_i, \boldsymbol{\beta}^k)}{\partial \beta_m}$$

$$b_m^k = \sum_{i=0}^{N} \frac{\left(y_i - f_i(x_i, \boldsymbol{\beta}^k)\right)}{\sigma_i^2} \frac{\partial f_i(x_i, \boldsymbol{\beta}^k)}{\partial \beta_m}.$$

To increase the region of convergence, a line search is used with

$$\boldsymbol{\beta}^{k+1} = \boldsymbol{\beta}^k + \lambda_i \boldsymbol{\nu}^k, \quad \lambda_i \in]0:1] \tag{5.10}$$

with

$$\lambda_i = 2^{-i}, \quad i = \{0, ..., \mathsf{max}\} \tag{5.11}$$

until a new set of parameters is found for which the objective function is sufficiently reduced. For each line search step, one run of the forward model is necessary. Close enough to the solution of the minimisation problem, the Gauss-Newton algorithm should converge quadratically.

5.2.2. Levenberg-Marquardt Algorithm

The Gauss-Newton algorithm converges well, if a quadratic function is a good approximation of the objective function. A direction in which the function is always decreasing is given by the gradient. The Levenberg-Marquardt (Marquardt 1963) algorithm therefore is an alternative to using a linear search. If the convergence is very good, the steps are in the Gauss-Newton direction and quadratic convergence is achieved. If the convergence is bad, the algorithm moves the search direction toward steepest descent and at the same time reduces the step size. This is achieved by using a modified matrix $\mathbf{A}'$ with

$$A'_{lm} = (1 + \delta_{lm}\lambda)A_{lm}$$

where δ_{lm} is the Kronecker delta. Thus only the diagonal element is modified. If λ is very large, the off-diagonal entries can be ignored and the correction will be in the direction of the gradient and scaled by λ^{-1}. If lambda is very small, the Gauss-Newton system will be solved. If in a step the objective function is not reduced λ is decreased, e.g. by a factor of ten, and a new correction is calculated by solving the linear equation system with the modified matrix. An iteration is terminated if there was a reduction of the objective function or if the maximal value for λ is reached. After a successful iteration step λ is always reduced by a given factor (e.g. ten). Algorithm 5.1 describes a whole Levenberg-Marquardt scheme as described by Press et al. 1992. The algorithm is terminated if there is no convergence with the maximally allowed λ, if the residual is small enough, the reduction in the last time step is below a certain threshold or if the number of iterations is too large.

5.2.3. Assembly of Hessian

The Assembly of the Hessian requires the knowledge of the partial derivatives of the model values to the different parameters. These derivatives are calculated by numerical derivative

$$\frac{\partial f_i(x_i, \boldsymbol{\beta}^k)}{\partial \beta_m} \approx \frac{f_i(x_i, \boldsymbol{\beta}^k + \Delta\beta_m \boldsymbol{e}_k) - f_i(x_i, \boldsymbol{\beta}^k)}{\Delta\beta_m}, \tag{5.12}$$

with increment $\Delta\beta_m = 10^{-4} \cdot \beta_k$ and $\boldsymbol{e}_k$ being the k-th Cartesian unit vector. As the measurement functions f_i are all evaluated during the same model run, the number of additional model runs to assemble the Hessian is equal to the number of parameters (the run with unmodified parameters is already needed to calculate the objective function).

5.2.4. Solution of linear Problems

If the model does not depend on a certain parameter, the matrix will be singular. With a non-linear model this can also be a temporary effect depending on the actual parameter combination. A linear solver thus has to be able to cope with singular matrices and

Algorithm 5.1: Levenberg-Marquardt algorithm (after Press et al. 1992)

```
function LevenbergMarquardt($\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\beta}_0, \lambda_0, \lambda_{\mathsf{min}}, \lambda_{\mathsf{max}}, \mathsf{maxit}, \epsilon_{\mathsf{rel}}, \epsilon_{\mathsf{abs}}$)
    $\beta \leftarrow \beta_0$
    $\lambda \leftarrow \lambda_0$
    $d_0 = d = F(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\beta})$
    $m = 0$
    repeat
        $d_{\mathsf{old}} = d$
        assemble matrix $\mathbf{A}$, vector $\boldsymbol{b}$
        $\mathbf{A}' \leftarrow \mathbf{A}$
        repeat
            for $(i = 0; i < N; i = i + 1)$ do
                $A'_{ii} = A_{ii} \cdot (1 + \lambda)$
            end for
            solve $\mathbf{A}' \cdot \boldsymbol{\nu} = \boldsymbol{b}$
            Check($\nu, \beta$)                          ▷ Make $\nu$ fulfil the parameter constraints
            compute $d = F(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\beta} + \boldsymbol{\nu})$
            if $d \geq d_{\mathsf{old}}$ then
                $\lambda = 10 \cdot \lambda$
                if $\lambda > \lambda_{\mathsf{max}}$ then
                    throw exception
                end if
            else
                $\lambda = \min(\lambda/10, \lambda_{\mathsf{min}})$
            end if
        until $d < d_{\mathsf{old}}$
        $\beta \leftarrow \beta + \nu$
        $m = m + 1$
    until $(d < \epsilon_{\mathsf{rel}} d_0) \vee (|d - d_{\mathsf{old}}| < \epsilon_{\mathsf{rel}}) \vee (|d - d_{\mathsf{old}}| < \epsilon_{\mathsf{abs}}) \vee (m \geq \mathsf{maxit})$
    if $m = \mathsf{maxit}$ then
        throw exception
    else                                                  ▷ algorithm converged
        Assemble matrix $\mathbf{A}$
        return $d, \boldsymbol{\beta}, \mathbf{A}^{-1}$      ▷ return residual, parameters and covariance matrix
    end if
end function
```

return a sensible solution for the non-singular parameters. One possibility is a singular value decomposition, where the $M \times N$ matrix $\mathbf{A}$ is decomposed into

$$\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T \tag{5.13}$$

where $\mathbf{U}$ is a unitary $M \times M$ matrix, $\mathbf{\Sigma}$ is a $M \times N$ diagonal matrix and $\mathbf{V}$ is a $N \times N$ unitary matrix. The (non-negative) diagonal values of $\mathbf{\Sigma}$ are called singular values. The pseudo inverse $\mathbf{A}^+$ of $\mathbf{A}$ can be calculated from this singular value decomposition as

$$\mathbf{A}^+ = \mathbf{V}\mathbf{\Sigma}^+\mathbf{U}^T \tag{5.14}$$

where the pseudo inverse of $\mathbf{\Sigma}$ is obtained by replacing all non-zero diagonal elements with their reciprocal. With the pseudo inverse it is possible to solve the linear equation system using only the equations with non-zero singular value. For non-singular matrices the pseudo inverse of the matrix is identical to the inverse of the matrix. The complexity of the singular value decomposition is a factor two times higher than the complexity of Gaussian elimination. This is completely irrelevant as the number of parameters will always be small.

5.2.5. Covariance Matrix

The uncertainty of the parameters and the correlation between them is indicated by the covariance matrix of the linearised problem. It is the inverse of the Hessian. An approximation can be obtained by inverting the approximated Hessian from the Gauss-Newton scheme or the Levenberg-Marquardt scheme with $\lambda = 0$

$$\mathbf{C}_{\beta\beta} = \mathbf{A}^{-1}. \tag{5.15}$$

To obtain a covariance matrix also for singular matrices, again the pseudo inverse is calculated from the singular value decomposition.

The square root of the reciprocal of the diagonal element gives the standard deviation of the parameters in the linearised problem. The correlation coefficients

$$\rho_{ij} = \frac{C_{\beta\beta,ij}}{\sqrt{C_{\beta\beta,ii} \cdot C_{\beta\beta,jj}}} \tag{5.16}$$

give normalised information about the correlation between individual parameters. If the correlation coefficient is close to plus or minus one, the parameters are directly or indirectly proportional. If it is close to zero, there is no correlation between the parameters.

Covariances, standard deviations and correlation coefficients calculated this way have to be handled with care as they are the values for an approximation of the linearised problem. Uncertainties for the non-linear problem can be much larger. However, their assessment would require e.g. some kind of Monte-Carlo scheme.

5.2.6. Parameter Constraints

For many of the parameters of a water flow model there are limits in which the parameters have to be, either due to physical restrictions or due to heuristic knowledge. For example the saturated hydraulic conductivity has to be a strictly positive number due to physical reasons and within the range of 10^{-9} and 10^{-1} m/s for most materials due to experience. Sometimes it is also necessary, that some values are in ascending or descending order, for example the saturations and matric potentials in a spline interpolation.

There is another reason to constrain parameter updates: If the model is only marginally sensitive to a certain parameter, a huge update will be calculated. This is often the case in the first steps of the optimisation when the parameter estimate is far away from the true values. The resulting parameters after such a huge update can result in forward problems which are very hard to solve, or in situations where other parameters do no longer have any effect. Thus the maximal change in a parameter is limited to one order of magnitude.

After calculating the correction in the Gauss-Newton or Levenberg-Marquardt algorithm the correction is first restricted to one order of magnitude. The resulting parameters are then checked with regard to them being within the range of validity. If this is not the case the correction is reduced to yield exactly the value at the boundary of the validity interval. A user definable function is then called, which can perform additional checks. *After* all the checks, the new objective function is calculated and the convergence checks are performed.

5.3. Implementation

The functionality for parameter estimation is directly included into the $\mu\varphi$ library. It forms a framework for general parameter estimation. The advantage of an own implementation over the usage of an existing external tool is that it is easier to integrate additional functionality like ordering constraints or parallel parameter estimation. The optimisation functionality is implemented both for the sequential and the parallel version of $\mu\varphi$.

5.3.1. Object Hierarchy

The central classes of this framework are (figure 5.2)

`FitDataClass` provides an interface to store experimental data which should be fitted, together with the type of the measurement, its measurement error, a flag if it should be included into the optimisation process and the index of this measurement in a continuous vector of measurement data needed by the optimiser

`FitModelClass` provides an interface for a model used to generate the model data for the measurements and for the handling of parameters.

`FitClass` provides a general interface for parameters to be optimised and objects of `FitDataClass` and `FitModelClass` need by any optimisation algorithm.

`IterativeFitClass` implements the common functionality for the iterative optimisation algorithms. This includes methods to assemble the sensitivity matrix (the approximated Hessian) and the residuum vector using numerical differentiation, to calculate the correction from the assembled matrix, calculate the covariance matrix and a `fit` method, which starts individual iteration steps and tests for convergence of the optimisation (algorithm 5.2). It also stores convergence criteria and parameters for numerical differentiation.

Based on these classes it is only necessary to implement the operations for one iteration step in the classes `LevMarqClass` and `GaussNewtonClass`. This includes the manipulation of the diagonal elements of the sensitivity matrix by the Levenberg-Marquardt algorithm or the line search for the Gauss-Newton scheme.

Algorithm 5.2: Generic part of an iterative fitting algorithm

```
function FIT($\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\beta}_0, \mathsf{maxit}, \epsilon_{\mathsf{rel}}, \epsilon_{\mathsf{abs}}$)
    $\beta \leftarrow \beta_0$
    $d_0 = d = F(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\beta})$
    $m = 0$
    repeat
        $d = \text{FITSTEP}(d_{\mathsf{old}}, \boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\beta})$
        $m = m + 1$
    until $(d < \epsilon_{\mathsf{rel}} d_0) \vee (|d - d_{\mathsf{old}}| < \epsilon_{\mathsf{rel}}) \vee (|d - d_{\mathsf{old}}| < \epsilon_{\mathsf{abs}}) \vee (m \geq \mathsf{maxit})$
    if $m = \mathsf{maxit}$ then
        throw exception
    else                                        ▷ algorithm converged
        Assemble matrix $\mathbf{A}$
        return $d, \boldsymbol{\beta}, \mathbf{A}^{-1}$    ▷ return residual, parameters and covariance matrix
    end if
end function
```

5.3.2. Data Management

A challenge for the implementation of the optimisation code is the management of the measurement data. For the optimisation algorithm it is only a vector of values with their standard error. But for the forward model it is also necessary to know at which time which kind of measurement has to be performed to generate the model value. Both tasks are handled by the `FitDataClass`.

One part of this class is the data file reader. Each file starts with a header in which the number of measurement times and the different measurement types and the relative or absolute standard error for each measurement are given. The measurement type is

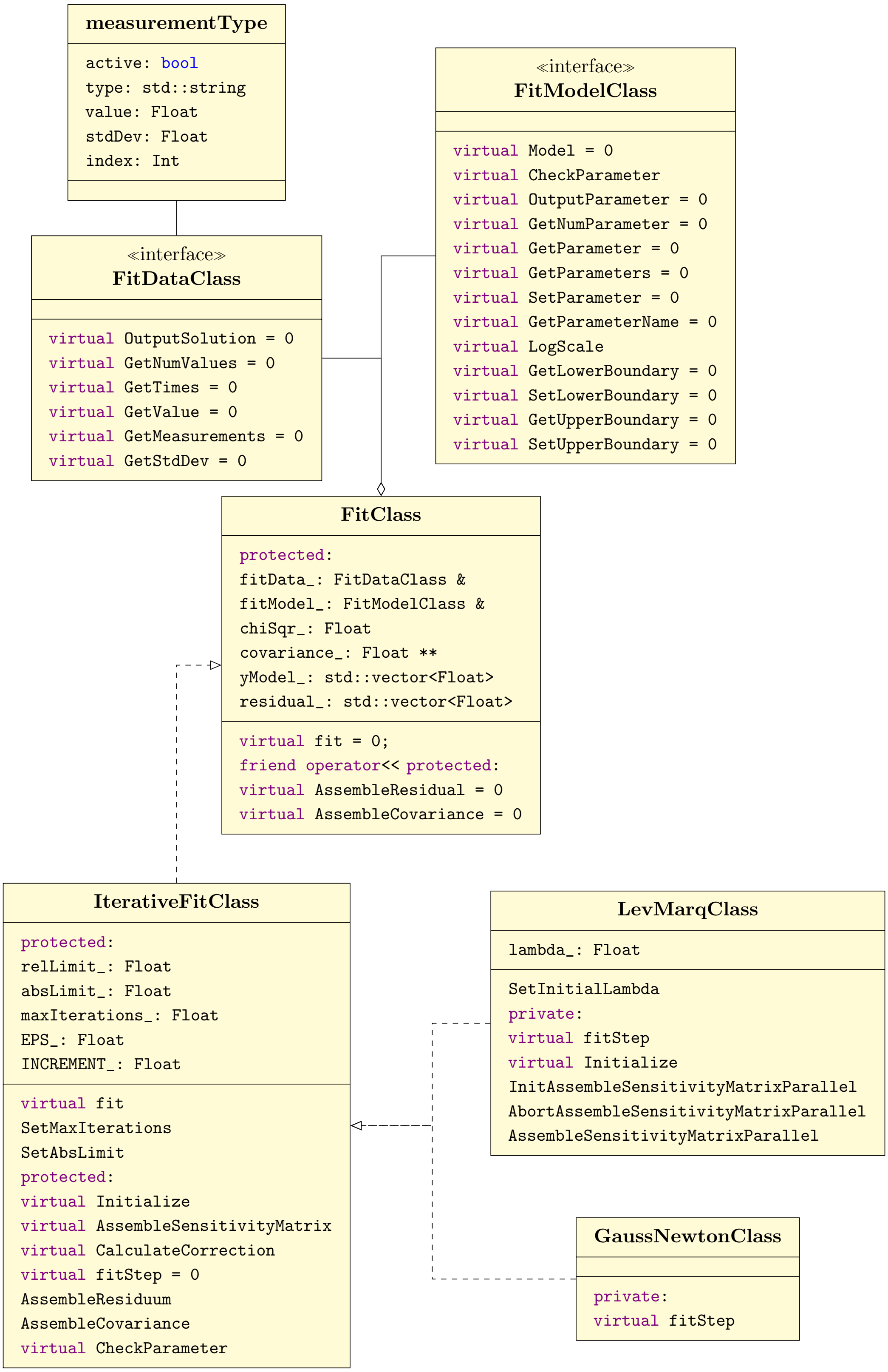

Figure 5.1.: Class structure of the parameter estimation part.

just a string, which is interpreted by the the forward model, and the number of different measurements is automatically derived from the number of white-space separated strings in the respective line. If the standard error is unknown, the values are a means to rescale the different measurements and define their relative weight. A standard error of zero deactivates the whole column of measurements for the parameter estimation process. A relative error is denoted by the char `'r'` after the value. The header is followed by the specified number of lines, each containing the time and the measured values. A missing value is marked by a negative value smaller than -1^{200}, which never occurs in reality. Comment lines starting with a hash mark can be inserted everywhere.

The class can provide the data in different forms. It can return a continuous vector of the experimental data and it can return just a vector of measurement times to be handed to the time control object to assure that the measurement times are exactly met by the forward model. For each of the measurement times it can return a list of the measurements which have to be performed together with information if and which measurement is to be evaluated and at which index of the result vector it has to be stored.

5.3.3. Model Implementation

From the perspective of the optimisation algorithm the model is just an object with a method which returns a vector of model results to be compared with the experimental data, and with methods to get and set the parameters to be estimated and their validity range. To guarantee this, each model is derived from an abstract base class. The actual model is mainly a $\mu\varphi$ main program with some additional methods to perform the measurements at the right time and methods to parse the parameter file and to handle the parameters. This includes the distinction between values which need to be estimated and values which are regarded as constant. Nearly every numerical quantity in an input file can be estimated (except parameters for space and time grid management). A parameter is marked for optimisation by adding three values after the parameter value (which then becomes the initial guess): an indicator if the value is to be fitted (0 or 1) and a lower and upper boundary for the value. From this information the model implementation determines at run-time the number of parameters to be estimated and provides a vector of the values to the optimisation algorithm. The set function defined in the interface of the base class then makes sure, that the value in the internal parameter structure of the forward model can be manipulated by the optimiser. Thus an existing forward model requires very few changes to be used in the inversion.

5.3.4. Parallelisation

The assembly of the Hessian is easy to parallelise using a master/slave approach. Each of the runs with a disturbed set of parameters can be executed by a different process. After all processes have been started and MPI has been initialised, only the master process will execute the full program. The slave processes wait until they receive a set of parameters via MPI. Every slave runs the forward model for its set of parameters and sends a vector of results to the master. The master assembles the sensitivity matrix, calculates the parameter updates and the new objective function. It performs the line search or the tests with increasing values of the Levenberg-Marquardt parameter λ until a reduction of the objective function has been achieved. During this phase the slaves are idle and only contribute again during the assembly of the next sensitivity matrix. It is assumed that there are as many processes as parameters, so load distribution is rather simple and on shared memory machines load balancing is done by the scheduler.

The process can be sped up and a better exploitation of the available computer power can be achieved if the matrix is always already assembled during the convergence check. If the iteration step is accepted, the next step can be performed immediately and the parallel run-time is the number of iteration steps times the run-time for one run of the forward model and thus independent of the number of parameters (though the number of processes required does increase with the number of parameters).

Both varieties have been implemented already for the sequential version of $\mu\varphi$ and can be chosen be a run-time switch. However, while this approach is sufficient for the inversion of one-dimensional simulations of experiments with homogeneous material, it is not appropriate for the simulation of heterogeneous materials. It also limits the number of parallel processes to the number of parameters and thus cannot even exploit mid-range clusters. Thus in combination with the parallel version of $\mu\varphi$, a two-level parallelism was implemented, where each forward simulation can be a parallel process. The total number of processes should be an integer multiple of the number of parameters plus one. To achieve this the `MPI_COMM_WORLD` communicator is split by colouring. The rank zero process of the master process group sends the parameters to all slaves and the rank zero process of the slave process groups send the results back to the rank zero process of the master process group, which calculates the updates.

Part II.

Solute Transport in Porous Media

6. Continuum Scale Component Transport

The wetting and non-wetting phase of a two-phase system can be composed of many different components (e.g. nutrients, pesticides, . . .). In this chapter we want to address the transport of components (solutes) dissolved in water (solvent) as wetting phase and ignoring phase change. The mean free path in water is well below typical pore sizes, thus effects like Knudsen-Diffusion can be ignored. Most aqueous solutions in natural porous media are dilute, i.e. collisions of dissolved components are mainly with water molecules and the density of the solution is (almost) the same as the density of water. The transport of each solute can then be assumed to be independent of the transport of other solutes and the component transport depends on the phase transport (via the velocity field) but not vice versa.

6.1. Convection-Dispersion Equation

The transport of solutes in an aqueous dilute solution can be described by the Convection-Diffusion Equation. In porous media the situation is a bit more complicated. The local velocities even in a single pore differ depending on the distance to the pore walls. Over different pores the spreading of flow velocities is even higher. In a near field limit this leads to stochastic convection, where the initial solute distribution is just propagated with the different flow velocities. Over time mixing and diffusion will result in an exchange between the different stream lines and in the far-field limit we get – as a consequence of the central limit theorem – again a convection-diffusion type equation (Roth 2012). The velocity for the convective part is given by the average pore water velocity $\boldsymbol{v} = \boldsymbol{j}/\theta_w$ obtained from the phase transport equations, where $\boldsymbol{j} = \boldsymbol{j}_w/\rho_w$ is the *volumetric* water flux density. Due to the additional spreading resulting from different pore scale velocities diffusion is increased, while due to the blocking influence of the solid matrix it is reduced compared to the molecular diffusion. The new effective diffusion coefficient is called dispersion coefficient and the equation is called the Convection-Dispersion Equation (CDE). As the effects in the direction of water flow and perpendicular to it are different, the dispersion coefficient is a full tensor. The dispersion tensor depends on the volumetric water content and also on the pore water velocity. The component transport equation than has the form

$$\boldsymbol{j_s} = \boldsymbol{j}_{s_{\mathsf{conv}}} + \boldsymbol{j}_{s_{\mathsf{disp}}} \tag{6.1}$$

with

$$\boldsymbol{j}_{s_{\mathsf{conv}}} = c_s \cdot \boldsymbol{j} \tag{6.2}$$
$$\boldsymbol{j}_{s_{\mathsf{disp}}} = -\mathbf{D}_s(\theta_w, \boldsymbol{v})\nabla c_s, \tag{6.3}$$

where c_s is the concentration of solute s and $\mathbf{D}_s(\theta_w, \boldsymbol{v})$ is the dispersion tensor.

6.1.1. Solute Dispersion

The combination of molecular diffusion, pore scale and macro-scale heterogeneity, diffusive mixing and convective mixing results in an increased macroscopic dispersion coefficient. This coefficient is a full tensor, which is symmetric with the main directions parallel (longitudinal) to and perpendicular (transversal) to the direction of water flow. According to Bear 1961 and Scheidegger 1961 it can be written as

$$D_{s_{ij}} = [\lambda_l - \lambda_t] \frac{v_i v_j}{||\boldsymbol{v}||_2} + [\lambda_t ||\boldsymbol{v}_w||_2 + D_{s_{\mathsf{eff}}}(\theta_w)]\, \delta_{ij} \tag{6.4}$$

The velocity dependent part is caused by hydromechanic dispersion. $D_{s_{\mathsf{eff}}}(\theta_w)$ is the part resulting from molecular diffusion. For the longitudinal dispersivity λ_l and the transversal dispersivity λ_t the condition $\lambda_l \geq \lambda_t$ holds. As the dispersivities have the dimension of a length and as they represent sub-scale features, they are smaller than the length scale at which the spatial heterogeneity of the soil is resolved. Thus if the spatial resolution is high and the flow velocity not too large, molecular diffusion is the dominant process.

Different models exist for the water content dependent reduction of the diffusion coefficient $D_{s_{\mathsf{eff}}}(\theta_w)$ by the solute matrix. A common feature of all these models is that the diffusion coefficient at saturation is given by $\Phi^{4/3} D_{s,\mathsf{m}}$, where $D_{s,\mathsf{m}}$ is the molecular diffusion coefficient for solute s in free water. Two popular parametrisations for the shape below saturation are given by the models of Millington 1959

$$D_{s_{\mathsf{eff}}}(\theta_w) = \frac{\theta_w^{10/3}}{\Phi^2} D_{s,\mathsf{m}} \tag{6.5}$$

and Millington and Quirk 1961

$$D_{s_{\mathsf{eff}}}(\theta_w) = \frac{\theta_w^2}{\Phi^{2/3}} D_{s,\mathsf{m}}, \tag{6.6}$$

also known as the first and second Millington-Quirk model.

6.2. Solute Adsorption

Hardly any solute is transported in a porous medium without some interaction with the solid matrix. A common mechanism is the adsorption of a solute to the surface of the solid phase. There are different binding mechanisms from van der Waals forces over electrical charges to weak chemical bonding.

Macroscopically adsorption is described by sorption isotherms (Moore 1990). They are relations between the solute concentration in the liquid phase and the mass or amount of substance of adsorbed solute. The easiest sorption isotherm is a linear relation. It assumes, that there is an unlimited amount of sorption sites, where each binding has the same energy. Thus the amount of adsorbed solute only depends on the concentration in the fluid phase and a material parameter characterising the intensity of the binding. These assumptions are nearly always true at low concentrations.

$$c_{s_{\mathsf{sorb}}} = K_s c_s \tag{6.7}$$

where K_s is the sorption parameter. The molar density of adsorbed solute is then given by

$$\nu_{s_{\mathsf{sorb}}} = \rho_b c_{s_{\mathsf{sorb}}} = \rho_b K_s c_s \tag{6.8}$$

where ρ_b is the bulk density of the soil, i.e. the mass of solid phase per volume of soil.

For linear adsorption

$$\theta_w c_s + \rho_b c_{s_{\mathsf{sorb}}} = \theta_w c_s + \rho_b K_s c_s = \left(1 + \frac{\rho_b K_s}{\theta_w}\right)\theta_w c_s \tag{6.9}$$

where the dimensionless term $R = 1 + \frac{\rho_b K_s}{\theta_w}$ is called retardation factor. The solution of the pure convection equation for linear sorption thus is the same as without sorption but with a time scale stretched by the factor R.

If the number of sorption sites is limited or the sorption sites have a different energy (which is always true at high-enough solute concentrations), the sorption isotherm gets non-linear. Two popular models are

Freundlich Isotherm

$$c_{s_{\mathsf{sorb}}} = K_F c_s^n$$

assumes that the energy of the adsorption sites decreases logarithmically

Langmuir Isotherm

$$c_{s_{\mathsf{sorb}}} = \frac{K_L c_{\mathsf{max}} c_s}{1 + K_L c_s}$$

assumes that adsorption occurs only in a mono-molecular layer with a limited number of sorption sites which all have the same energy, and that there is no interaction between neighbouring sorption sites.

The Convection-Dispersion equation including adsorption can be written as

$$\frac{\partial\left[\theta_w c_s + \rho_b c_{s_{\mathsf{sorb}}}(c_s)\right]}{\partial t} + \nabla \cdot (c_s \boldsymbol{j} - \mathbf{D}(\theta_w)\nabla c_s) + r_s = 0 \tag{6.10}$$

7. Numerical Solution of the Convection-Diffusion Equation

In this chapter we want to develop a numerical solution approach for the convection-dispersion equation. As the intended resolution is about 1 centimetre, the dispersivity is nearly of the same order as the molecular diffusion coefficient. Thus the error in only using a scalar diffusion coefficient is acceptable. The equation to solve is then given by

$$\frac{\partial(\theta_w c_s)}{\partial t} + \nabla \cdot [c_s \cdot \boldsymbol{j} - D_s(\theta_w)\nabla c_s] + r_s = 0 \tag{7.1}$$

The second model of Millington and Quirk (Millington and Quirk 1961) is used for the dependency of the diffusion coefficient on volumetric water content.

7.1. Classification of the Equation

To determine the type of the transient convection dispersion equation it is again necessary to transform it. Using product and chain rules one obtains:

$$-D_s(\theta_w) \cdot \frac{\partial^2 c_s}{\partial x^2} + \theta_w \cdot \frac{\partial c_s}{\partial t} + \left(j - \frac{\partial D_s(\theta_w)}{\partial x}\right) \cdot \frac{\partial c_s}{\partial x} + \left(\frac{\partial j}{\partial x} + \frac{\partial \theta_w}{\partial t}\right) \cdot c_s + r_s = 0. \tag{7.2}$$

Inserting the Millington and Quirk law one obtains:

$$-D_s(\theta_w) \cdot \frac{\partial^2 c_s}{\partial x^2} + \theta_w \cdot \frac{\partial c_s}{\partial t} + \left(j - \frac{2\theta_w D_m}{\Phi^{2/3}} \cdot \frac{\partial \theta_w}{\partial x}\right) \cdot \frac{\partial c_s}{\partial x} + \left(\frac{\partial j}{\partial x} + \frac{\partial \theta_w}{\partial t}\right) \cdot c_s + r_s = 0. \tag{7.3}$$

As with Richards' equation we again get a zero determinant

$$\det \begin{pmatrix} -D_s(\theta_w) & 0 \\ 0 & 0 \end{pmatrix} = 0 \tag{7.4}$$

and the additional rank condition

$$\operatorname{Rank} \begin{bmatrix} -D_s(\theta_w) & 0 & \left(j - \frac{2\theta_w D_m}{\Phi^{2/3}} \cdot \frac{\partial \theta_w}{\partial x}\right) \\ 0 & 0 & \theta_w \end{bmatrix} = 2. \tag{7.5}$$

In contrast to Richards' equation the transient convection diffusion equation is a linear partial differential equation as all coefficients only depend on properties of the flow field, the porous medium and the diffusion coefficient, but not on solute concentration. There are two interesting features:

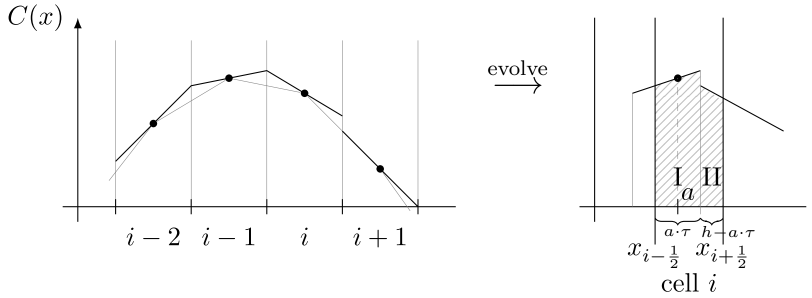

Figure 7.1.: REA principle: A piecewise polynomial function is reconstructed from the average concentration values (left), propagated and averaged again (right).

- A decrease in water content in the direction of flow increases the apparent flow velocity.
- A positive divergence of the flux field (a net source of water) acts like an increase of the water content. There is a dilution of the solute and a decrease in concentration. This marks the influence of sources or sinks for water only.

Like Richards' equation the CDE can get effectively hyperbolic on coarse grids. However, as the diffusion coefficient for solutes in water is rather small, this is much more probable. The grid Peclet number $\frac{hv}{2D}$ in this case is

$$\frac{h\cdot\left(j-\frac{2\theta_w D_{s,\mathrm{m}}}{\Phi^{2/3}}\cdot\frac{\partial\theta_w}{\partial x}\right)}{2D_s(\theta_w)}. \tag{7.6}$$

7.2. Spatial Discretisation

The discretisation is based on the assumption that convection-dominated flow is the rule and not the exception. Many different approaches are possible to solve the convection diffusion equation e.g. Streamline-Diffusion Finite-Elements (Brooks and Hughes 1982), Discontinuous-Galerkin schemes (Rivière 2008; Di Pietro and Ern 2012) and variations of methods tracking characteristics like the Modified Method of Characteristics (Douglas and Russell 1982), ELLAM (Binning and Celia 1996) or particle tracking (Bechtold et al. 2011). One drawback of many higher-order schemes is the occurrence of oscillations at sharp fronts, which due to the Satz of Godunov is unavoidable for linear methods which are more than first-order accurate (Leveque 2002).

A simple and monotonicity-preserving higher-order method for the solution of a first order hyperbolic PDE are Godunov methods in combination with a slope limiter. Godunov methods are based on the reconstruct, evolve, average (REA) principle (figure 7.1). A piecewise polynomial function is reconstructed from the cell averages, propagated with the velocity field, and the resulting concentration distribution is averaged over each element to calculate the new average concentrations. A piecewise constant reconstruction results in first-order full-upwind schemes. Second-order correct methods are obtained with a piecewise linear reconstruction. Different choices for the slope of the reconstructed linear function in each element lead to different discretisation schemes (Fromm, Lax-Wendroff, Beam-Warming). Monotonicity-preserving methods are obtained by the application of slope-limiters, which select the slope depending on the current solution. This makes the method non-linear and thus circumvents the Satz of Godunov.

As a Godunov method can be understood as a special kind of Finite-Volume scheme, the discretisation of the equation follows the same approach as described in section 3.3.1 yielding equation 3.22. Using C for the average solute concentration in the liquid phase, the storage, flux and source terms are given by:

$$\mathcal{M}_i(\boldsymbol{C}) = C_i \cdot \theta_{w_i} \cdot V_i \tag{7.7}$$

$$\mathcal{A}_{ij}(\boldsymbol{C}) = J_s(\boldsymbol{C}) \cdot n_j \cdot A_{ij} \tag{7.8}$$

$$\mathcal{Q}_i(\boldsymbol{C}) = R_{s_i} \cdot V_i \tag{7.9}$$

The numerical flux function $J_s(\boldsymbol{C}) = J_{s,\mathsf{diff}}(C_i, C_j) + J_{s,\mathsf{conv}}(\boldsymbol{C})$ consists of a convection and a diffusion part. Due to the piecewise linear reconstruction, the convection part can depend on the concentration in more than two cells. The diffusion part is given (as for water flux) by a difference quotient multiplied with a weighted harmonic mean of the two diffusion coefficients:

$$J_{s,\mathsf{diff}}(C_i, C_j) = -\frac{(h_i + h_j) D_i D_j}{h_j D_i + h_i D_j} \cdot \frac{C_i - C_j}{\frac{(h_i+h_j)}{2}} \tag{7.10}$$

As the diffusion coefficient does not depend on the concentration, no linearisation and stabilisation are necessary for the diffusive part. Due to the central-difference quotient this approximation is second-order accurate in space.

In an REA approach the numerical flux function for the convective part $J_{s,\mathsf{conv}}(\boldsymbol{C})$ is given by

$$J_{s,\mathsf{conv}}(\boldsymbol{C}) = j \cdot C_{i,i+1}, \tag{7.11}$$

where j is the volumetric water flux density and $C_{i,i+1}$ can be calculated from the REA approximation (Leveque 2002):

$$C_{i,i+1}^{k+1} = \begin{cases} C_i^k + \frac{1}{2}\left(h - v\tau^k\right) \cdot \sigma_i^k & \text{if } v \geq 0 \\ C_{i+1}^k - \frac{1}{2}\left(h + v\tau^k\right) \cdot \sigma_{i+1}^k & \text{if } v < 0 \end{cases} \tag{7.12}$$

with σ_i^k and σ_{i+1}^k being the slopes of the reconstructed linear function in element i and $i+1$, h the (equidistant) grid spacing, $v = j/\theta$ the pore water velocity and τ^k the time step.

In transient flow situations, however, the velocity v changes in space and time. Even in the RT_0 reconstruction used in this work it is already a linear function in space. Thus the determination of the correct propagation for the REA approach is complicated in itself.

In this work, we thus use a kind of MUSCL (monotonic upstream-centred scheme for conservation laws) approach (Leveque 2002) for the semi-discretisation of the hyperbolic part in space. The upstream concentration at the interface is reconstructed using a piecewise linear function

$$C_{i+1/2} = \begin{cases} C_i + \frac{h_i}{2} \cdot \sigma(\boldsymbol{C}, \boldsymbol{h}, i-1, i, i+1) & \text{if } j_{w_{i+1/2}} \geq 0 \\ C_{i+1} - \frac{h_{i+1}}{2} \cdot \sigma(\boldsymbol{C}, \boldsymbol{h}, i, i+1, i+2) & \text{if } j_{w_{i+1/2}} < 0 \end{cases}. \tag{7.13}$$

The indices of the concentrations and element width relevant for the calculation of the slope depend on the flow direction.

The concentration at the interface is then propagated with the flux density $j_{i+1/2}$ normal to the interface

$$J_{s,\mathsf{conv}}(\boldsymbol{C}) = j_{i+1/2} \cdot C_{i+1/2}. \tag{7.14}$$

In contrast to the pore water velocity the flux density is steady even over the interface between elements with different volumetric water content.

To prevent oscillations a minmod slope limiter is used to calculate σ_j. The minmod function is defined as (Leveque 2002)

$$\mathsf{minmod}(a,b) = \begin{cases} a & \text{if } |a| \leq |b| \text{ and } a \cdot b > 0 \\ b & \text{if } |b| < |a| \text{ and } a \cdot b > 0 \\ 0 & \text{if } a \cdot b < 0 \text{ (i. e. different sign)} \end{cases} \tag{7.15}$$

The slope can now be calculated from the concentrations at both sides of the interface and the concentration of the next upstream element. The generalisation to unstructured grids follows the suggestion of Leveque 2002:

$$\sigma(\boldsymbol{C}, \boldsymbol{h}, l, c, r) = \tilde{\sigma}(\boldsymbol{C}, \boldsymbol{h}, r, c) \cdot \mathsf{minmod}\left(1, \frac{\tilde{\sigma}(\boldsymbol{C}, \boldsymbol{h}, r, c)}{\tilde{\sigma}(\boldsymbol{C}, \boldsymbol{h}, c, l)}\right) \tag{7.16}$$
$$\text{with } \tilde{\sigma}(\boldsymbol{C}, \boldsymbol{h}, i, j) = \frac{C_i - C_j}{(h_i + h_j)/2}$$

As for the calculation of the slopes the neighbour after next is needed, this approach cannot be used close to the domain boundaries. Where it is not applicable, a full-upwind scheme is used.

7.3. Initial and Boundary Conditions

In contrast to Richards' equation, due to the linear nature of the convection-diffusion equation the initial and boundary conditions are rather simple. The concentration is given as initial condition. For the convection part boundary conditions are only to be given at inflow boundaries. As the flux is calculated as the flux density (which does not depend on the concentration) times concentration, Dirichlet and Neumann boundary conditions are equivalent. For the diffusive part Dirichlet or Neumann boundary conditions have to be given at all boundaries.

7.4. Time Discretisation

A one-step $\varTheta$ scheme with $\varTheta = 0$, i.e. an explicit Euler scheme is used for the time discretisation. The linear system is given by

$$\mathcal{M}_i(C^{k+1}) = \mathcal{M}_i(C^k) - \tau^k \cdot \left[\sum_{j=0}^{2d} \mathcal{A}_{ij}(C^k) + \mathcal{Q}_i(C^k) \right]. \tag{7.17}$$

As the flux functions and the sink terms only depend on the current solution and all equations are linear, the new concentration can be directly calculated without solution of a linear equation system.

The discretisation is first-order in time and second-order in space. To guarantee stability a CFL-condition has to be observed. For one-dimensional pure convection with a constant velocity v the CFL-condition is given by:

$$\mathsf{Courant} = \frac{v \cdot \varDelta t}{h} = \frac{j \cdot \varDelta t}{\theta_w h} \leq 1 \tag{7.18}$$

If diffusion is acting as well with a constant diffusion coefficient D, one can show that for the diagonal entry of the matrix to be positive (and thus the matrix to be an M-matrix) the restriction becomes

$$\varDelta t \left(\frac{j}{\theta_w h} + \frac{2D}{h^2} \right) \leq 1 \tag{7.19}$$

For this work we used the formulation

$$\frac{\max\left(|j_{i-1/2}|, |j_{i+1/2}|\right)}{h_i} + \frac{2D_i}{h_i^2} \leq \frac{\theta_{w_i}}{\varDelta t}. \tag{7.20}$$

For two- and three-dimensional simulations the term on the left was summed over all directions. With this time step restriction no oscillations occurred, while a slightly larger time step resulted in clearly detectable extrema.

Probably an even better criterion would be:

$$\max\left(0, -j_{i-1/2}\right) + \max\left(0, j_{i+1/2}\right) + \frac{2D_{i-1/2}}{h_{i-1} + h_i} + \frac{2D_{i+1/2}}{h_i + h_{i+1}} \leq \frac{\theta_{w_i} h_i}{\varDelta t} \tag{7.21}$$

where due to the upwinding only fluxes out of the element contribute to the restriction. However, as the use of slope limiters produces a complicated relation between C_i and the flux out of the element, both criteria are approximations anyhow.

For diffusion-dominated flow on very fine grids the time step can get prohibitively small due to the h^2 dependency of the time step condition for diffusion. For *diffusion dominated* flow thus an implicit time discretisation with central difference quotients for diffusion and convection is the method of choice, as it is stable and has no time step restriction.

7.5. Sorption

Linear sorption can easily be integrated into the discretised form by changing the storage term to

$$\mathcal{M}_i(\boldsymbol{C}) = (\theta_{w_i} + \rho_{b_i} K_{s_i}) \cdot C_i \cdot V_i \tag{7.22}$$

For non-linear sorption isotherms we get a non-linear partial differential equation with the non-linearity only on the time derivative. Thus it is sufficient to solve one additional scalar non-linear equation for each grid element. If a Langmuir isotherm is used, the storage term is given by

$$\mathcal{M}_i(\boldsymbol{C}) = \left(\theta_w C_i + \rho_{b_i} \frac{K_{L_i} c_{\mathsf{max}_i} C_i}{1 + K_{L_i} C_i}\right) \cdot V_i \tag{7.23}$$

With the explicit time discretisation we can calculate

$$M_i^{k+1} = M_i^k - \tau^k \cdot \left[\sum_{j=0}^{2d} A_{ij}^k + Q_i^k\right]. \tag{7.24}$$

We can then rewrite equation 7.23 to

$$\left(\frac{M_i^{k+1}}{V_i} - \theta_{w_i} C_i^{k+1}\right) \cdot \left(1 + K_{L_i} C_i^{k+1}\right) - \rho_{b_i} K_{L_i} c_{\mathsf{max}_i} = 0 \tag{7.25}$$

which is a quadratic equation in C_i^{k+1}. As the concentration has to be positive it has the unique solution

$$C_i^{k+1} = \frac{1}{2K_{L_i}\theta_{w_i}} \left[\left(K_{L_i} M_i^{k+1} - \rho_{b_i} K_{L_i} c_{\mathsf{max}_i} - \theta_{w_i}\right)\right. \tag{7.26}$$

$$\left. + \sqrt{\left(\rho_{b_i} K_{L_i} c_{\mathsf{max}_i} + \theta_{w_i} - K_{L_i} M_i^{k+1}\right)^2 + 4 K_{L_i} \theta_{w_i} M_i^{k+1}}\right] \tag{7.27}$$

The storage equation for the Freundlich Isotherm

$$M_i^{k+1} = \theta_{w_i} C_i^{k+1} + \rho_{b_i} K_{F_i} (C_i^{k+1})^{n_i} \tag{7.28}$$

is not directly invertible and it is necessary to solve the equation numerically. While this is not a problem, the Freundlich isotherm is not currently implemented in the solute transport model.

7.6. Interpolation of Water Content and Flux Densities

If the field of volumetric flux densities and water contents to be used for the calculation of solute transport stems from a solution of Richards' equation as described in the previous chapters, the time step for the solution of water transport and the time step given by the CFL-condition will in general not be the same and most often the time step for solute transport will be smaller. Thus several time steps for solute transport have to be performed for one time step of water transport. As a cell-centred Finite-Volume scheme, especially with an explicit time discretisation, is always mass conservative, eventual errors in the interpolation of water content and flux densities will show only in erroneous concentrations. One basic requirement which has to be met by a reasonable numerical solution of solute transport is that a constant concentration propagated in time will remain constant if there are no sources and sinks for water or solute only.

If an implicit Euler scheme with a locally mass-conservative cell-centred Finite-Volume scheme is used for the time-discretisation of Richards' equation, the divergence in the flux field at the new time is balanced by an increase in the volumetric water content from θ^k to θ^{k+1}. Let us assume that m sub-time steps are made in the time interval $[t^k : t^{k+1}]$ with a sub-time step size of $\tau^{k,l}$ and

$$\sum_{l=0}^{m} \tau^{k,l} = \tau^k. \tag{7.29}$$

For the solute transport water content is linearly interpolated in time

$$\theta_w^{k,l+1} = \theta_w^{k,l} + \tau^{k,l} \cdot \frac{\theta_w^{k+1} - \theta_w^k}{\tau^k} \tag{7.30}$$

with $\theta_w^{k,0} = \theta_w^k$ and $\theta_w^{k,m+1} = \theta_w^{k+1}$. The discretised PDE for the solute concentration C_i in element i is

$$\left(C_i^{k,l+1}\theta_w^{k,l+1}V_i - C_i^{k,l}\theta_w^{k,l}V_i\right) + \tau^{k,l}\left\{\sum_{j=0}^{2d}\left(J_s(C_i^{k,l}, C_j^{k,l}) \cdot n_j \cdot A_{ij}\right) + R_s^{k+1} \cdot V_i\right\} = 0. \tag{7.31}$$

For a concentration $C^k = C$ constant everywhere and Dirichlet boundary conditions with $C_D = C$ the concentration C^{k+1} should be C as well. This is the case if the discretised evolution equation is fulfilled for a constant concentration. If we assume that $R_s^{k+1} = C\frac{1}{\rho_w}R_w^{k+1}$ with constant water density ρ_w and sum the right-hand side over all sub-time steps we get

$$\sum_{l=0}^{m}\left[\left(C\theta_w^{k,l+1}V_i - C\theta_w^{k,l}V_i\right) + \tau^{k,l}\left\{\sum_{j=0}^{2d}\left(J_s(C,C)n_jA_{ij}\right) + C\frac{1}{\rho_w}R_w^{k+1}V_i\right\}\right] \tag{7.32}$$

$$= CV_i\sum_{l=0}^{m}\left(\theta_w^{k,l+1} - \theta_w^{k,l}\right) + \left(\sum_{l=0}^{m}\tau^{k,l}\right)\cdot\left[\sum_{j=0}^{2d}\left(J_s(C,C)n_jA_{ij}\right) + C\frac{1}{\rho_w}R_w^{k+1}V_i\right]. \tag{7.33}$$

The diffusive flux is zero for constant concentration and the concentration at the interface calculated with the slope limiters by equations 7.13 and 7.16 is C as well. The sum over the differences of the water contents is $\theta_w^{k,m+1} - \theta_w^{k,0} = \theta_w^{k+1} - \theta_w^k$ and the sum over all sub-time steps is τ^k according to equation 7.29. Thus the sum becomes

$$CV_i \sum_{l=0}^{m} \left(\theta_w^{k,l+1} - \theta_w^{k,l}\right) + \tau^k \left[\sum_{j=0}^{2d} \left(C\frac{1}{\rho_w} J_w(U_i^{k+1}, U_j^{k+1}) n_j A_{ij}\right) + C\frac{1}{\rho_w} R_w^{k+1} V_i\right] \tag{7.34}$$

$$= \underbrace{C\left\{V_i\left(\theta_w^{k+1} - \theta_w^k\right) + \tau^k \frac{1}{\rho_w}\left[\sum_{j=0}^{2d}\left(J_w(U_i^{k+1}, U_j^{k+1}) n_j A_{ij}\right) + R_w^{k+1} V_i\right]\right\}}_{=0 \text{ (discretised PDE for water transport)}} \tag{7.35}$$

$$= 0. \tag{7.36}$$

Thus constant concentration remains constant if the water content is linearly interpolated in time and if the fluxes and sink terms at time t^{k+1} are used. This also remains true, if a spatially refined grid is used for the solute transport, as the RT_0 field has an element wise constant divergence and linearly interpolated normal fluxes.

8. Implementation of an Efficient and Scalable Solver for the Convection Diffusion Equation

Solute transport was also implemented using C++ . In an explicit scheme very few operations have to be performed for each grid element. Thus efficiency is crucial. The code therefore uses templates more intensively.

A first parallel version was developed by Peter Bastian solving the CDE on equidistant grids with a constant diffusion coefficient and porosity. It was originally integrated with a solver for the stationary groundwater flow equation. As underlying grid it used an early version of YaspGrid, which is now part of the grid module of DUNE. An input for a (stationary) flow field, as well as support for spatially varying moisture content and a moisture dependent diffusion coefficient were added by the author to use results from $\mu\varphi$ simulations. In this version the whole parameter fields are read (and kept in memory) by every process. This limits its use to small clusters with ample memory. Linear and Langmuir-type sorption with material-dependent sorption parameters were also added to this model, which was used successfully in studies by Michela Rossi (Rossi et al. 2007; Rossi et al. 2008) at ETH Zürich to simulate solute transport in a highly heterogeneous sand tank experiment. Special output options were integrated for Beatrice Kulli, who performed simulations to create virtual experiments used in teaching at ETH Zürich (Kulli 2004).

To perform large-scale simulations with transient flow fields and moisture distributions on tensor-product grids, a new version of the solute transport model was developed in 2012. It uses the parallel grid developed for $\mu\varphi$ and the same distributed parallel I/O. The implementation was mainly done by Jorrit Fahlke. At the moment this massively parallel version does not support adsorption. However, this can be added very easily.

8.1. Parallelisation

Parallelisation is again carried out with a domain decomposition approach. The parallel grids developed for water transport can be for the computations (section 4.4.1). The rectangular sub-domains are chosen to match the sub-domains with which the flow field was calculated. However, due to the use of slope limiters, a larger overlap of two is required. As the slope limiter only needs the concentration and the element width of the second to next element, this element was not added to the general element information. Rather two methods where added to the grid interface, which return the index and element width for elements with an arbitrary distance from the current element in a specified direction. The same ISTL vector format as for water transport (section 4.4.2) is used to store concentrations, water contents and porosities. Thus the communication facilities of DUNE can be used to communicate information for the overlap.

8.2. File I/O

The keyword value based I/O system of $\mu\varphi$ is also used for the parameter file of the solute transport model. The adjustable units system will be integrated in the near future. At the moment time is measured in seconds, length in metre and concentration in mol/m^3. In principle it is even possible to use a single input file for both solute and water transport simulations, as the keywords are compatible.

The normal components of the volumetric flux density and the volumetric water contents are exchanged between $\mu\varphi$ and the solute transport code by data files. It is intended to later integrate the models more closely and directly integrate solute transport into the water transport simulation. Then, after one time step of water transport, solute transport will be simulated for the same time interval. This will need only very little extra memory, as the matrix needed for the water transport simulations is not needed during solute transport and the computed flux field does not need to be stored during water transport simulations. Thus the additional memory requirement would be in the order of one additional vector for the solute concentration after the last water time step. This will result in significant savings in computation time, as the input of flux fields and water contents can require a significant fraction of the total computation time (section 12.2.2.2).

The normal volumetric water fluxes are stored by $\mu\varphi$ with all values for one direction in consecutive fashion. The volumetric water contents are stored in the same file. Each process of $\mu\varphi$ stores the values for its interior partition (i.e. the value it owns) in a parallel I/O operation using SIONlib. Output must be performed after every time step, as the divergence of the fluxes and the change in water content will no longer balance else. The porosity is written in a separate file at the program start. The solute transport model initially reads the porosity file, and two files with normal fluxes and water contents needed for the simulation of the first time interval. Every process in a parallel operation reads only the values of its interior partition. The overlap values of porosity, water content and normal fluxes are communicated after the input is finished.

The solute transport model can store the concentration field and the normal components of the solute flux at points of time specified in the main input file again using parallel file I/O with SIONlib. A utility program is used to convert it to a Paraview readable vtr or a HDF5 file format. It also stores the boundary fluxes after every time step to obtain breakthrough curves.

8.3. Optimisation

Two aspects are important to achieve optimal performance on a modern high-performance CPU: Optimal use of memory bandwidth and exploitation of the SIMD units of the processor cores.

In each time step water content has to be interpolated. To reduce the necessary communication, the water content is also interpolated for the overlap. To enable the compiler to use SIMD instructions for the involved operations, they are performed in simple loops over the whole index range. Interpolation is done by

```
FLOAT factorS = (currentIntervalEnd_-time) / (currentIntervalEnd_ -
    currentIntervalStart_);
FLOAT factorE = (time-currentIntervalStart_) / (currentIntervalEnd_ -
    currentIntervalStart_);
for (size_t i=0;i<thetaStart_->size();++i)
  theta_[i] = (*thetaStart_)[i] * factorS + (*thetaEnd_)[i] * factorE;
```

These operations can easily be performed using SIMD instructions. Additionally the diffusion coefficient has to be calculated from the Millington-Quirk model as $D_{s,\mathrm{m}} \cdot \xi(\theta_w)$ where the tortuosity coefficient ξ is given by

$$\xi(\theta_w) = \frac{\theta_w^2}{\Phi^{2/3}} \tag{8.1}$$

After input the cube root of porosity is calculated using the fast library function `cbrt`. Thus the tortuosity coefficient can be calculated from the interpolated water contents by

```
xi = Dune::SQR(theta/porosity)
```

where the `SQR` function calculates the square by multiplication of the argument. This again is done in a vectorisable loop.

There is also some room for improvement in the calculation of a single time step. Let us assume that we have two vectors containing the concentration $\boldsymbol{C}$ and the amount of substance of solute per volume of soil $\theta_{w_i} V_i C_i$.

As the fluxes are continuous at interfaces, the flux for each interface needs only to be calculated once. Thus in a loop over all elements using the (old) concentration

- the fluxes are calculated for the interior faces in the direction of the coordinate axes.
- the face flux is multiplied by the area and the time step, and is added to the total concentration vector for the elements on both sides of the face
- boundary faces of the element are handled, but only influence the total concentration in the current element
- the source term times volume is added to the total concentration in the element

At the end of this loop the new total concentration is known. The new solute concentration is calculated from the total concentrations by division with $\theta_{w_i} V_i$ (again in a vectorisable loop) and the procedure can be repeated for the next time step.

For an explicit finite volume discretisation of a linear PDE memory bandwidth limitation is especially severe, as very few operations have to be performed with each value. Best exploitation of memory bandwidth and caches can be achieved if all involved data has the same ordering to avoid jumps between many different cache blocks. While this is easy to arrange for diffusion coefficients, volumetric moisture contents and concentrations it is harder to achieve for the volumetric water flux densities. In the file produced by $\mu\varphi$ they are stored in as little memory as possible with a storage requirement of $(N_x + 1) \times N_y \times N_z$ floating point values for the flux densities in x-Direction and $N_x \times (N_y + 1) \times N_z$ and $N_x \times N_y \times (N_z + 1)$ floating point values for the fluxes in y- and z-direction. However, this format is not optimal for the calculations. It proved advantageous to store the fluxes in a $(N_x + 1) \times (N_y + 1) \times (N_z + 1)$ field with the three components of the flux density required by the same element stored consecutively. Even as the flux field had to be reordered after import it still saved significant amounts of computation time.

The water content and flux field for the next time interval of water transport could also already be preloaded in a multi-threading operation during the calculation and be exchanged by a pointer swap with the current flow field and water content when they are needed. Unfortunately, SIONlib seams not to be thread-safe at the moment. However, the solute transport model already uses pointers to access the arrays for water content and flux vectors which are rotated to avoid copying.

8.4. Code Verification

A global mass balance is also a very good error indicator for solute transport. However, due to the explicit time stepping used for solute transport, it is only an indicator for grave programming errors. Most errors will rather show up in erroneous concentrations (e.g. negative concentrations or concentrations higher than the initial concentration).

8.4.1. Comparison with Analytical Solutions

An analytical solution for the breakthrough curve of a conservative tracer with a constant concentration at the inflow boundary of an initially tracer-free homogeneous medium is given by (Roth 1996):

$$\frac{C^f(t;z)}{C_0} = \frac{1}{2}\mathrm{erfc}\left(\frac{z/j - t/\theta_w}{\sqrt{4Dt/j^2}}\right) + \frac{1}{2}\exp\left(\frac{jz}{\theta_w D}\right)\mathrm{erfc}\left(\frac{z/j + t/\theta_w}{\sqrt{4Dt/j^2}}\right) \tag{8.2}$$

where $\mathrm{erfc}(x) = 1 - \mathrm{erf}(x)$ is the complimentary error function. The equation has already been rewritten in terms of the volumetric flux density j. With the CDE given in the transient form of equation 7.1, the diffusion coefficient D is given by $D = D_{s_\mathsf{eff}}(\theta)/\theta$. $C^f(t;z)$ is the flux concentration at depth z at time t. The flux concentration is defined by $C^f = j_s/j$ and is the measured quantity in breakthrough experiments.

If the transport is strongly convection-dominated, the second summand of equation 8.2 goes to zero while the first factor of it goes to infinity and the second factor to zero.

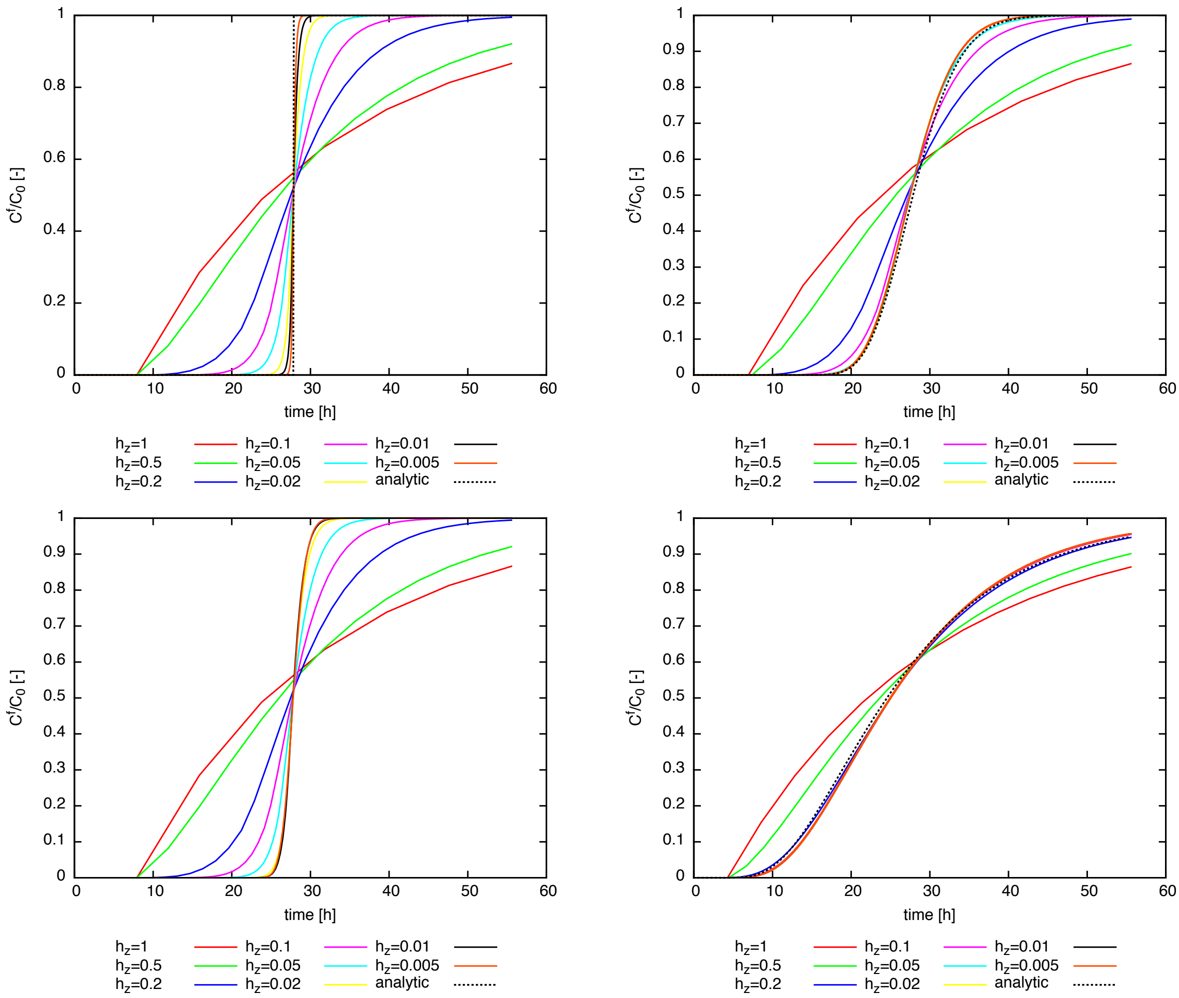

Figure 8.1.: Flux concentration in the outflow at the lower boundary of the domain for a constant velocity and diffusion coefficients of 0 (top left), 10^{-8} (bottom left), 10^{-7} (top right) and 10^{-6} m^2/s (bottom right).

The analytical solution under these conditions can be calculated more robustly by the approximation:

$$\frac{C^f(t;z)}{C_0} = \frac{1}{2}\mathrm{erfc}\left(\frac{z/j - t/\theta_w}{\sqrt{4Dz\theta_w/j^3}}\right). \tag{8.3}$$

In the following comparison the second formulation was used if the argument of the exponential function was $\frac{jz}{\theta_w D} > 100$.

Transport with a constant velocity of 1.148 cm/h and diffusion coefficients of $0, 10^{-8}, 10^{-7}$ and 10^{-6} m^2/s of a step-function was simulated for a domain of 1 m length with the transient solute transport model. The time dependent flux concentration in water leaving the domain at the lower boundary is shown in figure 8.1 together with the respective analytical solution. In all test cases the solution approaches the analytical solution with decreasing mesh size with grid convergence in the picture norm.

Part III.

Applications

9. Parameter Estimation for Heterogeneous Soil Samples

The determination of the hydraulic functions of porous media is one of the preconditions for process simulation. Popular methods for this are the evaporation method (e.g. Peters and Durner 2008; Schindler et al. 2010) and multi-step outflow experiments (see Van Dam et al. 1994; Hopmans et al. 2002; Durner and Iden 2011). Recently combinations of both have been suggested (Schelle et al. 2011) to improve the quality of the estimated parameters and extend the pressure range for which parameters can be estimated reliably. A crucial assumption in all these methods is the assumption that the soil sample can be considered as a homogeneous material. The inverse option of $\mu\varphi$ has also been used to estimate parameters from evaporation experiments (Schneider et al. 2006) and multi-step outflow experiments (e.g. Vogel et al. 2008) for homogeneous samples.

If the assumption of homogeneity fails at the scale of typical laboratory samples it is hard to determine the hydraulic properties of the sub-scale materials. The homogeneous regions are often too small to take samples of a sufficient size. One possibility to solve this problem is the use of downscaling approaches, where the parameters of the sub-scale materials are derived from some proxy information like the density in X-ray tomography (e.g. Vogel et al. 2002). The direct estimation of the material properties from multi-step outflow experiments with inverse modelling of the heterogeneous sample would be an attractive alternative. Without additional knowledge the estimation of distributed hydraulic properties is a hopeless endeavour as the number of data points is much smaller than the number of parameters to estimate. However, if we assume that the internal structure of the sample is known e.g. by X-ray tomography it might be possible to reduce the problem to the estimation of hydraulic functions for a small number of materials, the spatial arrangement of which is known. It might even be safe to assume that a material which is denser in an X-ray tomography has also a smaller hydraulic conductivity. In this chapter the feasibility of this approach will be examined using a synthetic test case and experimental data.

9.1. Multi-step Outflow Experiments

In multi-step outflow (MSO) experiments a soil sample is placed on a ceramic plate, a pressure gradient is applied to the sample and the time evolution of the outflow, the matric potential and sometimes also the water content at one or more locations in the sample is measured. The pressure at the boundary is changed step-wise to cover a range of potentials. There exist two fundamentally different setups. In one version the

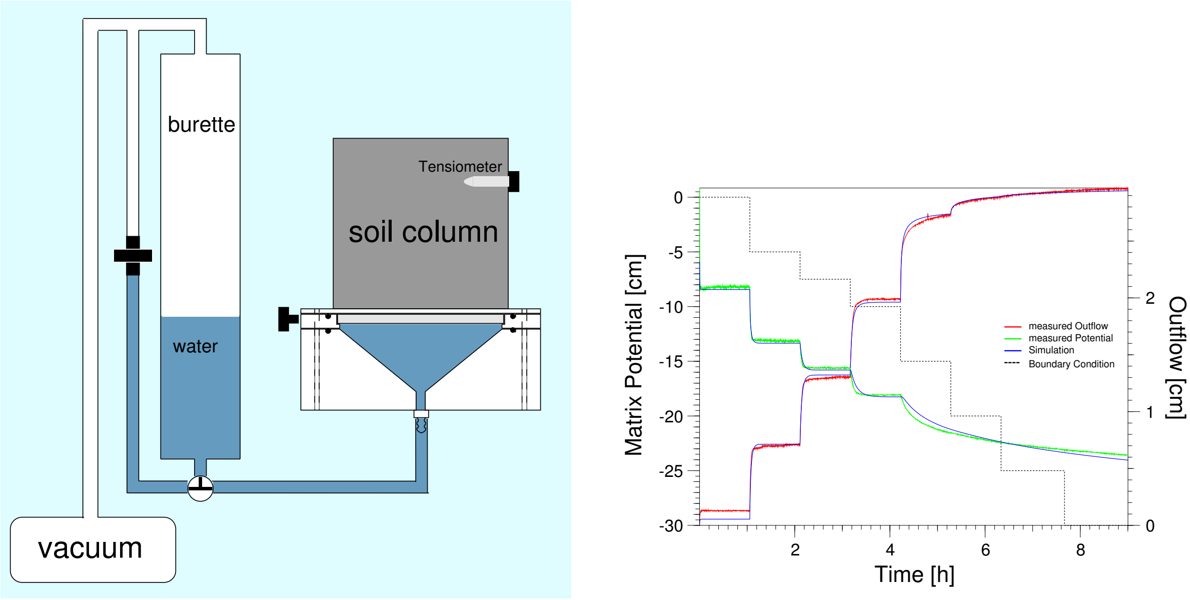

Figure 9.1.: Multi-Step Outflow Experiment: Setup (left) and typical result (right).

pressure of the gas phase above the sample is increased above atmospheric pressure to enforce the outflow. In the alternative approach the pressure of the water phase below the ceramic plate is decreased (figure 9.1). The water is collected in a burette and the amount of water can be measured with pressure transducers (Schultze 1998). This has the advantage that the pressure can also be increased again to perform not only drainage but also imbibition experiments. The direct forcing of the more viscous liquid phase should also avoid viscous fingering which is to be expected if the pressure of the gas phase is increased rapidly. A typical size for soil samples in MSO experiments is a diameter of 16 cm and a height of 10 cm.

9.2. Synthetic Test Case

To test the possibility of estimating parameters for the hydraulic functions of different materials simultaneously, an MSO experiment with a simple heterogeneous sample consisting of a fine and a coarse sand was simulated with $\mu\varphi$. The structure (figure 9.2) was chosen to allow a two-dimensional radially symmetric simulation, which of course is much faster than a full three-dimensional inversion. The ceramic plate on which the sample was placed is included in the simulation to account for the potential damping effect of the plate. Random noise was added to the simulated measurement values of outflow and matric potential at one position in the sample. The pressure steps at the boundary and the resulting curves of outflow and potential are shown in figure 9.3. The parameters used for the simulation are given in table 9.1 and table 9.2.

The optimisation process was started with rather general parameters, which were identical for both materials except for the van Genuchten α. The value of α for the coarse

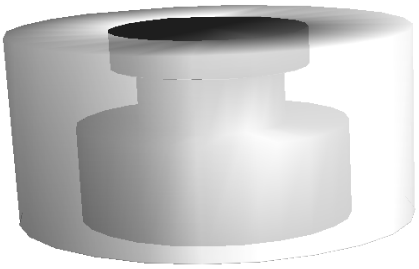

Figure 9.2.: Structure used for the synthetic test case with coarse coarse sand in the middle (black) and fine sand at the outside (light grey).

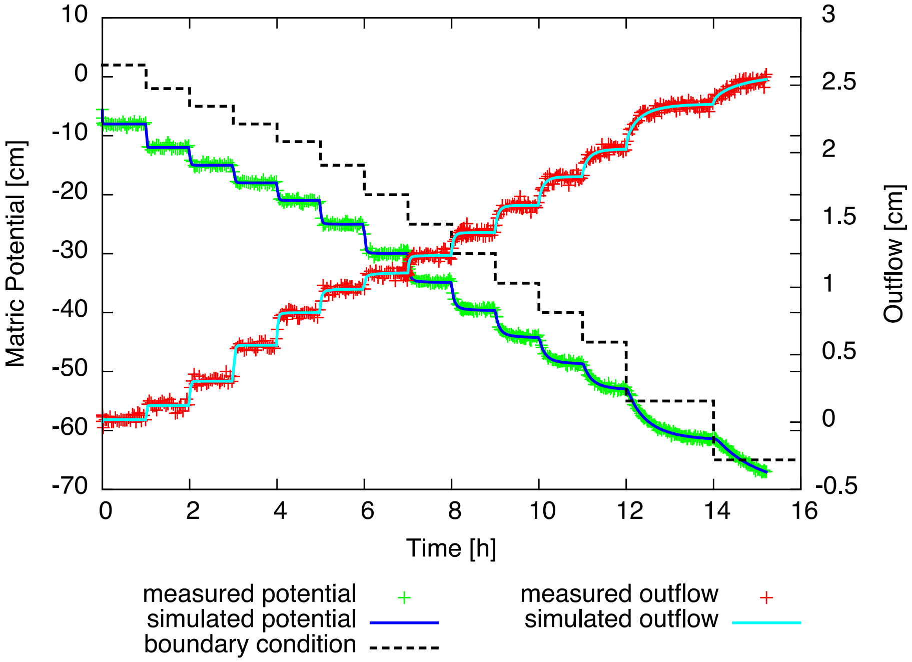

Figure 9.3.: Boundary condition, measured values and best fit for the synthetic test case.

parameter	unit	value
h_x	cm	0.2
h_z	cm	0.2
n_x	–	40
n_z	–	60
Plate thickness	cm	2
Height of tensiometer	cm	10
K_{plate}	cm/h	20.
θ_r	–	0.0
τ	–	0.5

Table 9.1.: Fixed parameters for the synthetic test case.

parameter	unit	value		
		exact	initial	estimated
coarse sand				
α	cm^{-1}	0.080	0.2	0.0803 ± 0.0003
n	–	6.2	2.0	6.42 ± 0.03
K	cm/h	120.	40	$188. \pm 2$
θ_s	–	0.30	0.34	0.297 ± 0.001
fine sand				
α	cm^{-1}	0.021	0.1	$0.02093 \pm 4 \cdot 10^{-5}$
n	–	4.5	2.0	4.46 ± 0.02
K	cm/h	20.	40.	19.6 ± 0.1
θ_s	–	0.30	0.34	0.304 ± 0.001

Table 9.2.: True, initial and estimated parameters for the synthetic test case.

	α_c	α_f	K_c	K_f	$\theta_{s,c}$	$\theta_{s,f}$	n_c
α_f	0.37						
K_c	0.21	0.07					
K_f	0.17	0.24	0.01				
$\theta_{s,c}$	-0.67	-0.53	-0.26	-0.18			
$\theta_{s,f}$	0.42	-0.21	0.15	-0.02	-0.62		
n_c	-0.72	-0.58	0.11	0.07	0.50	-0.09	
n_f	-0.42	-0.28	-0.10	-0.24	0.64	-0.65	0.14

Table 9.3.: Correlation coefficients for the synthetic test case. Coarse sand parameters are marked by a subscript c and fine sand parameters with a subscript f.

sand was set to twice the α value of the finer material. Without additional constraints the measured parameters were not satisfying. There are two sets of parameters which yield a local minimum for the estimation problem, with the two materials just exchanged. Only when it was additionally required that $\alpha_c > \alpha_f$, the correct branch was selected. The agreement between true and estimated parameters was very good (table 9.2). However, the results also show the limited significance of the standard deviation of the linearised problem. For the conductivity of the coarse sand the standard deviation is more than a factor of 20 smaller than the difference between true and estimated value. The saturated hydraulic conductivity is notoriously hard to estimate from MSO experiments. Thus the large error in the conductivity of the coarse sand is not surprising considering the low conductivity of the ceramic plate. Several parameters are correlated but the correlation is not very high. The highest (negative) correlation of -0.72 occurs between the van Genuchten n and α values of the coarse sand.

This test case shows, that for a sample with a rather simple structure composed of two materials, which have contact to both phases at the boundary, the parameters of the two van Genuchten models can be estimated reliably from a single multi-step outflow experiment.

9.3. Undisturbed Soil Sample

For a more realistic test the optimisation approach was applied to an MSO experiment with an undisturbed soil sample (Orthic Luvisol) taken from the upper A-horizon of a silty agricultural soil near Merzenhausen, Germany (Kasteel et al. 2000). The structure of the sample was measured by X-ray tomography. A solute transport experiment was conducted with the same sample and can be used to verify the correctness of the (principally unknown) parameters. The sample consists of two different materials: dense aggregates and a lighter matrix (figure 9.4, left). The X-ray tomography also allows to distinguish macropores, the tensiometer ceramic and the tensiometer void.

The data was already used in previous studies. Kasteel et al. 2000 performed the experiments, segmented the tomography data into different materials (figure 9.4) and performed a spatially resolved simulation of the solute transport experiment with the Finite-Element code SWMS. To perform the water transport simulation the hydraulic functions of the light matrix were determined by network-modelling. For the dense aggregates the parameters of the matrix were used with a smaller α. It was assumed that the known saturated hydraulic conductivity of the whole sample can be calculated from a volume weighted geometric average of the conductivities of matrix and aggregates $\sqrt{K_{s,\mathsf{eff}}} = V_{\mathsf{mat}}\sqrt{K_{s,\mathsf{mat}}} + V_{\mathsf{agg}}\sqrt{K_{s,\mathsf{agg}}}$. This allowed a downscaling by assuming different ratios of the conductivities of the two materials. A good agreement between the measured and simulated data for the solute transport experiment was obtained.

Vogel et al. 2002 used a downscaling approach based on the grey values from the X-ray tomography to derive hydraulic parameters for 48 different density classes in the sample. They demonstrated, that a spatially resolved simulation of the MSO experiment with the downscaled parameters agreed well with the experimental data but they did not simulate the solute transport experiment.

In this study two different approaches to determine the hydraulic properties of the light matrix and dense aggregates are compared: a variation of downscaling in combination with a one-dimensional optimisation and a fully three-dimensional parameter estimation.

9.3.1. Parameter estimation with Dual van Genuchten/Mualem Model

The basic idea of a multiple van Genuchten model (section 2.3.1.3) is that a material is a composite of two or more materials, which are so intimately mixed, that they are always in local equilibrium. The aggregates in the sample are rather large to justify this assumption. The advantage of this approach is that parameter estimation can be performed with a one-dimensional homogeneous model. A modified dual van Genuchten/Mualem model was used to exclude an influence of the missing air entry value in the van Genuchten/Mualem

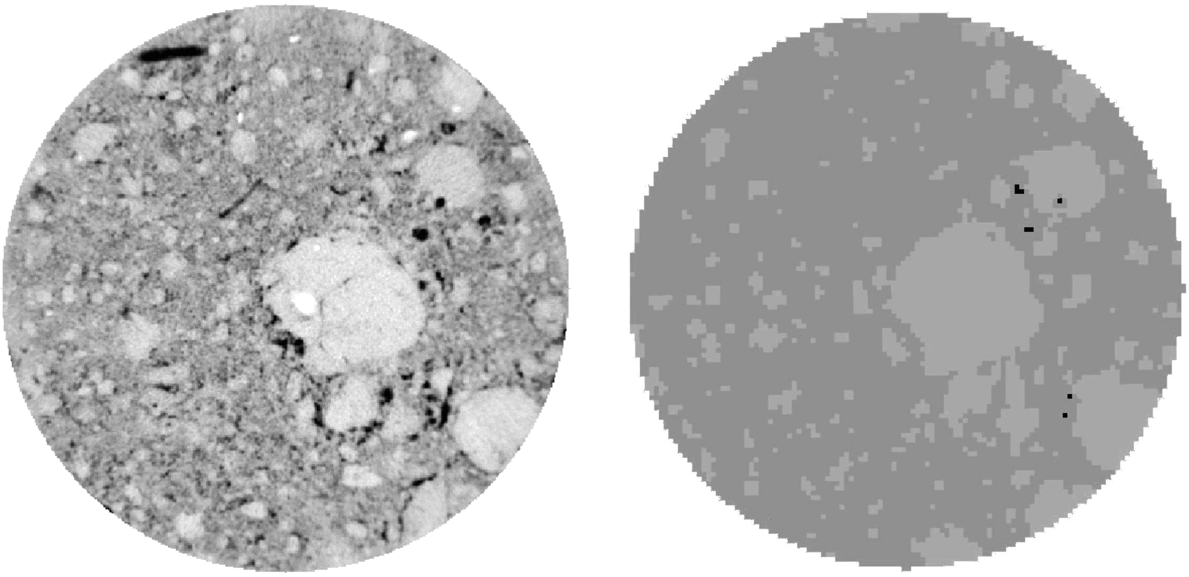

Figure 9.4.: Grey-scale CT-Image of the undisturbed soil sample (left) and averaged cross-section (right) segmented into light matrix (dark grey), dense aggregates (light gray) and macropores (black).

model. A modified dual van Genuchten/Mualem model requires ten parameters: the saturated conductivity K_{eff}, tortuosity τ, air-entry value ψ_{m_0}, the saturated and residual water content θ_s and θ_r of the composite material, the van Genuchten α and n values for the two sub-materials and the fraction f_{mat} of the light matrix contribution to the soil water characteristic (the volume fractions of the aggregates is given by $f_{\mathsf{agg}} = 1 - f_{\mathsf{mat}}$). As the measured outflow curve showed a rapid drainage already at the beginning, an additional leakage parameter was fitted, which is an offset added to the calculated outflow and accounts for water coming from gaps at the sample boundary and not from the sample itself. As it is impossible to estimate both the saturated and residual water content from outflow data alone, the residual water content was fixed at zero. The tortuosity was assumed to be 0.5 as suggested by Mualem 1976.

A parameter optimisation with nine parameters was performed starting with the same initial values as in the simple heterogeneous test case before. Unfortunately only the outflow data of the original MSO experiment is still available while the potential measurements are lost. There was a very good agreement between simulated and measured outflow, but the parameters had a high uncertainty (table 9.4) and some of them are strongly correlated (table 9.5) with the highest correlation coefficient of 0.994 between conductivity and air entry value.

This is usually an indicator for an over-parametrised model. A reduced model was thus developed. The n value for the aggregates was fixed to 2 as they seem to remain close to saturation and thus this value is very uncertain. As the n value for the matrix was well above 2, the air entry value was dropped and an ordinary multiple van Genuchten/Mualem model was used. Because of the high correlation between saturated water content and the matrix fraction, saturated water content was fixed at a value of 0.4. The

parameter	unit	value	
		initial	estimated
K_{eff}	cm/h	40	2.8 ± 3
θ_s	–	0.34	0.33 ± 0.1
θ_r	–	0.0	(not fitted)
τ	–	0.5	(not fitted)
ψ_{m_0}	cm	-0.01	-1.5 ± 1.8
f_{mat}	–	0.5	$0.0.2 \pm 0.1$
α_{mat}	cm^{-1}	0.2	0.06 ± 0.002
n_{mat}	–	2	4.4 ± 0.9
α_{agg}	cm^{-1}	0.1	0.02 ± 0.01
n_{agg}	–	2	1.5 ± 0.3 (not fitted)
leakage	cm	0	0.02 ± 0.05

Table 9.4.: Initial and estimated parameters for the undisturbed soil sample with a nine parameter modified dual van Genuchten/Mualem model.

	ψ_{m_0}	α_{mat}	α_{agg}	K_{eff}	f_{mat}	leakage	θ_s	n_{mat}
α_{mat}	0.549							
α_{agg}	-0.790	-0.774						
K_{eff}	0.994	0.480	-0.726					
f_{mat}	0.870	0.814	-0.950	0.819				
leakage	-0.956	-0.396	0.628	-0.977	-0.711			
θ_s	-0.864	-0.735	0.768	-0.835	-0.928	0.724		
n_{mat}	-0.580	-0.789	0.933	-0.504	-0.832	0.422	0.582	
n_{agg}	0.901	0.773	-0.873	0.862	0.979	-0.754	-0.982	-0.704

Table 9.5.: Correlation coefficients for the undisturbed soil sample with a nine parameter modified dual van Genuchten/Mualem model.

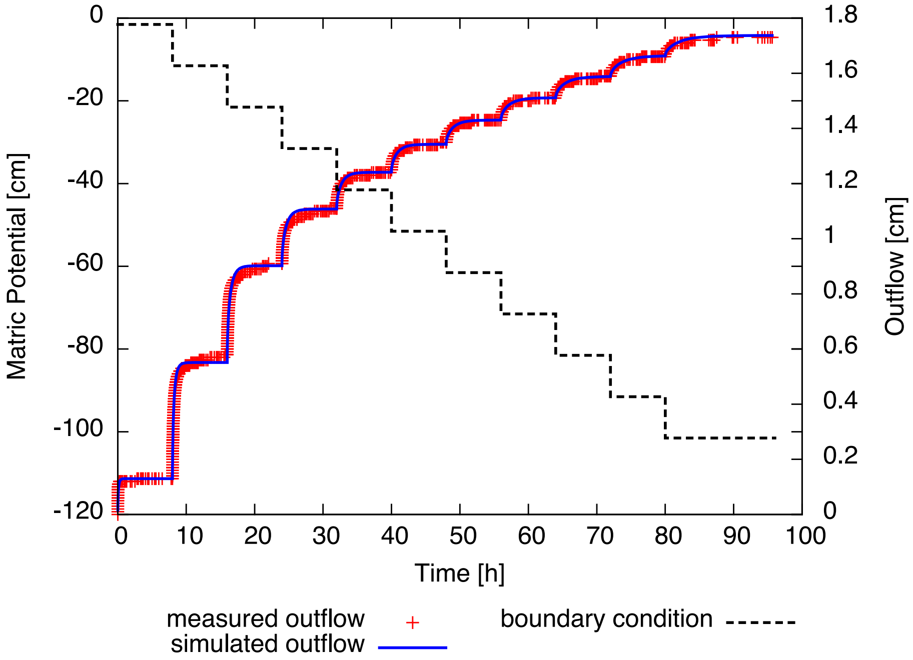

Figure 9.5.: Boundary condition, measured outflow and best fit for the undisturbed soil column with the five parameter model.

parameter	unit	value	
		initial	estimated
K_{eff}	cm/h	40	2.6 ± 0.1
θ_s	–	0.4	(not fitted)
θ_r	–	0.0	(not fitted)
τ	–	0.5	(not fitted)
f_{mat}	–	0.5	0.328 ± 0.005
α_{mat}	cm^{-1}	0.2	0.0588 ± 0.0005
n_{mat}	–	2	2.83 ± 0.06
α_{agg}	cm^{-1}	0.1	0.0063 ± 0.0001
n_{agg}	–	2.0	(not fitted)

Table 9.6.: Initial and estimated parameters for the undisturbed soil sample with a five parameter modified dual van Genuchten/Mualem model.

	α_{mat}	α_{agg}	K_{eff}	f_{mat}
α_{agg}	0.532			
K_{eff}	-0.579	-0.152		
f_{mat}	-0.432	-0.954	0.094	
n_{mat}	0.102	0.778	0.014	-0.909

Table 9.7.: Correlation coefficients for the undisturbed soil sample with a five parameter modified dual van Genuchten/Mualem model.

leakage parameter was obviously irrelevant and was dropped. The agreement between estimated and measured outflow for the resulting five parameter model was nearly as good as for the nine parameter model (figure 9.5) with residuum slightly increased from 63 to 65. The uncertainty of the parameters was markedly reduced (table 9.6). The correlation between parameters is also reduced. However, there is still a high correlation between the fraction of the matrix and the α value of the aggregates and the n value of the matrix. There seems to be not enough information in the experiment to fix the mobile water fraction of the aggregates and the potential at which they start to drain at the same time, while a reduced fraction of the matrix can be compensated by a higher n value.

9.3.2. Deduction of sub-material Porosity and Conductivity from Dual van Genuchten/Mualem Model

While the α and n values for the matrix and the aggregates can be directly taken from the multiple van Genuchten model, only the effective conductivity and saturated water content for the composite sample are estimated. Thus the deduction of these parameters has to be based on additional assumptions.

We assume that the effective conductivity of the composite material is given by the geometric mean of the saturated conductivity of the aggregates and the matrix, which was suggested by Matheron et al. 1967.

$$K = \sqrt{K_{\mathsf{mat}} \cdot K_{\mathsf{agg}}}. \tag{9.1}$$

To get another condition we assume that – as in Miller similarity scaling – the conductivity is proportional to the square of the α coefficient:

$$\frac{K_i}{K_j} = \left(\frac{\alpha_i}{\alpha_j}\right)^2. \tag{9.2}$$

Both conditions together allow the derivation of the saturated conductivity for the two materials from the effective conductivity of the composite as

$$K_i = \frac{\alpha_j}{\alpha_i} K_{\mathsf{eff}}. \tag{9.3}$$

The water content calculated by the multiple van Genuchten model has to be equal to the volume weighted sum of the water content calculated from the van Genuchten models for matrix and aggregates. If we assume a constant potential in the sample and call $\theta_s - \theta_r$ the drainable water content θ_{mob}, the amount of water coming out of the sample has to be equal for the composite and the spatially resolved column. With this we get the condition

$$\begin{aligned}\theta_{\mathsf{mob}} \left[1 - \left(f_{\mathsf{mat}} S_{\mathsf{mat}}(\psi_m) + f_{\mathsf{agg}} S_{\mathsf{agg}}(\psi_m)\right)\right] = \\ \theta_{\mathsf{mob,mat}} f_{\mathsf{vol,mat}} \left(1 - S_{\mathsf{mat}}(\psi_m)\right) + \theta_{\mathsf{mob,agg}} f_{\mathsf{vol,agg}} \left(1 - S_{\mathsf{agg}}(\psi_m)\right).\end{aligned} \tag{9.4}$$

If the drainable water content of the aggregates is known, the drainable water content of the matrix can be calculated from (omitting the common argument ψ_m of the saturation functions)

$$\theta_{\mathsf{mob,mat}} = \frac{\theta_{\mathsf{mob}} \left[1 - f_{\mathsf{mat}} S_{\mathsf{mat}} + (1 - f_{\mathsf{mat}}) S_{\mathsf{agg}}\right] - \theta_{\mathsf{mob,agg}} f_{\mathsf{vol,agg}} \left(1 - S_{\mathsf{agg}}\right)}{\left(1 - S_{\mathsf{mat}}\right) f_{\mathsf{vol,mat}}}. \tag{9.5}$$

The volume fraction of the aggregates from image analysis is 0.285 and the image fraction of the matrix 0.715. As the estimated fraction of the aggregates in the multiple van Genuchten model is 0.672 and thus much higher than the volume fraction of the aggregates, we assume that the whole saturated water content of the aggregates is drainable water.

The porosity of the sample is not sure. Kasteel et al. 2000 and Vogel et al. 2002 used a porosity of 0.5 and 0.503 for their work. The breakthrough curve suggests that the volumetric water content during the solute transport experiment was around 0.33. As the sample was still close to saturation this suggests a saturated water content of 0.4. As the saturated water content is crucial for the velocity of convection in the solute transport experiment, parameters have been calculated using saturated water contents of 0.4 and 0.5 (table 9.8) and will be used to assess the sensitivity on θ_s.

A forward simulation of the MSO experiment was performed with both parameter sets and the sample structure in two different resolutions (figure 9.6, middle and right). There are some additional materials involved in the forward simulation. Reasonable guesses were made for their parameters (table 9.9). As the sample has a cylindrical shape and is simulated with a rectangular grid, a background material is used for the elements outside the cylinder. For the MSO simulations a Brooks-Corey parametrisation with a high air-entry value and a very low conductivity was used which does not influence the result. For the plate, the tensiometer ceramic and the tensiometer void Brooks-Corey models with high air-entry value were used as well. The conductivity of the plate was measured in the experiment, the conductivity of the tensiometer void is just a very high value also used for the macropores. The macropores have parameters chosen to force a rapid drainage. The possibility to directly include potential disturbance by measurement devices like the tensiometer is an additional advantage of this approach.

The agreement is not perfect, with the reconstruction of the material property being only an approximation. The simulations with the high resolution structure (356 × 356 × 93 voxel) are a little bit closer to the measurements than the one with the medium resolution (118 × 118 × 62 voxel), but the difference is rather small.

parameter	unit	value	
		low porosity	high porosity
light matrix			
K	cm/h	24.3	24.3
τ	–	0.5	0.5
θ_s	–	0.4	0.5
θ_r	–	0.184	0.29
α	cm^{-1}	0.0588	0.0588
dense aggregates			
K	cm/h	0.279	0.279
τ	–	0.5	0.5
θ_s	–	0.4	0.5
θ_r	–	0.0	0.0
α	cm^{-1}	0.0063	0.0063
n	–	2.0	2.0

Table 9.8.: Parameters for the light matrix and the dense aggregates used in the forward simulation of the MSO experiment and in the simulation of solute transport.

K	τ	θ_s	θ_r	α	n	λ	ψ_{m_0}
macropores: van Genuchten							
1000	0.5	1	0	1	10		
ceramic plate: Brooks-Corey							
2.16	0.5	0.34	0	–	–	3	200
tensiometer ceramics: Brooks-Corey							
2.16	0.5	0.3	0	–	–	2	200
tensiometer void: Brooks-Corey							
1000	0.5	1	0	–	–	2	200
background MSO: Brooks-Corey							
10^{-10}	0.5	10^{-5}	–	–	2	200	
background solute transport: van Genuchten							
0	0.5	10^{-40}	0	20000	3		

Table 9.9.: Parameters used for the other materials involved in the three-dimensional forward simulation of the MSO experiment and of solute transport and in the three-dimensional optimisation.

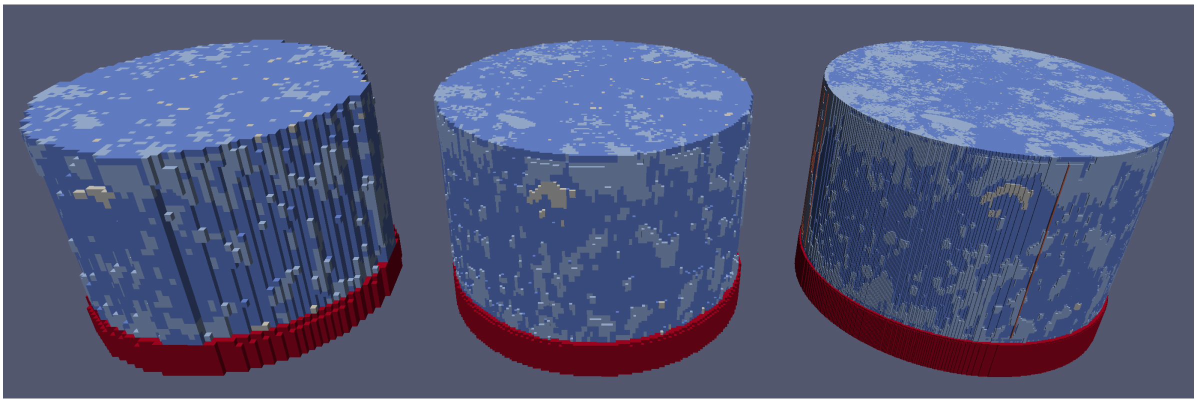

Figure 9.6.: Categorised structure in different resolutions used for the three-dimensional inversion (left) and for the forward simulations with medium (middle) and high resolution (right). The light matrix is shown in dark blue, the dense aggregates in light blue, macropores in grey and the ceramic plate in red.

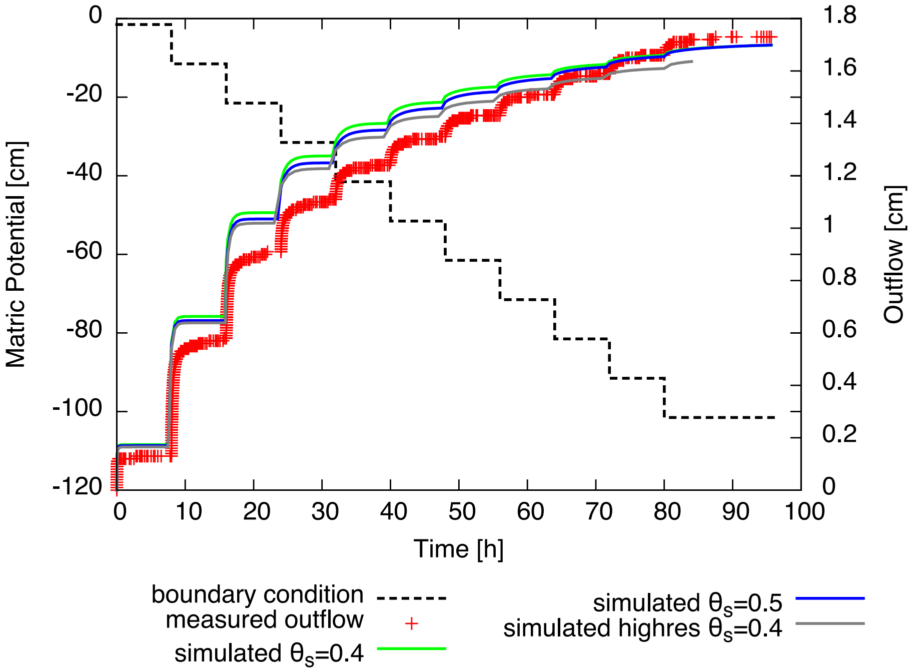

Figure 9.7.: Outflow simulated with the reconstructed parameters for matrix and aggregates with different water content and different resolution of the structure used.

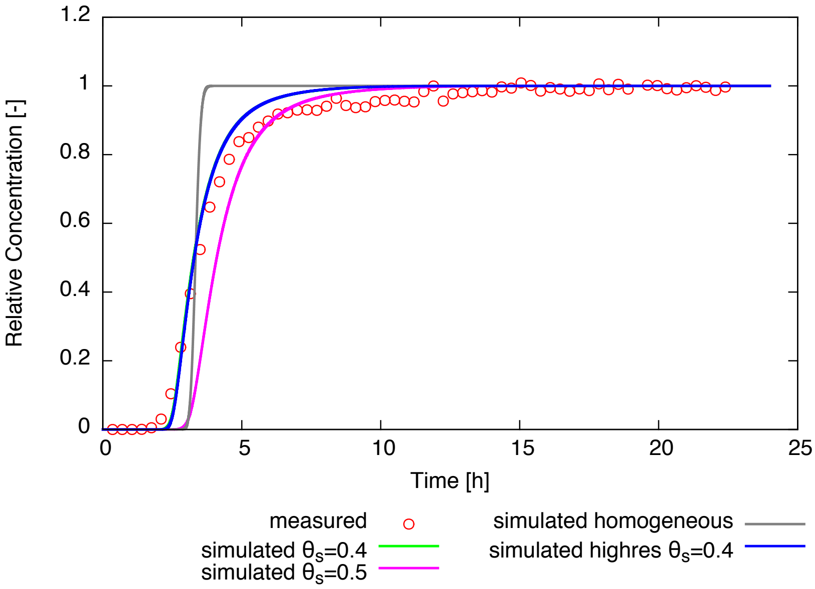

Figure 9.8.: Measured and simulated breakthrough curve for the medium-resolution structure (118 × 118 × 62 voxel) with different saturated water content, for the high-resolution structure (356 × 356 × 93 voxel) and a homogeneous sample (2 × 2 × 2000 elements). The breakthrough curves for the medium and high-resolution structure are on top of each other.

9.3.3. Solute Transport Simulation

A solute transport experiment was conducted by Kasteel et al. 2000 where first a steady-state flow with 11.4 mm/h was established. The matric potential measured inside the sample was -15 cm. Initially the sample was infiltrated with a 0.5 mmol/l solution of $CaCl_2$. After steady-state had been established it was switched to a 0.5 mmol/l solution of $CaBr_2$. Thus effects due to a change in ionic strength were avoided. The breakthrough curve resulting from the step change was measured with one sample taken every 21 minutes for about 22 hours. The concentrations were determined with ion chromatography.

The steady-state water flux field was simulated with the parallel version of $\mu\varphi$ for the reconstructed parameters with a saturated water content of 0.4 and 0.5 (table 9.8), with the medium resolution structure and with the parameters for a saturated water content of 0.4 also for the high-resolution structure. The CDE for a homogeneous material was solved as a reference with a very high resolution in the direction of transport ($2 \times 2 \times 2000$ elements). It is essential that no solute enters the background region around the sample as this would result in artificial tailing. For the background therefore a van Genuchten parametrisation with a conductivity of zero and an extremely low porosity is used. As the diffusion coefficient is proportional to $\phi^{4/3}$ the diffusion coefficient is essentially zero. A zero conductivity is possible without the Jacobian becoming singular, if the derivative of saturation with respect to matric potential is non-zero. With an initial

scenario	run-time	platform
3D parameter estimation		
low resolution	s	28 cores 2.1 GHz AMD Opteron 6172
water transport MSO		
medium resolution, $\theta_s = 0.4$	8641 s	2 cores 2.4 GHz Intel i5
medium resolution, $\theta_s = 0.5$	10432 s	2 cores 2.4 GHz Intel i5
water transport		
homogeneous	11 s	1 core 2.4 GHz Intel i5
medium resolution, $\theta_s = 0.4$	500 s	2 cores 2.4 GHz Intel i5
medium resolution, $\theta_s = 0.5$	730 s	2 cores 2.4 GHz Intel i5
medium resolution, fitted	453 s	2 cores 2.4 GHz Intel i5
medium resolution, fitted constrained	463 s	2 cores 2.4 GHz Intel i5
high resolution, $\theta_s = 0.4$	2482 s	16 cores 2.1 GHz AMD Opteron 6172
solute transport		
medium resolution, $\theta_s = 0.4$	8470 s	2 cores 2.4 GHz Intel i5
medium resolution, $\theta_s = 0.5$	8355 s	2 cores 2.4 GHz Intel i5
medium resolution, fitted	7360 s	2 cores 2.4 GHz Intel i5
medium resolution, fitted constrained	6932 s	2 cores 2.4 GHz Intel i5
high resolution, $\theta_s = 0.4$	61971 s	16 cores 2.1 GHz AMD Opteron 6172

Table 9.10.: Computation time for water and solute transport for the different scenarios of water and solute transport.

condition of $\psi_m = -15$ cm and a van Genuchten parametrisation this is guaranteed. At the upper boundary a constant flux of 11.4 mm/h is applied and a no-flow boundary condition is used where the material is background. A Dirichlet boundary condition with $\psi_m = -15$ cm is used at the bottom.

Solute transport was simulated with the parallel transient solute transport model. Given the high spatial resolution, dispersion can be neglected and the molecular diffusion coefficient of $2 \cdot 10^{-9}$ m^2/s was used. A Dirichlet boundary condition with a concentration of 1 is used at the top of the sample. The flux concentration at the bottom of the sample is calculated as quotient of the solute flux density and the volumetric water flux density. Computation times for the different simulations are given in table 9.10.

While the breakthrough with a saturated water content of 0.5 occurs too late, it nearly matches perfectly with a saturated water content of 0.4. In contrast to the solution of the CDE for homogeneous material, the shape of the breakthrough curve agrees very well with the experiment for all scenarios simulated with the soil structure, however, with a less pronounced tailing than in the experiment. As the difference between the high-resolution and the medium-resolution simulation is negligible this seems not to be an effect of under-resolved structure or of grid resolution.

parameter	initial value	3D inversion	3D inversion K_{agg} fixed
light matrix			
α [m^{-1}]	5.88	5.23 ± 0.2	5.6 ± 0.2
K [cm/h]	24.3	2.9 ± 2	5.8 ± 0.6
θ_r [–]	0.184	0.19 ± 0.02	0.206 ± 0.003
n [–]	2.83	2.5 ± 0.1	2.6 ± 0.2
aggregates			
α [m^{-1}]	0.63	0.8 ± 0.6	1.2 ± 0.5
K [cm/h]	0.279	2.7 ± 3	fixed
θ_r [–]	0.0	0.02 ± 0.2	$10^{-3} \pm 0.05$
n [–]	2.0	fixed	$K_{\mathsf{mat}}\frac{\alpha_{\mathsf{agg}}}{\alpha_{\mathsf{mat}}} = 1.2$
τ [–]	0.5		
θ_s [–]	0.4		
steps		30	6
successful steps		15	3
residual	5012	123	135

Table 9.11.: Parameters estimated with the fully three-dimensional optimisation with a seven and six parameter model.

9.3.4. Three-dimensional Inversion

A three-dimensional two-level parallel inversion was started with a low resolution version of the structure (59 × 59 × 31 voxel). With the additional elements for the plate, the forward problem had 128′797 unknowns. With permeabilities, residual saturations and α values for both materials and the n value for the matrix, seven parameters had to be estimated. The van Genuchten n for the aggregates was fixed at a value of 2. The results obtained from reconstruction were used as initial guess.

Parallel inversion was performed with 28 processes, using always 4 processes for each forward problem. The inversion reduced the residuum from 5012 to 123 (table 9.11) in

	α_{mat}	α_{agg}	K_{mat}	$\theta_{r,\mathsf{mat}}$	$\theta_{r,\mathsf{agg}}$
α_{agg}	0.964				
K_{mat}	-0.668	-0.607			
$\theta_{r,\mathsf{mat}}$	0.959	0.984	-0.551		
$\theta_{r,\mathsf{agg}}$	-0.019	0.100	-0.138	-0.063	
$n_{r,\mathsf{mat}}$	0.862	0.918	-0.435	0.963	-0.159

Table 9.12.: Correlation coefficients for the three-dimensional parameter estimation of the undisturbed soil column with the six parameter model.

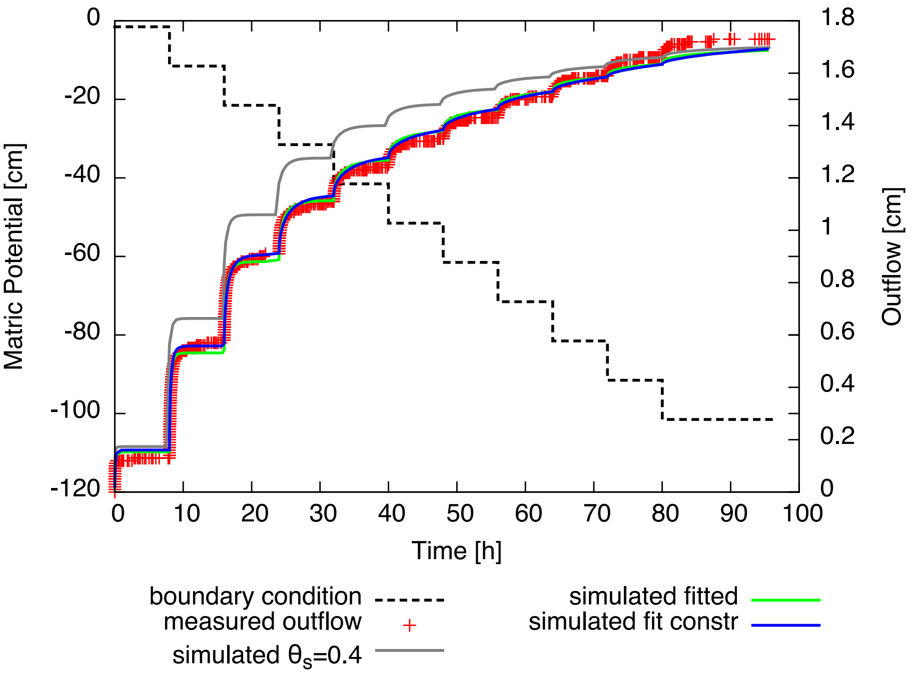

Figure 9.9.: Boundary condition, measured outflow and best fit for the three-dimensional parameter estimation of the undisturbed soil column.

30 steps, 15 of which were accepted, . One step took 15 minutes on a four processor server with 2.1 GHz AMD Opteron 6172 (Magny-Cour) twelve-core CPU's. While the model agrees well with the measured data (figure 9.9), the parameters have a high uncertainty. The saturated hydraulic conductivity is nearly the same for aggregates and matrix, which is not very realistic. Thus a new inversion was started with only six parameters. The permeability of the aggregates was fixed internally in the forward model to $K_{\mathsf{agg}} = K_{\mathsf{mat}} \frac{\alpha_{\mathsf{agg}}}{\alpha_{\mathsf{mat}}}$. This optimisation reduces the residuum in only 6 steps, 3 of which were accepted, to 135. This is only slightly larger than for the seven parameter model and the outflow curves are nearly the same (figure 9.9). However, there is still a high correlation between the two α values, the residual water content of the aggregates and the n value of the matrix (table 9.12).

The breakthrough curve simulated with the parameters from the three-dimensional optimisation with seven parameters predicts the experiment significantly worse than the breakthrough curve simulated with the reconstructed parameters from the homogeneous inversion (figure 9.10). The curve is too steep and arrives too late. The curve obtained with the six parameter optimisation gives a much better prediction of the tailing, but the front is still slower than in the experiment.

While it could be demonstrated that a three-dimensional two-level parallel inversion works in principle, the information content of the available data set is too low for a reliable estimation of the parameters. Additional information from e.g. tensiometers would be helpful, but it would also be necessary to use measurement techniques like evaporation experiments, which can reduce the potential enough to also drain the aggregates. Given

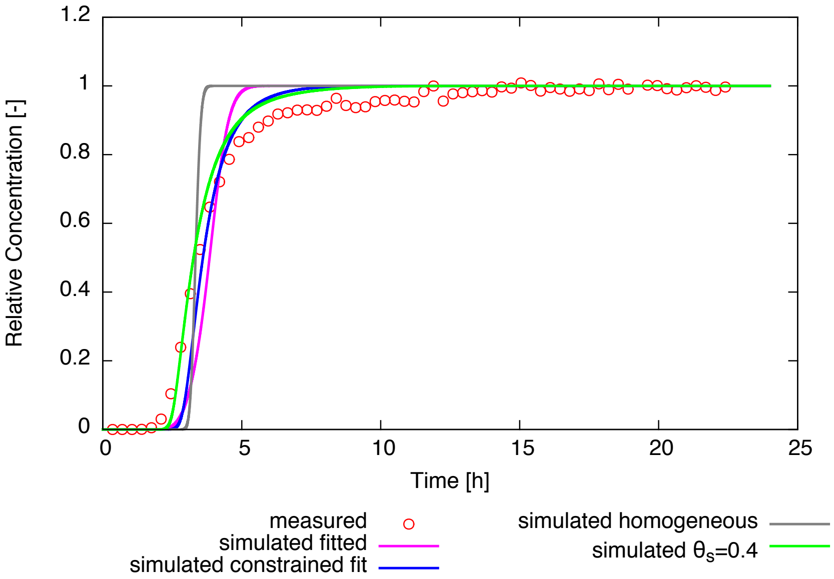

Figure 9.10.: Measured and simulated breakthrough curve for the medium-resolution structure (118 × 118 × 62 voxel) with $\theta_s = 0.4$ and the reconstructed parameters , the parameters from the three-dimensional optimisations, and a homogeneous sample (2 × 2 × 2000 elements).

the very limited available information, the prediction of the breakthrough curve with the homogeneous parameter estimation with a multiple van Genuchten model and a subsequent reconstruction of the material parameters was astonishingly good.

10. Virtual Soil Systems

10.1. Motivation

The scale of small agricultural fields (0.01–1 ha) is relevant for practical applications like precision agriculture. It is also the size of the smallest elements in regional scale modelling of catchments or in the interpretation of remote-sensing data. In such simulations each of these blocks is treated as a homogeneous porous medium. As soils are heterogeneous in reality it is an open question under which circumstances this assumption is justified. Simulations of sufficiently realistic soils could help to test this.

It also has often be observed that the transfer of measured hydraulic properties from the laboratory to the field is difficult. There are different explanations for this starting from the limited size of laboratory samples. Recent measurements by Weller et al. 2011a; Weller and Vogel 2012 indicated that hydraulic non-equilibrium and its different consequences depending on the type of forcing applied, could also be a reason for the difficulties.

One problem of field scale measurements is, that there is much less experimental control over the system than in the laboratory. Differences between simulation and observation can have various reasons:

- an insufficient precision of the performed measurements
- spatial and temporal averaging of the measurement devices especially during fast processes
- the unresolved heterogeneity of the system
- principal errors in the basic process model or the effective model used to describe the flow or transport process

It is nearly impossible to separate these reasons with real measurements. However, if the measurements would be performed on a realistic enough virtual soil system, it would be possible to quantify the influence of the first three of these points. As the virtual soil system has to be simulated with a chosen set of equations, the last point can only be tested to a certain degree. It is strongly linked to the question of the system being "realistic enough". If it behaves similar to real soil and the same problems occur already with the virtual soil system, the basic process model probably is not the main reason for the difficulties. If the virtual soil system, however, is much easier to describe effectively, the real processes are not represented precisely enough.

Within the framework of the virtual research institute "INVEST" virtual soil systems have been developed and water and solute transport has been simulated with $\mu\varphi$.

"INVEST" was funded by the Helmholtz foundation with Jan Vanderborght and Harry Vereecken (Forschungszentrum Jülich), Wolfgang Durner (TU Braunschweig), Peter Bastian and Kurt Roth (Heidelberg University) and Hans-Jörg Vogel (Helmholtz Centre for Environmental Research,UFZ) as principal investigators. Jan Vanderborght from the Institute of Bio- and Geosciences, Agrosphere at Forschungszentrum Jülich was the speaker of the virtual institute which ended in 2012. The author of this work was responsible for the forward modelling of water and solute transport.

The purpose of the virtual soil systems is to provide data which might be used to find answers to the following questions:

- Are the properties of virtual soil systems similar to real soils?
- Do effective one-dimensional models exist at the scale of small agricultural fields?
- What are suitable effective models?
- How can the effective properties of soils be determined more accurately?
- What are optimal measurement strategies?
- How can parameter estimation be optimally performed?
- How precise are the predictions?
- How large is the variability of predictions?
- How can the predictions be improved?
- What is the influence of additional measurements on the precision of the predictions?

The central assumption of the simulations is the validity of Richards' equation. The basic requirement to a virtual soil system was a resolution and problem size large enough to represent a two-scale heterogeneity consisting of a deterministic structure (horizons) with a random sub-scale heterogeneity. This requires grids with 10^6 (2D) to 10^9 (3D) degrees of freedom. The system should be simulated with a high temporal resolution of one hour to allow for virtual time-dependent measurements with a realistic resolution. Thus thousands of time steps are necessary to simulate several weeks. The three-dimensional simulations therefore require massive parallel computing with hundreds to thousands of processes. If measurements are to be performed after the costly simulations, the whole simulated data at least for potential, water content and flux field have to be stored. This requires tens of Terabyte of hard disk storage. The handling of this data also requires an efficient parallel file I/O.

After the creation of the virtual institute a comparison between $\mu\varphi$ and the Richards' equation solver ParSWMS was performed. ParSWMS (Hardelauf et al. 2007) is developed at Forschungszentrum Jülich and uses a Finite-Element discretisation in space with an implicit Euler scheme in time, a fixed-point iteration for the linearisation of the non-linear equations and PETSc for the solution of the linear equation systems. The application of both models to a heterogeneous test problem showed that depending on the number of processes ParSWMS needed a factor 13 to 16 times longer than $\mu\varphi$ to solve the test problem. This is partially caused by the overhead of using an unstructured grid, which leads to a higher memory consumption per grid cell. As $\mu\varphi$ also had a better scalability it was chosen as the model to be used for the virtual soil systems.

10.2. Water Transport

10.2.1. Soil Structure and Parameters

The virtual soil is intended to represent the features of an average agricultural soil. The selected structure resembles a Luvisol. It consist of four horizons:

- an A_{p2} horizon with tractor wheel tracks, compacted clods and a harrowed seedbed (A_{p1}) on top.
- a plough pan with holes (A_{p3})
- a marmorated B_{t}-horizon
- a sandy C_{v1}-horizon with loam lenses (C_{v2})

Different parametrisations were used for topsoil, sand, loam, plough pan and seedbed. The marmorated features, the wheel tracks and the heterogeneity of the upper horizons are realised with scaling parameters.

A structure generator was written by Hans-Jörg Vogel at UFZ Halle, which generates two TIFF images (in 2D) or ddd images (in 3D) for the material distribution and the scaling parameters. This structure generator was optimised by the author and an HDF5-output was added so that the structure and scaling image can be read by $\mu\varphi$ using parallel HDF5. Instead of unsigned char values in the TIFF and ddd-images, single precision floating point variables are used for the scaling parameters in the HDF5-file.

Different structure scenarios have been constructed for two-dimensional simulations by Steffen Schlüter (UFZ Halle). The domain size is 5×5 m. They differ by the amount of structural features included. Soil C has the most structural features including a harrowed seedbed with wheel tracks, a plough pan and holes in the plough pan. Soil B does not have wheel tracks and holes in the plough pan and soil A has neither plough pan nor seedbed. The upper and lower boundary of the plough pan have a prescribed roughness. Its shape is generated using an auto-correlated random field with a correlation length λ_{rough}. The A_{p2} and B_{t} horizon have a small-scale heterogeneity. The field of scaling parameters was also generated using auto-correlated random fields. The correlation lengths are different for the two horizons and also in horizontal and vertical direction. The shape of the clay lenses was generated by taking the threshold of another auto-correlated random field. All correlation lengths are given in table 10.2.1.

All three soils exist in a version with a flat boundary between the B_{t} and C_{v} horizon in 1.5 m depth and a version with a sinusoidal boundary between the two horizons. One period of a sinus curve was used with the maxima in 1 m depth at the domain boundary and the minimum in 2 m depth at the centre. This resulted in a set of six different scenarios in total (figure 10.1).

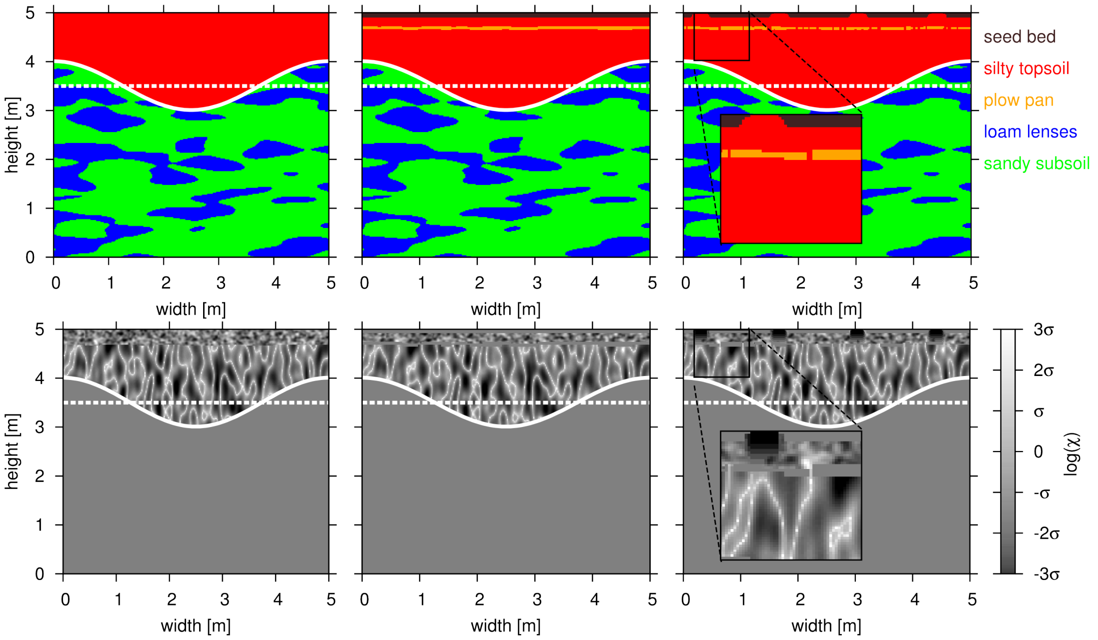

Figure 10.1.: Soil structure (top) and scaling parameter images (bottom) for the six scenarios of two-dimensional virtual soil systems: soil A (left), soil B (middle) and soil C (right) (from Schlüter et al. 2012).

For the three-dimensional simulation only soil C was used due to the high computational demand of the simulations. A larger domain size of 10 × 10 × 6.4 m was chosen to allow virtual measurements with a larger support like georadar or geoelectrics. The depth of each horizon is given in table 10.2.1.

Following a suggestion of Jan Vanderborght, the depth z of the horizon boundary was described by

$$z = 1.5 + 0.5 \cdot sin\left(4\pi\left(\frac{x}{d_x}+\frac{1}{4}\right)\right) \cdot sin\left(4\pi\left(\frac{y}{d_y}+\frac{1}{4}\right)\right). \qquad (10.1)$$

where d_x and d_y are the size of the domain in x and y-direction. The resulting boundary can be seen on the right of in figure 10.2.

The material parameters to be used for the different horizons have been intensively discussed among the members of INVEST. The parametrisations for the $\mathrm{A_{p2}}$, $\mathrm{B_t}$, $\mathrm{C_{v1}}$ and $\mathrm{C_{v2}}$ horizon where taken from the ROSETTA database (Schaap et al. 2001). Steffen Schlüter derived parametrisations for the harrowed horizon and the plough pan from the silt parameter set. The details are given in Schlüter et al. 2012. All parameters are given in table 10.2.1 and plots of the hydraulic functions are shown in figure 10.3.

It was intended to use Miller similarity scaling with the scaling parameters. However, due to a software bug which was only detected very recently, the scaling was only applied to the saturated conductivity and the soil water characteristic but not to the relative permeability function. Two-dimensional simulations with the correct Miller similarity

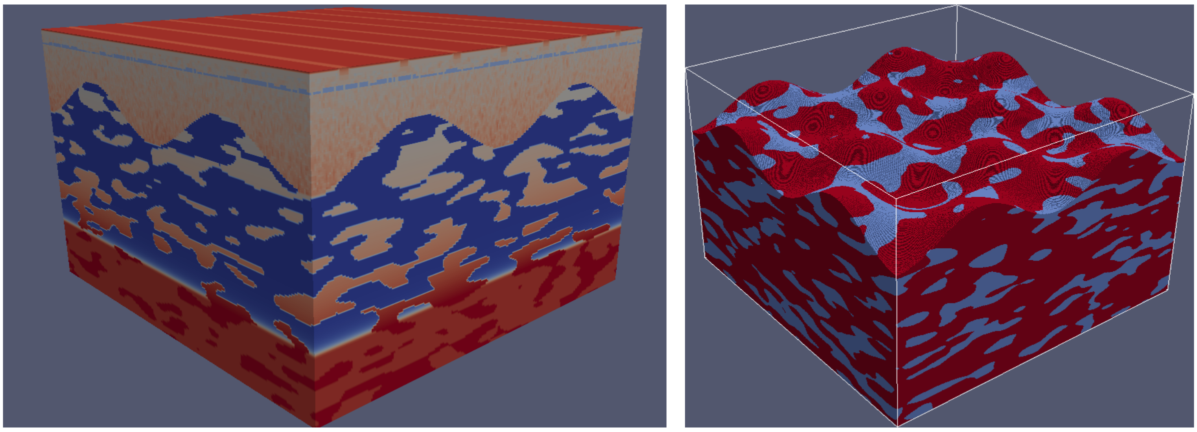

Figure 10.2.: Soil structure of the three-dimensional virtual soil systems: Water content distribution (left) and subsurface boundary with sandy subsoil (red) and clay lenses (blue).

	A_{p1}	A_{p2}	A_{p3}	B_{t}	C_{v1}	C_{v2}
	harrowed		plough pan		sand	loam
Source		ROSETTA silt		ROSETTA silt	ROSETTA sand	ROSETTA loam
Type	mmVG	mVG	mVG	mVG	VG	mVG
θ_s [–]	0.54	0.49	0.41	0.49	0.38	0.40
θ_r [–]	0.05	0.05	0.06	0.05	0.05	0.06
τ [–]	5	0.5	0.5	0.5	0.5	0.5
K_s [cm/h]	9.11	1.82	0.02	1.82	26.8	0.5
α [cm^{-1}]	0.66/20	0.66	0.387	0.66	10	1.11
n [–]	1.68/4	1.68	1.68	1.68	3.18	1.47
ψ_{m_0} [cm]	-2	-2	-2	-2	–	-2
$\sigma_{\log}$ [–]	0.5	0.5	–	0.5	–	–
depth [cm]	0 - 8	8 - 27	27 - 33	33 - 150	150 - 640	–
roughness [cm]	0	2	2	50	0	–
λ_{rough} [cm]	0	20	20	250	–	–
$\lambda_{x,y}$ [cm]	–	4	8	8	50	–
λ_z [cm]	–	4	∞	24	15	–

Table 10.1.: Soil parameters used for the virtual soil system. The possible types of parametrisation are mmVG = modified multiple van Genuchten/Mualem, mVG = modified van Genuchten/Mualem, VG = van Genuchten/Mualem.

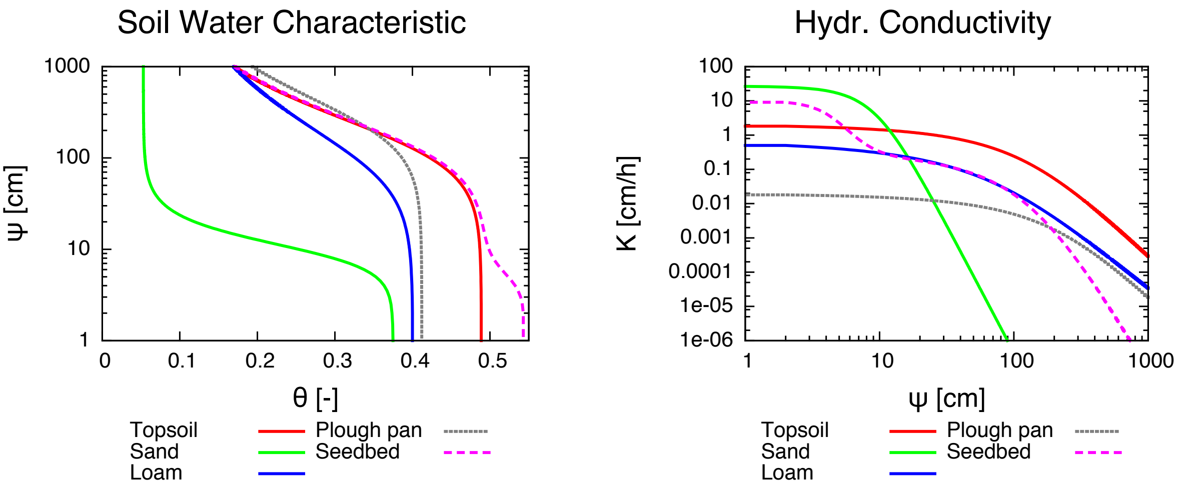

Figure 10.3.: Soil hydraulic parameters for different materials used in the virtual soil systems.

scaling show that the average fluxes calculated are nearly identical, while the local flux field can be different.

For the two-dimensional simulations the scaling parameter was calculated from the grey-scale images (figure 10.1) as described in section 4.6.4. For the three-dimensional simulations the scaling parameter was calculated from

$$\eta = \exp\left(\sigma_{\log} \cdot \chi\right). \tag{10.2}$$

with χ being the value of the normalised normal distribution read from the HDF5-file and $\sigma_{\log}$ is the standard deviation of the resulting log-normal distribution (table 10.2.1) for the material.

10.2.2. Simulation Setup

For the analysis of the system response somewhat extreme artificial weather conditions are used[1]. For the two-dimensional virtual soil systems, simulations with atmospheric boundary conditions haven been performed by Schlüter et al. 2014. Ten days of constant precipitation with a flux rate of 2 mm/h are followed by 2 days of no-flow and 30 days of evaporation with 0.4 mm/h. Limited flux boundary conditions are used for the precipitation and evaporation, where the potential at the upper boundary might never get positive during precipitation and never get more negative than -10^5 cm during evaporation. No-flow boundary conditions are used at the sides and a Dirichlet boundary condition with a constant potential of 140 cm at the bottom (table 10.2). Thus the lower boundary condition is essentially the same as for the two-dimensional simulations where a Dirichlet boundary condition with a pressure of 0 cm was used in a depth of 5 m. Hydrostatic equilibrium with a water table in 5 m depth was used as initial condition.

[1] However, similarly long periods of rain and drought occurred in Germany last year.

Boundary Conditions				
boundary	*time*	*type*	*flux density*	*potential*
top	0-10 d 42-52 d	limited flux	2 mm/h	0 cm
	10-12 d 52-54 d	no flow	0 mm/h	
	12-42 d 54-84 d	limited flux	-0.4 mm/h	-10^5 cm
bottom	Dirichlet			140 cm
sides	no flow		0 mm/h	

Table 10.2.: Boundary conditions for the three-dimensional virtual soil system simulations.

	Discretisation	
	depth	*grid spacing*
vertical	0-15 cm	1 mm
	15-50 cm	2.5 mm
	50-200 cm	5 mm
	200-640 cm	20 mm
horizontal		10 mm
time step		1 h

Table 10.3.: Discretisation used for the three-dimensional virtual soil system simulations.

As the potential gradient can be very steep during evaporation, a high spatial resolution is needed close to the soil surface. Tests with the two-dimensional soil system have shown that a resolution of 1 mm is sufficient. The element height is increased in three steps with depth (table 10.3). Thus the discretisation is second order accurate except at the three positions where the element width changes. In the horizontal direction an element width of 1 cm is used. For the given domain size this results in a system with $1024 \times 1024 \times 810 \approx 850 \cdot 10^6$ unknowns. A constant time step of one hour is used for the whole simulation. This makes it possible to perform hourly measurements on the virtual soil system. Thus 2016 time steps had to be performed.

10.2.3. Results of three-dimensional Simulations

The simulations where run on 1024 quad-core nodes of the BlueGene/P type system JUGENE at Forschungszentrum Jülich. The whole simulation required 17 days of computation time. Thus the ratio between computed time and computation time is around five. With the potential being stored as double precision values (for restarts) and the water content and flux field as single precision values, more than 40 TB of data have been produced. 27 hours of the 17 days were needed for the output of the results corresponding to a file transfer rate of 350 MB/s. As the maximal run-time per job is limited to 24 hours, the job had to be restarted several times. The time for the input of structure and scaling value thus can not be neglected, but with a total of 1.5 hours it is still comparatively low.

Surface plots of the distribution of potential and volumetric water content for six different times are given in figures 10.4 and 10.5. At the start of the simulation the soil is very dry with a potential of −500 cm at the soil surface (which is rather challenging for the numerical solver). Infiltration occurs with a rather sharp front until the plough pan is reached after 16 hours. Water is ponding on the plough pan and infiltrates through the holes. A capillary barrier effect occurs at the upper boundary of the $\mathrm{C_v}$ horizon. The water is accumulating in the troughs and enters the loam lenses until a breakthrough into the sand occurs at the tip of the troughs after 140 hours. When the precipitation stops after 240 hours the water continues to percolate through the sand in a kind of macroscopic fingers, while the flow is reversed close to the surface. At the end of the evaporation after 1008 hours the soil surface and the uppermost horizon have dried considerably. The wheel tracks are still wetter and act as a kind of wick for the evaporation. In the subsoil the water has reached a nearly uniform distribution. As the subsoil is much wetter during the second infiltration the water percolates much faster. Though the capillary barrier effect has been overcome previously the water still flows preferentially in the wetter part of the subsoil. The final distribution is nearly identical to the situation at the end of the first cycle and is thus not shown.

The ratio of actual to potential evaporation and the seepage to the groundwater are shown in figure 10.6 in a comparison with the two-dimensional simulations of the soils with the same structural features and a sinusoidal or flat boundary to the $\mathrm{C_v}$ horizon (from Schlüter et al. 2012). Evaporation is exactly the same for the first and the second cycle and thus is only shown once, while there are pronounced differences between the cycles for the seepage.

Initially all three soils are evaporating with the potential evaporation rate. In the two-dimensional simulation with the sinusoidal boundary the evaporation rate drops first, as most of the water is stored in the trough in the centre of the domain relatively far from the surface. In the two-dimensional simulation with the flat boundary to the $\mathrm{C_v}$ horizon the evaporation starts to decrease last and the evaporation rate remains consistently higher than in the other scenarios as more water is stored close to the surface. The evaporation calculated with the three-dimensional simulation stays at potential rate nearly as long as in the scenario with the flat boundary and then drops quickly to values only slightly above the values in the scenario with the sinusoidal boundary. As a consequence of the topography of the subsoil boundary (figure 10.2) the fraction of material close to the surface is higher than in the case with the sinusoidal boundary. However, after this water has evaporated or drained into the trough, the distance to the surface is very similar to the case with the sinusoidal boundary which explains the actual evaporation rate.

While the subsurface geometry is reflected very clearly and for long time spans in the evaporation rate, the seepage for the three scenarios is much more similar. While

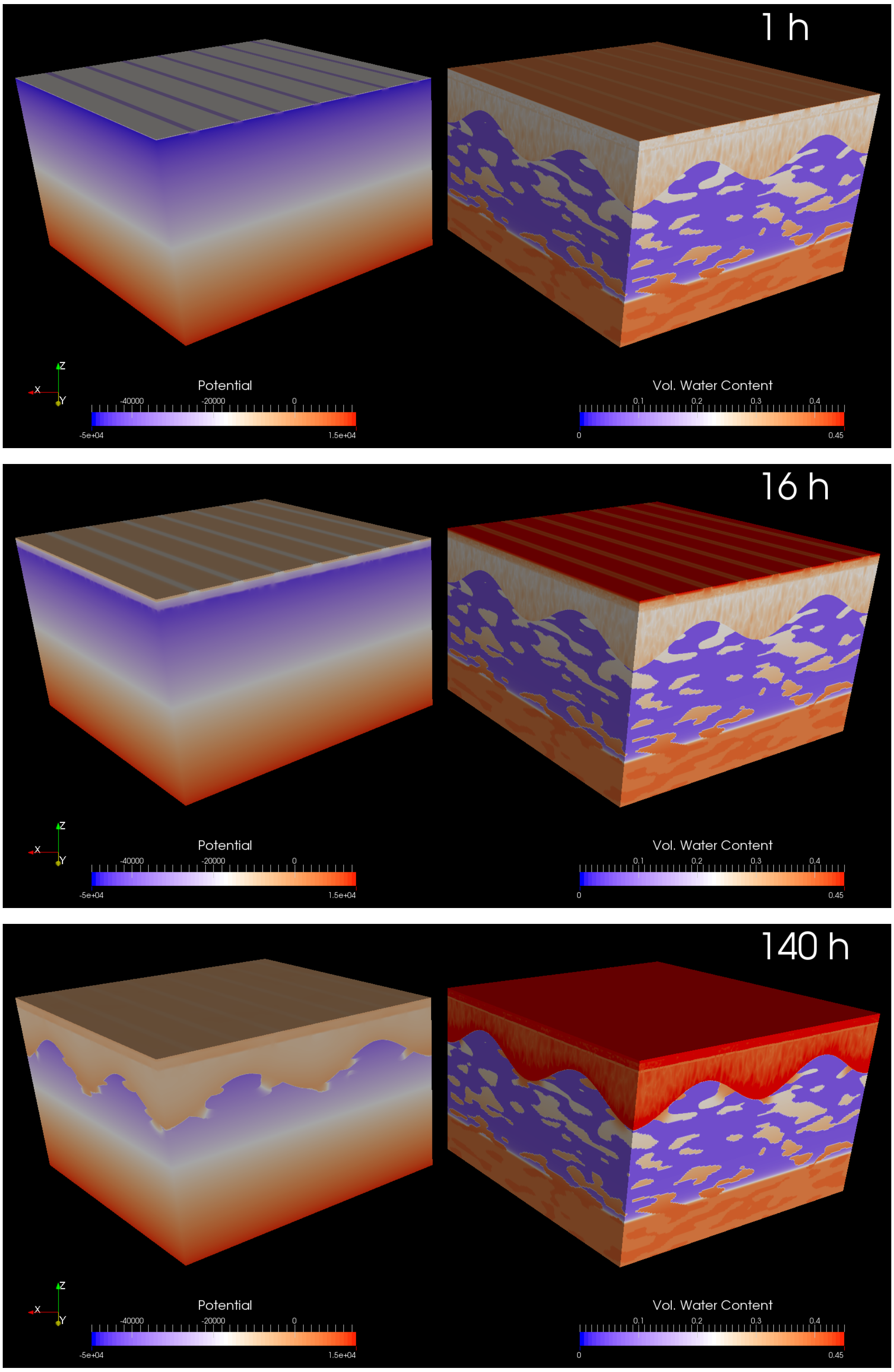

Figure 10.4.: Distribution of potential (left) and volumetric water content (right) at the start of the simulation (top), after 17 hours (middle) and after 140 hours (bottom).

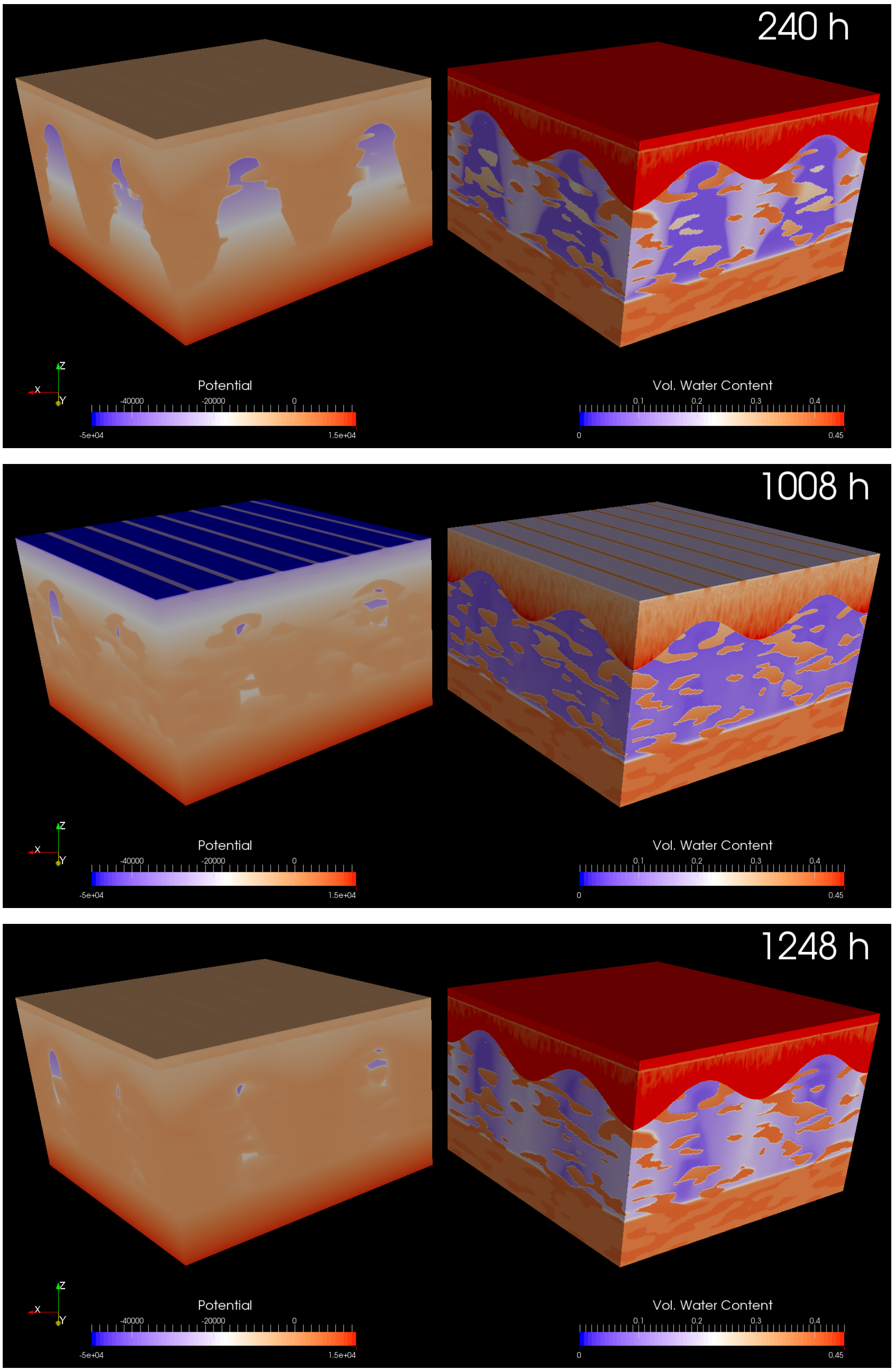

Figure 10.5.: Distribution of potential (left) and volumetric water content (right) at the end of the first infiltration period (top), at the end of the first cycle (middle) and at end the second infiltration period (bottom).

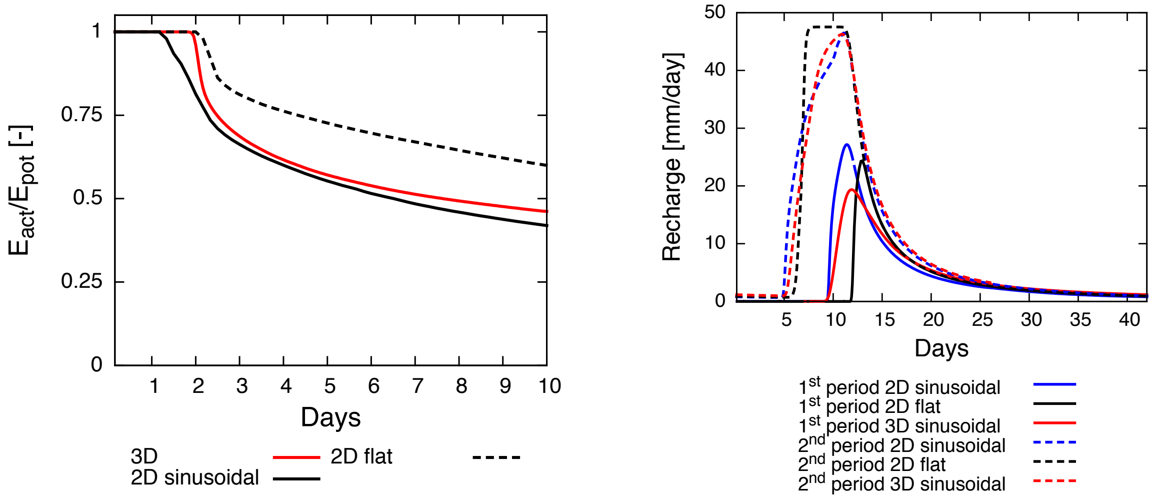

Figure 10.6.: Evaporation (left) and seepage (right) in the 2D and 3D virtual soil systems.

the onset of seepage occurs later for the scenario with a flat subsoil boundary, the seepage rates approach each other very quickly. The total amount of seepage in the three-dimensional case is smaller in the first period than for the other scenarios. This is partially due to the higher evaporation (which also shows in the later onset in the second period) but mainly it is a consequence of the fact that the water after infiltrating into the sand can be distributed more efficiently away from the finger. The onset of seepage is much earlier and the cumulative seepage is much higher in the second period for all scenarios due to the already much wetter subsoil.

One interesting result of this study is that evaporation and seepage are obviously controlled by different features of the subsurface. While the moisture content of the deep subsoil is crucial for the dynamics of the seepage, the evaporation rate is mostly controlled by the water content (and the soil structure) close to the surface. As land-surface models are often calibrated using run-off data, this might have consequences for the predicted evaporation rates.

10.3. Solute Transport

Solute transport was simulated with the water content distribution and flux fields of the second cycle. A constant concentration of 1 mmol/l was added to the infiltrating water, i.e. a Dirichlet boundary condition of 1 was used during infiltration. There was no solute in the soil initially. After the end of the precipitation a no-flow boundary condition was applied on top. A molecular diffusion coefficient of $2 \cdot 10^{-9}$ m^2/s was used. Solute transport was simulated for 42 days with the maximal time step allowed by the CFL condition using the same grid as for the water transport. The concentration and the solute flux field were written to the disk after every hour of simulated time.

10.3.0.1. Results of 3D Simulations

4096 processes on 128 nodes of the BlueGene/Q type supercomputer JUQUEEN were used for the simulation. Total simulation time for the $30'252$ time steps was 18.7 hours. Thus solute transport could be simulated much faster than water transport. Only 8.4 hours were needed for the actual calculations but 5.6 hours for the input of flux fields and volumetric water content and 2.7 hours for the output of the result. This is not a consequence of slow hard disk access but of the huge amount of data. A total of 13.7 TB of data has to be read and the same amount has to be stored. Thus the transfer rates are 680 MB/s for reading and 820 MB/s for writing.

Isosurfaces of the solute concentration are shown in figure 10.7. The initially very homogeneous infiltration front can also be seen in the solute concentration after 36 hours, as a very sharp front is moving into the soil. The holey plough pan has a more pronounced effect on the shape of the front of the solute infiltration after 100 hours than could be seen in the water content distribution. This structure is preserved for a long time as can be seen in the isosurfaces after 200 and 300 hours. The solute flow concentrates on the troughs and starts to infiltrate the sandy subsoil after 300 hours. With the onset of evaporation after 288 hours there is a flow reversal close to the surface. As the solutes remain in the soil, the solute concentration increases above 1 close to the surface and an additional (red) isosurface occurs. The solute transport essentially comes to a hold. None of the solute seems to reach the lower boundary which is also reflected by the negligible flux over the lower boundary until the very end of the simulation. As a considerable seepage was found in the water transport simulations (figure 10.6) all the water reaching the groundwater is old water, which was already in the system before the infiltration started.

The fronts of the concentration step remain sharp during the whole simulation. Thus numerical diffusion is not very high as is to be expected for a second-order scheme with such an extremely high spatial resolution.

10.4. Virtual Data Server

A special storage server with a total storage capacity of 124 TB has been acquired for the simulation results. The data is converted from the SIONlib-format used for the parallel output to a compressed HDF5-file format. This saves considerable amounts of disk space. As it is impossible to send the whole data set to any interested user, it must be possible for users to access the data remotely. This should be realised without having to create user accounts for all interested persons and without human interaction.

A web front-end has been created to allow data access over the internet. In the web-form a set of measurements can be defined. The measurements are performed by the web-server, and a link to the data is provided to the user.

Python scripts have been implemented to perform different measurements on the data. A measurement in this context is understood as the sampling of data points or sub-samples of the data from water content, potential or flux fields. A measurement also

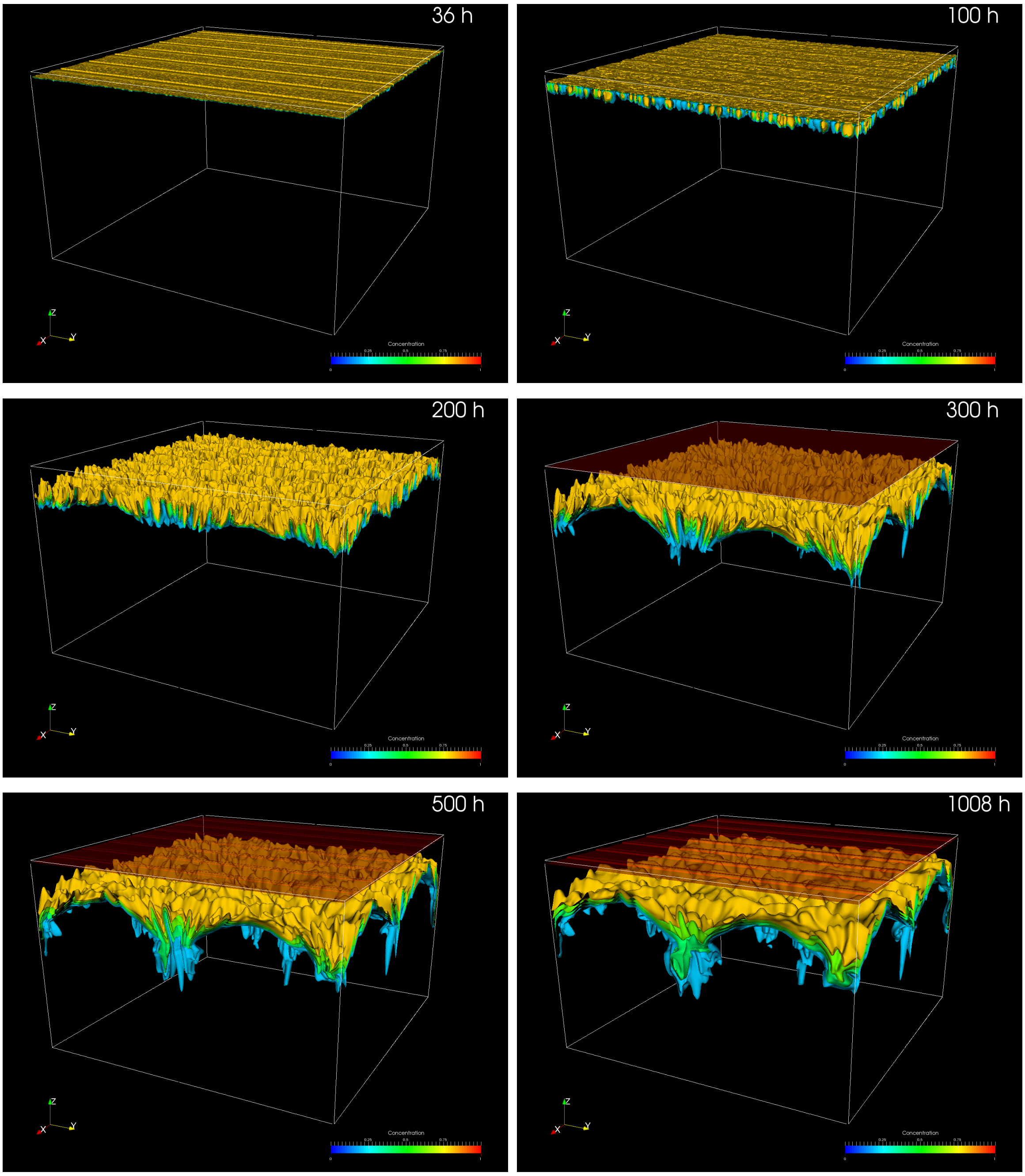

Figure 10.7.: Isosurfaces for solute concentrations of 0.25, 0.5, 0.75 and 1.0 at different times. The concentration only increases above 1.0 when evaporation starts after 288 hours and solute is accumulating in the uppermost part of the soil. As evaporation is largest in the wheel tracks they can be clearly distinguished in the last picture.

may include a spatial averaging of the data. The system is still in the setup and testing phase and will be available soon.

11. Dynamic Effects in Heterogeneous Soil

Infiltration into heterogeneous soils can be preferential, resulting in rough infiltration fronts. While small scale heterogeneities should average out, the equilibration time should be too long for larger scale heterogeneities. To study this effect, structures with a different horizontal correlation length have been used in a work published already in 2010 (Vogel et al. 2010). Different approaches for a one-dimensional effective model which is equivalent to the heterogeneous simulation have been tested. The work was mostly done by Hans-Jörg Vogel. $\mu\varphi$ was used for all simulations and some new features were added by the author.

An auto-correlated Gaussian random field with an isotropic correlation length of 8 cm was generated using the Quantim-library and stored as a grey-scale image. Log-normal scaling parameters with a standard deviation of $\sigma_{\log} = 0.25$ were calculated using this image (as described in section 4.6.4). The parameters of the van Genuchten/Mualem model used for the reference curve are given in table 11.1. Because of the software bug mentioned before, the scaling was only applied to the saturated conductivity and the soil water characteristic but not to the relative permeability function. In this case this was rather fortunate, as the full Miller-similarity scaling produces much smoother infiltration fronts.

The scaling parameter field was used to perform simulations of heterogeneous systems with different horizontal correlation lengths by just changing the horizontal size of the domain with a domain size of 2×1 m for the simulation with a horizontal correlation length of 8 cm, a domain size of 0.5×1 m for a correlation length of 2 cm and a domain size of 8×1 m for simulations with a correlation length of 24 cm.

The initial condition was hydrostatic equilibrium with a matric potential of -20 cm at the bottom, which was also the lower boundary condition. A constant flux of 3 cm/h

	K_{eff}	θ_s	θ_r	τ	α	n	$\sigma_{\log}$
	cm/h	–	–	–	cm^{-1}	–	–
reference	20	0.32	0.02	0.5	0.063	3.0	0.25
heterogeneous	24.5	0.32	0.02	0.5	0.062	2.56	–

Table 11.1.: Parameters of the reference curve used for the simulation of the heterogeneous structures with different correlation lengths and parameters obtained by fitting the effective values obtained for the heterogeneous structure.

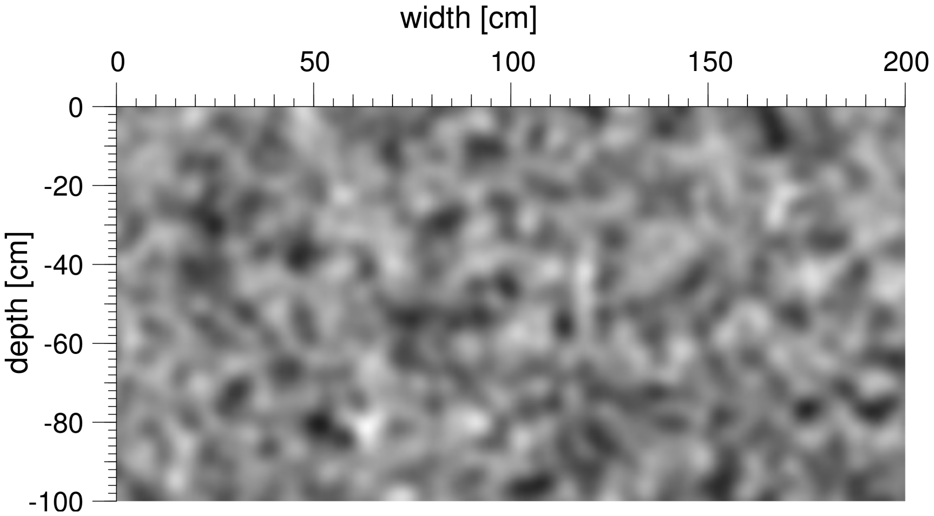

Figure 11.1.: Scaling parameter field with an isotropic correlation length of 8 cm used as base for the simulation of the heterogeneous infiltrations. A grey value of 128 yields the reference curve, darker grey values correspond to finer and lighter grey values to coarser material (from Vogel et al. 2010).

was applied at the top. A grid size of 128 × 64 elements was used for all simulations. The time step was adjusted automatically.

An example of the resulting spatial distributions of flux density, volumetric water content and matric potential for the structure with a correlation length of 8 cm after a cumulative infiltration of 66.7 mm is given in figure 11.2. The rough infiltration front and the preferential flow paths are clearly visible.

Figure 11.3 shows a comparison of the potential for the simulations with 2 and 24 cm correlation lengths also after 66.7 mm of cumulative infiltration. The results of both simulations have been scaled to the same width (please note the different horizontal scales) for easier comparability. The infiltration front for the larger correlation length is much rougher, as the time for equilibration available during the advancement of the front is too short.

To perform an effective one-dimensional simulation of the experiment, the effective hydraulic properties of the heterogeneous system have to be determined. They are *not* the same as the parameters of the reference curve. The effective soil water characteristic can be directly calculated from the soil water characteristic of the reference curve and the fraction f_i of the scaling value η_i (as they are calculated from the grey-scale image there are only 256 material classes) as

$$\theta_{\mathsf{het}}(\psi_m) = \theta \sum f_i S_{\mathsf{ref}}(\eta_i \psi_m). \tag{11.1}$$

For the random field used in this work the resulting curve is simple enough to be fitted by an ordinary van Genuchten model. The parameters of this curve are given in table 11.1. At hydrostatic equilibrium an effective curve can also be obtained by averaging over each horizontal line, where each depth corresponds to a potential. As the number of data

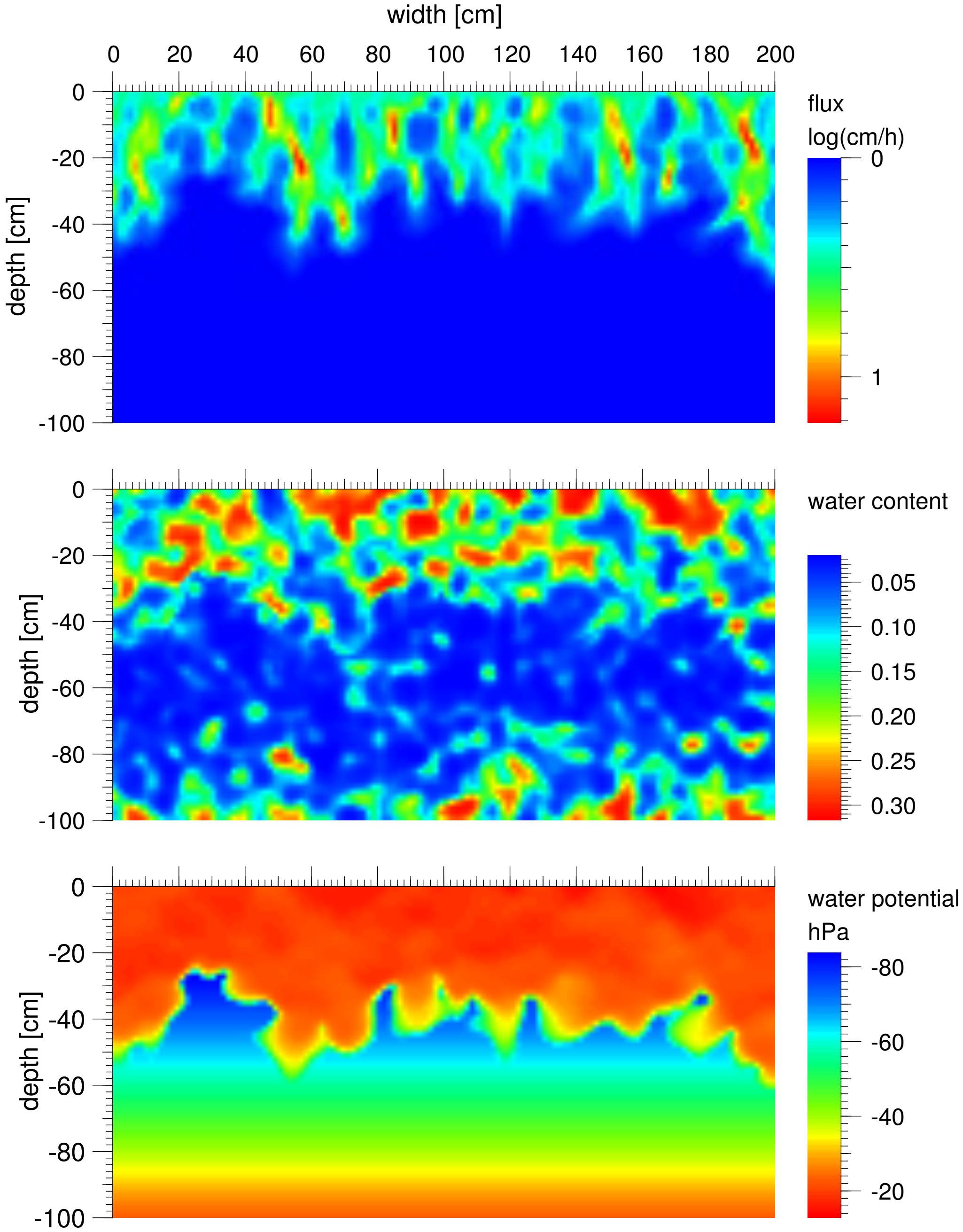

Figure 11.2.: Distribution of flux density (top), water content (middle) and potential (bottom) for the scenario with an isotropic correlation length of 8 cm after 66.7 mm of cumulative infiltration (from Vogel et al. 2010).

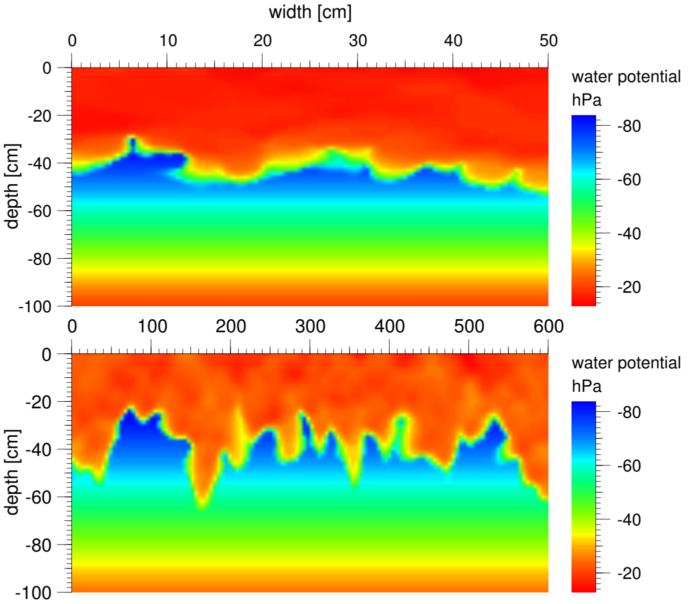

Figure 11.3.: Potential distribution for the fields with a horizontal correlation length of 2 cm (top), and 24 cm (bottom) after 66.7 mm of cumulative infiltration (from Vogel et al. 2010).

points in each line is limited, the resulting curve is a bit noisy but agrees very well with the effective curve (figure 11.4).

The relative permeability function can be calculated with the Mualem approach from the multiple van Genuchten/Mualem model or the fitted van Genuchten/Mualem model. The saturated hydraulic conductivity is calculated as a geometric mean of the saturated conductivities of the materials (Matheron et al. 1967). Alternatively the effective hydraulic conductivity can be measured by simulating gravity flow in the heterogeneous system at different flow rates yielding the hydraulic conductivity at different potentials. These values are then interpolated with cubic Hermite-splines. Both approaches produce similar values (figure 11.4) with the values from direct simulations being slightly higher. A special hydraulic function object was added to $\mu\varphi$ which combines a multiple van Genuchten model for the soil water characteristic with a spline interpolation for the relative permeability curve. For the sake of efficiency it is mandatory to use interpolation tables with this parametrisation as the evaluation of 256 van Genuchten models for every data point makes calculations prohibitively slow.

Horizontally averaged profiles of the water content after different amounts of cumulative flow into the profile are shown in figure 11.5 on the left for the simulations with 2 cm and 24 cm correlation lengths. The simulations performed with the effective soil water characteristic calculated with equation 11.1 and the spline interpolated, measured conductivities describe the average position of the infiltration front rather well. However, the infiltration fronts of the heterogeneous systems are less steep with a more disperse shape for the higher correlation length as is to be expected due the longer time needed for equilibration with a high correlation length. As this is a sign of hydraulic non-equilibrium,

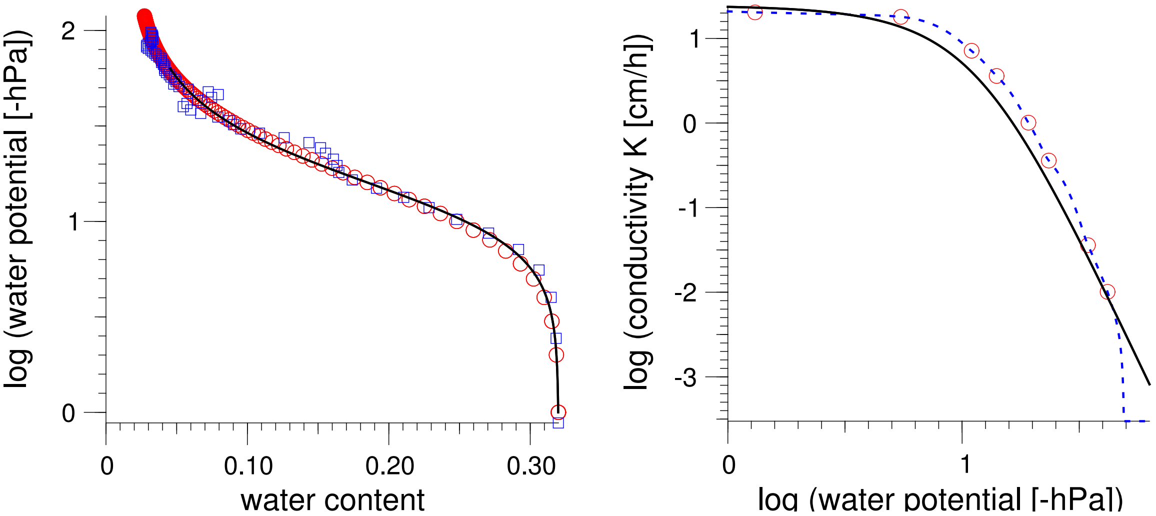

Figure 11.4.: Effective hydraulic functions derived for the heterogeneous medium. Volumetric water content (left) for the multiple van Genuchten model (red circles), line averaging during hydrostatic equilibrium (blue squares) and a van Genuchten function fitted to the red circles (black line). Conductivity function (right) derived with the Mualem model from the fitted van Genuchten curve using a saturated conductivity obtained by geometric averaging (black line) and spline interpolation (blue dashed line) of effective conductivity values calculated from stationary flow at different potentials (red dots) (from Vogel et al. 2010).

simulations of the homogeneous system with the dynamic process model of Ross and Smettem (section 3.9) and different relaxation times τ have been performed (figure 11.2, right). While they show a similar widening of the infiltration front with large values of τ, the water content in the upper part of the profile is too low especially at early times. Thus the general approach to derive effective properties is promising but still requires future research.

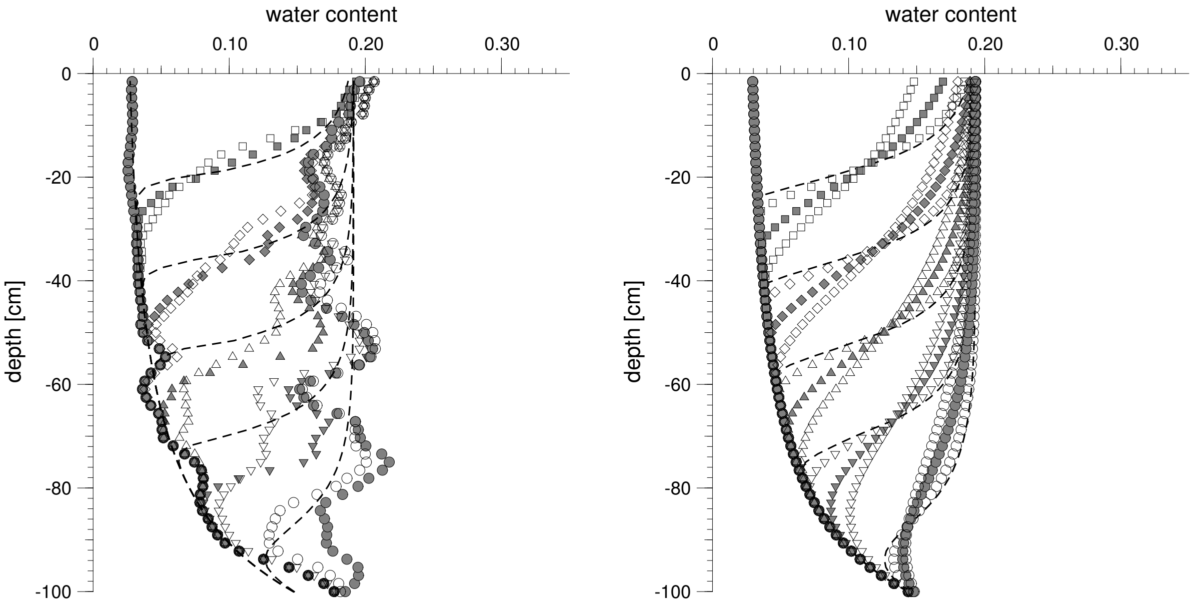

Figure 11.5.: Profiles of water content after five different amounts of cumulative infiltration. In the figure at the right line-averaged water contents from the structures with 2 cm and 24 cm horizontal correlation lengths are denoted by closed and open symbols, respectively. Profiles in the left figure are calculated with the effective properties and the dynamic model of Ross and Smettem for different relaxation times τ of 500 s (open symbols), 1200 s (closed symbols) and 2000 s (open symbols). The dashed lines in both figures are calculated with the effective parameters and an ordinary Richards' equation (from Vogel et al. 2010).

12. Large Scale Simulation of Water and Solute Transport

Applications in water management, remote sensing and weather prediction require the simulation of water transport on large spatial scales from several up to thousands of kilometres. These simulations typically have to be performed with rather large grid elements. As there is no known effective coarse scale model, Richards' equation is still used in these calculations. In one grid block, which often has a size of a hundred metre to several kilometre, very different soil types can be present and vegetation cover varies considerably from pasture over different agricultural crops up to woodland. There is also anthropogenic impact from irrigation to human settlements. All this different influences are lumped into a single set of parameters for the hydraulic functions, where the parameters are either averaged or estimated. Often just an ordinary van Genuchten/Mualem model is used. While local heterogeneity can probably be averaged and represented with effective models, even this is not a trivial task, as has been shown in the previous chapters. It is rather hard to imagine, how processes in soils which are hundreds of metres apart should produce a simple average. Due to the non-linearity of the hydraulic functions they operate on very different time scales even if they experience the same forcing by atmospheric boundary conditions. It is also quite obvious, that soils composed of very different material like sands and loams can hardly be represented by simple averaged parameters. Thus it seems reasonable to increase the grid resolution sufficiently to resolve individual units of soil types and land use. This requires massively parallel computing. However, existing codes only have a limited scalability. Parflow has been shown to scale to several thousands of processors, but the parallel efficiency dropped markedly from 4096 to 16384 processes (Kollet et al. 2010).

Weak scaling tests have been performed for $\mu\varphi$ and the transient solute transport model with different test problems and supercomputers to assess its applicability in large-scale simulations (with large-scale referring to large domain size *and* a huge number of parallel processes).

12.1. Scalability Tests for $\mu\varphi$

12.1.1. Setup

As Richards' equation is a non-linear partial differential equation, it can assume a different PDE type in different parts of the domain (section 3.2). This may also have consequences for parallel scalability. The scalable solution of the linear equation systems from a discretisation of an elliptic second order PDE is notoriously harder than the solution of a parabolic PDE. While the matrices produced by the former are only weakly diagonally dominant, the time derivative makes the matrices occurring in the latter strongly diagonally dominant with implications for the complexity of the linear solver. The soil structure used in the computations will also influence the difficulty of the problem. The case of a homogeneous material distribution and a saturated system for example is equivalent to the solution of Laplace's equation. Different test cases are needed to separate these effects.

12.1.1.1. Test Platforms

The BlueGene/P type supercomputer JUGENE at Jülich Supercomputing Center was used for the first weak scalability tests. It was operative from February 2008 to July 2012. Each of the 73'728 nodes of JUGENE had one quad-core CPU with Power-PC 450 cores and 2 GB of memory (512 MB per core). JUGENE thus had a total of 294'912 cores and 151 TB of main memory. The storage cluster of JUGENE, attached via one I/O node per 128 compute nodes, used the IBM General Parallel File System (GPFS) with a block size of 2 MB and a measured maximum usable band width of 25 GB/s.

JUQUEEN is the successor of JUGENE and has been operative since 2012. It is a BlueGene/Q type computer and has only 28'672 nodes. However, each node has now 16 compute cores (and 1 core for operating system assistance, MPI handling and asynchronous I/O). Each of the PowerPC A2 cores has a floating point unit for four double precision SIMD operations and supports quadruple hardware hyper-threading. A total of 1'835'008 threads can thus be run in parallel. As each node is equipped with 16 GB of memory each thread can have as little as 256 MB of main memory. JUQUEEN is also connected to an improved storage cluster using GPFS with 4 MB block size.

12.1.1.2. Test Cases

Two different sets of initial and boundary conditions where used in the scalability tests which enforce different PDE types. In the “elliptic” scenario stationary ground water flow was calculated, while for the “parabolic” scenario vertical infiltration in a relatively dry unsaturated porous medium was simulated. According to typical practical use cases the shape of the domains was also different for both scenarios. For the elliptic scenario a cubic domain was used, while the domain in the parabolic scenario was a large quadrilateral resembling a land surface. The domain in the parabolic scenario therefore only was scaled in the horizontal directions with a growing number of processes.

For each scenario three different test cases were performed regarding the structure of the porous medium: in the “large” test case a structure with the resolution of the whole computational grid was read by parallel file I/O. In the “block” test case the structure is the same for every process. Thus it could be read by the rank zero process and be broadcasted to all other processes. Finally in the “homogeneous” test case the same hydraulic properties were used everywhere.

parameter	θ_s	θ_r	K_s	n	α	τ
value	0.34	0.0	40.0 cm/h	2.0	5.0	0.5

Table 12.1.: Parameters for the van Genuchten/Mualem model used for the reference curve in the scalability tests.

12.1.1.3. Simulation Parameters

For the three parabolic test cases $64 \times 64 \times 128$ unknowns were used per process. Infiltration with a constant flux rate of 2 mm/h in a soil initially at hydrostatic equilibrium with a matric potential of zero at the bottom and no-flow boundary conditions at the sides was simulated. With a grid resolution of 1 cm and P processes the size of the domain thus was $\sqrt{P} \cdot 0.64 \times \sqrt{P} \cdot 0.64 \times 1.28$ m^3. The structure in the "large" and "block" test case had a two-scale heterogeneity which was created by using for every point the mean of a value from a coarse structure with a correlation length of 102 centimetre in the horizontal and 20.5 centimetre in the vertical direction (resulting in an anisotropy ratio of 5), and a value from a fine structure with an isotropic correlation length of 2 centimetre. One time step of 1 hour was simulated. A reduction of the non-linear defect by 10^{-5} was required.

80^3 unknowns per process were used for the elliptic test cases. Dirichlet boundary conditions at the left and right side of the domain were chosen to produce a horizontal pressure gradient of 1 m/m. No-flow boundary conditions were used everywhere else. A grid resolution of 10 centimetre resulted in a domain size of $(P^{1/3} \cdot 8)^3$ m^3 for P processes. A two-scale heterogeneity was used as well. The correlation lengths of the coarse structure were 10.2 metre horizontally and 2.05 metre vertically, the correlation length of the fine structure 20 centimetre. As this scenario results in a linear problem, the solution was accepted when a reduction of the linear defect by 10^{-11} had been achieved.

To generate the heterogeneity the hydraulic properties of a reference material described by a van Genuchten/Mualem model of a medium sand (table 12.1) were scaled using a scaling parameter from an auto-correlated random field. While it was intended to use Miller-similarity scaling, due to a software bug only the saturated conductivity and the soil water characteristic have been scaled and not the relative permeability function. For the elliptic scenario both approaches are completely equivalent as only the saturated conductivity is relevant. In the parabolic test cases the conductivity functions are parallel shifted, while the conductivity function in Miller-similarity intersect at some point. As the task of Miller-similarity here was only to generate challenging non-linear problems with strong heterogeneity, this does not affect the validity of the scalability tests.

12.1.1.4. Structure Generation

The auto-correlated random field of scaling parameters for the test cases was generated using a modified version of the random field generator `grf3` of the Quantim image processing library (Vogel 2008). The data type of the field values was changed to `float` instead of `double` to save memory and computation time. For the "large" test cases structures with a size up to $34'752 \times 34'752 \times 128$ for the parabolic and $5'280^3$ for the elliptic scenario were needed. As the generation of such huge structures would need excessive amounts of memory and computation time, the large structures were generated by repetition of a periodic base structure with 1024^3 values. The generation of this structure with the required two-scale heterogeneity already took 3 hours of computation time on one core of a 2.1 GHz AMD Opteron 6172. To save computation time the base structure was stored and can be read by the structure generator for later usage. For smaller structures a block with the required size was taken from the upper, left, front part of the base structure.

12.1.1.5. File I/O

As the structures to be read and the results to be written were huge, efficient file transfer was a central part of the considerations from the very beginning. Two different libraries have been used: HDF5 (The HDF Group 2000–2013) and SIONlib (Frings 2009–2013).

HDF5 allows the storage of multi-dimensional data structures and the parallel access to arbitrary sub-blocks of such structures with so-called hyperslabs. As HDF5 is a widely used file format, this has the advantage, that the data can be directly read by other programs like the visualisation tool ParaView. Additionally, data partitioning can be done directly at the start of the input operation. However, the disadvantage is that the data belonging to a single process in a parallel computation is not stored consecutively on disk but has to be reordered during input and output operations. While this can be done in a memory cache for small clusters with ample main memory, it can result in very inefficient data transfer on large super computers.

SIONlib is optimised for parallel file systems, especially GPFS. A file system block size of 1 to 4 MB is recommended with GPFS on supercomputers. This is larger than the block size of ordinary file systems. Without additional measures several processes will frequently try to access the same file system block, which results in very poor file transfer rates. SIONlib guarantees that each process gets always assigned complete file system blocks to store its data. This may result in a considerable overhead in file size if the blocks are not fully used, but can speed up file transfer considerably. SIONlib does not provide any support for the storage of structured data. Each process only gets a POSIX-file pointer from which it can read or to which it can write its data. SIONlib also allows to use a separate file for every I/O-node of JUGENE/JUQUEEN during output, which can improve write rates further. The drawback of SIONLIB-based file I/O is the need to partition the input data in advance and to post-process the output data. However, this can be done very efficiently sequentially on a high-memory compute server.

The difference between the two approaches was already visible in the preparation of the files with scaling values. The largest structure had a size of 575 GB and thus was too large to be stored in the 128 GB of main memory of the available compute server. The HDF5-file thus had to be written with hyperslabs. This slowed down the output operations markedly. Nearly 24 hours were needed to write the 575 GB file. With

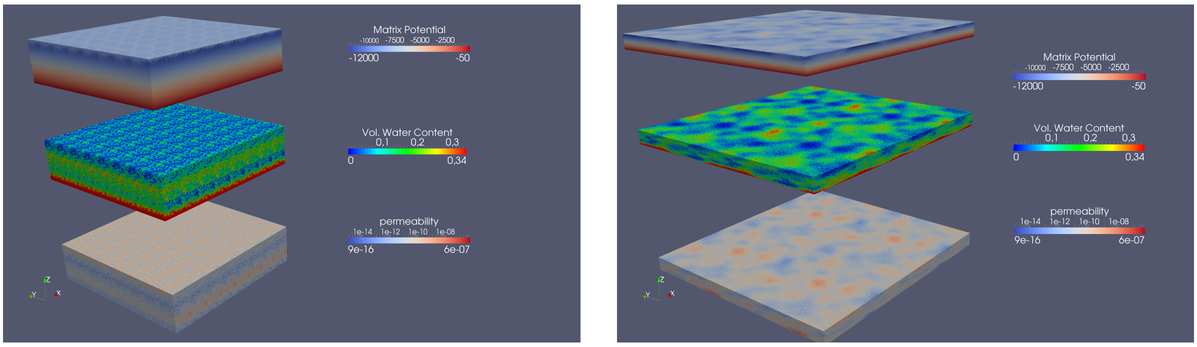

Figure 12.1.: Matric potential [Pascal] (top), volumetric water content [–] (middle) and conductivity [m/s] (bottom) for the parabolic scenario with a block structure with 64 processes (left) and a large structure with 1024 processes (right).

SIONlib the data for the individual processes is always written one after the other. Thus the transfer rate remained high even for the very large files.

12.1.2. Results on JUGENE

12.1.2.1. Parabolic Scenario

Figure 12.1 shows the conductivity field, the water content and the matric potential after a single time step for the block structure with 8×8 processes and the large structure with 32×32 processes. Especially in the water content it is clearly visible that the structure in the block test case is the same for each process and that there is no smooth transition between processes. For the large test case the full structure is first resolved with 16×16 processes. It can be seen in the figure how the structure repeats periodically twice in every direction. It is also important to note that the conductivity varies over 9 orders of magnitude at the surface for the large scenario, but the variation is – as a consequence of the structure size – smaller for the block scenario which looks almost macroscopically homogeneous.

The run-time without I/O (figure 12.2, left) is smallest for the homogeneous test case as it requires less Newton iterations, less linear iterations (table 12.2) and the times required to calculate the non-linear defect, assemble the Jacobian, build the coarse grid matrices and apply one step of the linear solver are smaller than for the two heterogeneous test cases. There is a marked increase in the computation time when the number of processes is increased from four to eight as for the larger grids five newton iterations are required instead of four. The total computation time stays nearly constant afterwards. The calculation of the non-linear defect, the assembly of the Jacobian and the application of one step of the linear solver scale nearly perfectly up to 16’384 processes with an efficiency well above 90 per cent (figure 12.2, right). Only the coarsening step, where the coarse grid hierarchy of the AMG solver is assembled shows a more pronounced drop in the parallel efficiency from 1 to 16 processes. As the efficiency stays nearly constant

		homogeneous			*block*			*large*		
		iterations		defect	iterations		defect	iterations		defect
P	unknowns	Newton	linear	eval.	Newton	linear	eval.	Newton	linear	eval.
1	$524 \cdot 10^6$	4	9	8	5	20	9	5	20	9
4	$2.10 \cdot 10^6$	4	11	8	5	20	9	7	28	11
16	$8.39 \cdot 10^6$	5	12	9	5	20	9	6	29	13
64	$33.6 \cdot 10^6$	5	12	9	5	20	9	12	57	36
256	$134 \cdot 10^6$	5	13	9	5	20	9	12	60	36
1'024	$537 \cdot 10^6$	5	13	9	5	21	9	12	61	36
4'096	$2.15 \cdot 10^9$	5	14	9	5	21	9	12	62	36
294'849	$155 \cdot 10^9$							12	63	36

Table 12.2.: Number of unknowns, non-linear and linear iterations and defect evaluations for the different test cases of the parabolic scenario.

	homogeneous				*block*				*large*			
P	defect	matrix	build	t_{it}	defect	matrix	build	t_{it}	defect	matrix	build	t_{it}
1	8.7	35.3	18.2	4.7	9.9	39.8	20.8	4.9	9.9	39.7	20.7	4.9
4	8.9	36.0	21.0	4.7	10.1	40.6	23.8	5.4	10.2	41.0	24.5	5.4
16	8.9	36.1	23.8	4.9	10.3	41.4	26.9	5.8	10.3	41.2	27.7	5.6
256	8.8	36.0	24.0	4.9	10.3	41.5	26.9	5.8	10.2	41.1	29.5	5.7
1024	8.8	35.9	24.1	4.8	10.3	41.2	27.1	5.8	10.3	41.1	29.6	5.7
4096	8.9	36.0	24.0	4.8	10.3	41.4	27.1	5.6	10.3	41.3	29.5	5.7
16'384	8.9	36.1	23.9	4.8	10.3	41.4	27.2	5.6	10.3	41.3	29.5	5.6
294'849									11.3	45.1	30.5	5.7

Table 12.3.: Computation times for the calculation of the non-linear defect, the assembly of the Jacobian, the setup of the coarse grid hierarchy and the application of one step of the linear solver (t_{it}) for the different test cases of the parabolic scenario.

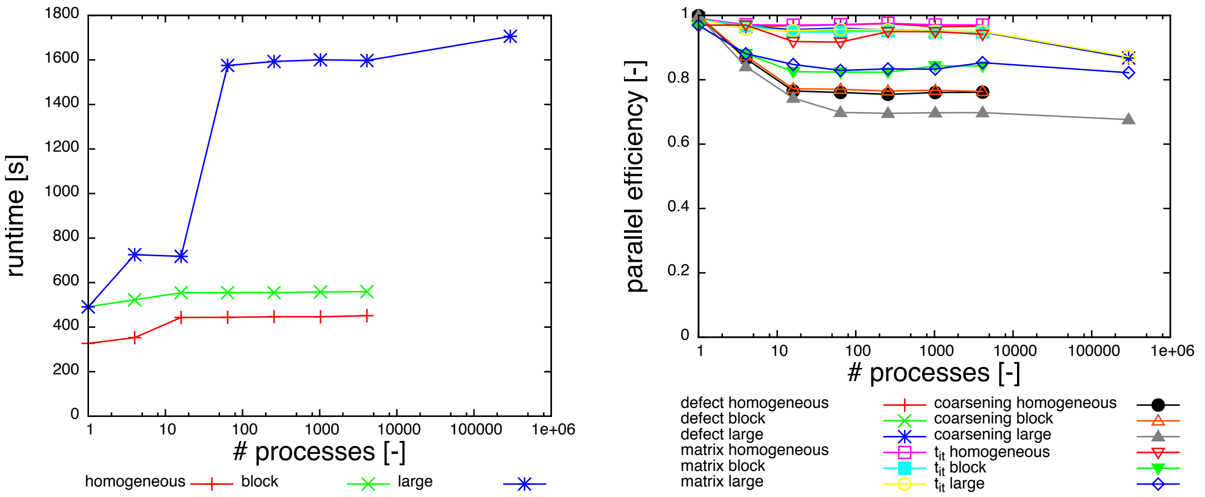

Figure 12.2.: Run-time without I/O (left) and parallel efficiency of the components (right) for the three test cases of the parabolic scenario with up to 294’849 processes. t_{it} is the time for one iteration of the linear solver.

afterwards this is not problematic. If the efficiency would be calculated relative to 16 or 64 processes as often done in the literature, the scaling would still be nearly perfect. While the number of iterations should be constant for a strictly diagonally dominant problem, there is an increase from 9 to 15 iterations.

The behaviour of the block test case is very similar. The total computation time is higher as nearly twice as many linear iterations are needed to solve the more complex problem. In contrast to the homogeneous test case the number of linear iterations stays nearly constant with increasing problem size. Probably due to more complex coarse grid problems the time for the application of one step of the linear solver increases slightly during the transition from sequential to parallel but stays constant afterwards.

The parallel run-time for the large test case increases by a factor of 3.5 from 1 to 64 processes and stays nearly constant afterwards. This is a consequence of an increase in the number of non-linear and above all the linear iterations. The number of defect evaluations rises as well, as line-search steps in the Newton scheme are necessary to achieve convergence. With an increase of the domain size more and more of the large structure (figure 12.1) is taken into account. The structure is fully resolved for the first time with 256 processes. The problem is thus becoming more heterogeneous with the increase in P and therefore more difficult to solve. When most of the structure is included in the simulations, the computation time stays essentially constant. As in the other test cases the parallel efficiency of the individual components is excellent even up to 294’849 processes.

For the parabolic test case, $\mu\varphi$ and the ISTL-AMG show an excellent scalability with the number of processes and a very good complexity in the number of unknowns as reflected in the nearly constant number of linear iterations.

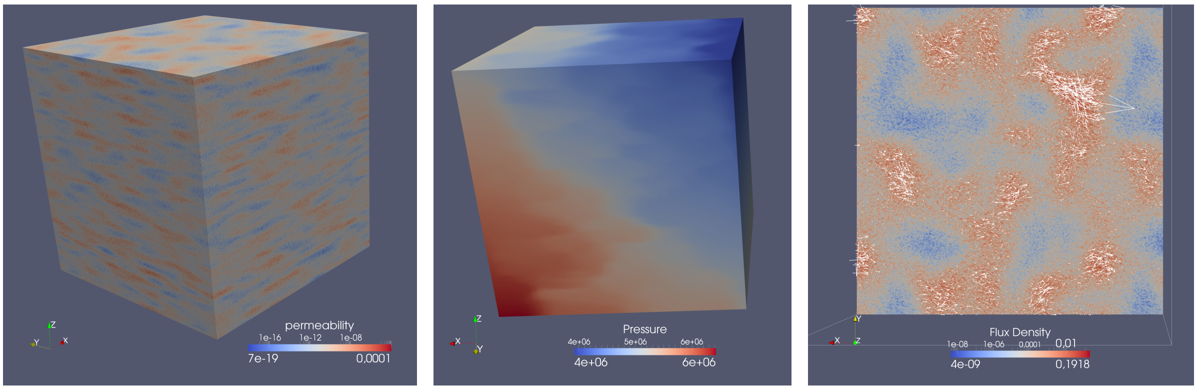

Figure 12.3.: Surface plot of the saturated hydraulic conductivity [m/s] (left), pressure distribution [Pa] (middle) and horizontal cross-section of the flux density in the middle of the domain (right), where colours show the euclidean norm of the flux density and glyphs the flux density vectors, for the large test case of the elliptic scenario and 4096 processes.

12.1.2.2. Elliptic Scenario

The structure used in the elliptic scenario with 4096 processes can be seen in a surface plot of the saturated hydraulic conductivity, (figure 12.3, left). The high variation of the conductivity which varied by 14 orders of magnitude resulted in a correspondingly heterogeneous distribution of pressure (figure 12.3, middle) and flux density (figure 12.3, right).

Whereas the parallel run-time stays nearly constant for the parabolic test case a linear increase can be seen in the computation time for the homogeneous and block test cases of the elliptic scenario (figure 12.5, left). The main reason for this increase is a linear increase in the number of iterations needed for the solution of the linear equation system (table 12.4, figure 12.4). This is to be expected for the agglomeration AMG used in the calculations and is responsible for its $\mathcal{O}(N \log(N))$ complexity. For the large test case the increase in the number of iterations does no longer seem to be linear. In the elliptic scenario the domain grows in all three-dimensions with an increasing number of processes. The full structure is therefore resolved for the first time with $16^3 = 4096$ processes. While this could explain the above-linear increase of the number of iterations up to 4096 processes, it does not explain the increase from 4096 to 287'496 processes.

In the (linear) elliptic test case it is not necessary to evaluate the non-linear relative permeability function and the soil water characteristic. Thus the time for the calculation of the defect and the assembly of the Jacobian is only about 50 % of the time needed in the parabolic test case while the number of unknowns per process is nearly the same (table 12.3). The calculation of the defect, assembly of the matrix and the application of one iteration of the linear solver have a parallel efficiency very similar to the parabolic scenario (figure 12.5, right). The time for one solver step is initially significantly lower in the homogeneous test case. With increasing problem size it approaches the time needed

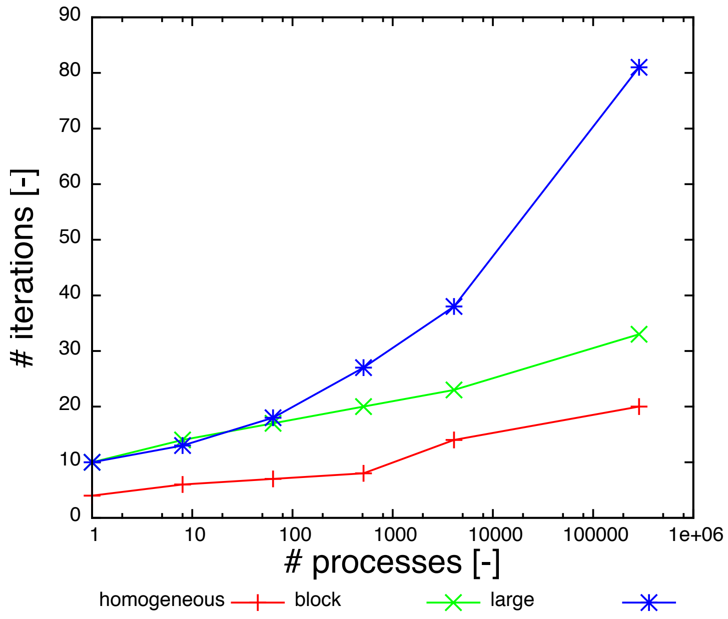

Figure 12.4.: Linear iterations for the elliptic scenario with up to 287’496 processes for the three test cases.

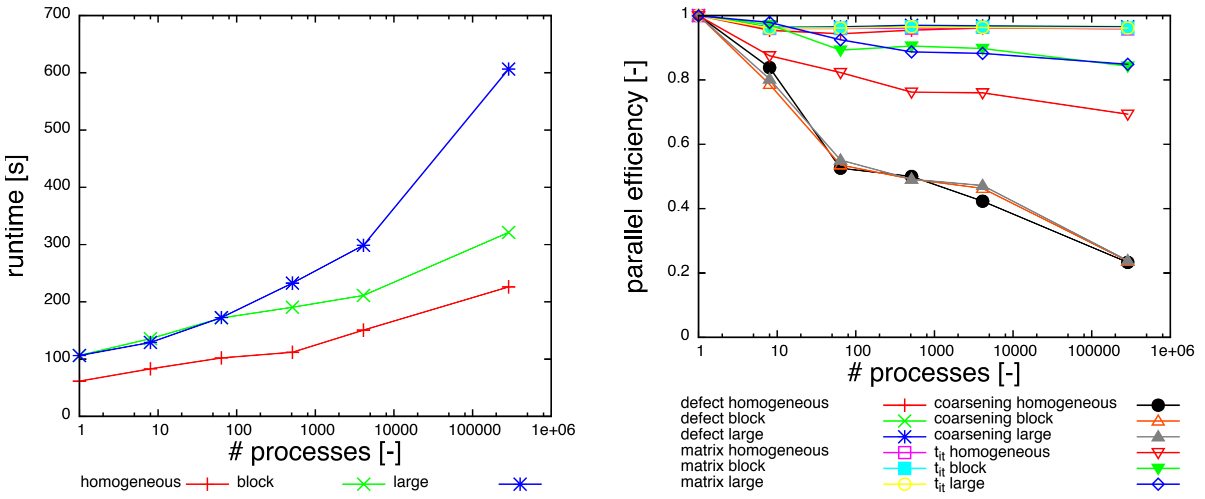

Figure 12.5.: Run-time without I/O (left) and parallel efficiency of the components (right) for the three test cases of the elliptic scenario with up to 287’496 processes. t_{it} is the time for one iteration of the linear solver.

		linear iterations		
P	unknowns	*homogeneous*	*block*	*large*
1	$512 \cdot 10^3$	4	10	10
8	$4.10 \cdot 10^6$	6	14	13
64	$32.8 \cdot 10^6$	7	17	18
512	$262 \cdot 10^6$	8	20	27
4096	$2.10 \cdot 10^9$	14	23	38
287'496	$147 \cdot 10^9$	20	33	81

Table 12.4.: Number of unknowns and linear iterations for the different test cases of the elliptic scenario.

	homogeneous				*block*				*large*			
P	defect	matrix	build	t_{it}	defect	matrix	build	t_{it}	defect	matrix	build	t_{it}
1	4.7	17.8	16.8	4.1	5.2	18.8	19.9	5.1	5.2	18.8	19.8	5.1
8	5.0	18.6	20.0	4.7	5.4	19.6	25.3	5.3	5.4	19.6	24.6	5.2
64	5.0	18.6	31.9	5.0	5.4	19.6	37.1	5.7	5.4	19.6	35.9	5.5
512	5.0	18.6	33.5	5.4	5.3	19.5	40.5	5.6	5.3	19.5	40.2	5.7
4096	4.9	18.5	39.6	5.4	5.3	19.6	43.0	5.7	5.3	19.6	41.9	5.8
287'496	4.9	18.6	71.9	5.9	5.4	19.6	84.0	6.1	5.4	19.6	82.8	6.0

Table 12.5.: Computation times for the calculation of the non-linear defect, the assembly of the Jacobian, the setup of the coarse grid hierarchy and the application of one step of the linear solver (t_{it}) for the test cases of the elliptic scenario.

in the other test cases. This results in a seemingly lower parallel efficiency.

The only part of the program which is not scaling very well is the coarsening. There is a fundamental challenge when applying our type of agglomeration AMG to elliptic problems in a massively parallel environment. Initially the coarsening is performed by aggregation of unknowns on each process. If this works perfectly, there is one unknown left for each process on the coarsest grid. Thus the coarse grid matrix has a dimension of at least P. If the coarse grid problem is solved iteratively this might result in slow convergence on large-scale computers like JUGENE. This was not a problem for the parabolic scenario as for this case it was sufficient to perform some smoothing on the coarse grid to obtain good convergence. For elliptic problems, however, it is necessary to solve the coarse grid problem exactly. Thus if the number of unknowns on each process drops below a certain threshold, the coarse matrix is redistributed to fewer processes. For small clusters it is possible to directly redistribute the matrix to a single process, which performs the further coarsening and the LU-decomposition of the coarsest matrix, as soon as it is small enough, while the other processes fall idle. However, this approach will not scale to large numbers of processes. An alternative is the successive repartitioning of the matrix to fewer and fewer processes. To reduce the amount of communication a (semi-)optimal partitioning has to be found. This is a graph-partitioning problem and

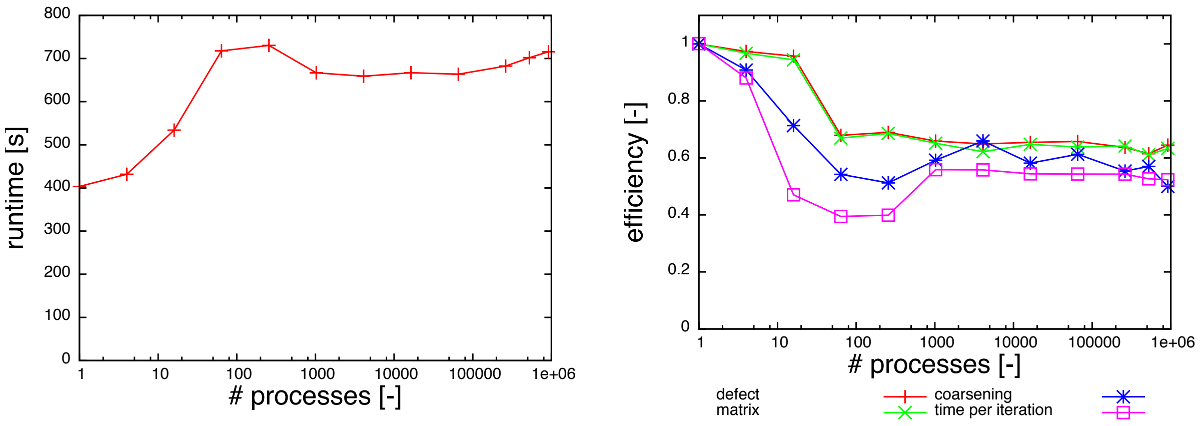

Figure 12.6.: Run time for the parabolic test case with up to 917’504 processes for the three scenarios.

can be solved with libraries like METIS or ParMETIS. A first implementation solving the partitioning problem for the full coarse grid matrix with ParMETIS did not work on JUGENE, as ParMETIS tried to allocate a $P \times P$ matrix which does not fit into the memory of a compute node. A working alternative was obtained by sending only the connectivity pattern of whole processes to the rank zero process and then solving the partitioning problem sequentially with METIS. While this works, it involves an all-to-one communication, which seems to be responsible for the bad scalability of the coarsening. This problem is currently addressed in a project within the framework of the German Priority Programme “Software for Exascale Computing”.

Despite the fact that the coarsening still has to be improved, $\mu\varphi$ and the ISTL-AMG scaled reasonably well also for the elliptic scenario and produced solutions for problems with nearly 150 billion unknowns in a reasonable time of 200 to 600 seconds.

12.1.3. Results on JUQUEEN

With JUQUEEN, the successor of JUGENE, it is possible to use even more parallel processes. JUQUEEN has 524’288 cores but as it also supports 4-way hyper-threading, up to 1’835’008 threads can be started. An interesting question is the efficiency of the hyper-threading. To test this, a simulation of the parabolic block test case with 4096 processes was started either on 256 compute nodes without hyper-threading or on 128 compute nodes with 2-fold hyper-threading. The computation time without I/O increased from 508 seconds with 256 nodes to 701 seconds with 128 nodes. However, as the accounting of computation time is done by multiplication of the number of nodes with the time they were used, it is still cheaper with hyper-threading. A more thorough testing of the hyper-threading was performed with the solute transport code (section 12.2.2.1).

For the scalability test on JUQUEEN only the block scenario was used, as it allows to assess the parallel efficiency of the components and is not trivial, while it is not too costly in computation time. By using two-fold hyper-threading, the weak scaling test

		iterations		defect	time per step			
P	unknowns	Newton	linear	eval.	defect	matrix	build	t_{it}
1	$524 \cdot 10^6$	5	20	9	8.2	32.3	20.3	3.3
4	$2.10 \cdot 10^6$	5	20	9	8.4	33.4	22.3	3.7
16	$8.39 \cdot 10^6$	5	20	9	8.5	34.2	28.4	7.0
64	$33.6 \cdot 10^6$	5	20	9	12.0	48.2	37.4	8.4
256	$134 \cdot 10^6$	5	21	9	11.9	47.1	39.6	8.3
1024	$537 \cdot 10^6$	5	21	9	12.4	49.5	34.2	5.9
4096	$2.15 \cdot 10^9$	5	21	9	12.6	51.9	30.7	5.9
16'384	$8.59 \cdot 10^9$	5	20	9	12.5	49.8	34.8	6.1
65'536	$34.4 \cdot 10^9$	5	20	9	12.4	50.6	33.1	6.1
262'144	$137 \cdot 10^9$	5	20	9	12.8	50.3	36.6	6.1
524'288	$275 \cdot 10^9$	5	20	9	13.3	52.8	35.5	6.3
917'504	$481 \cdot 10^9$	5	20	9	12.7	50.9	40.5	6.3

Table 12.6.: Number of unknowns, non-linear and linear iterations and defect calculations, as well as computation times for the calculation of the non-linear defect, the assembly of the Jacobian, the setup of the coarse grid hierarchy and the application of one step of the linear solver (t_{it}) for the block test case of the parabolic scenario on JUQUEEN.

could be performed with up to 917'504 processes and 481 billion unknowns.

After an initial increase the parallel run-time without I/O (figure 12.6, left) stays more or less constant up to 917'504 processes. With 64 processes the 2-fold hyper-threading was used for the first time, which partially explains the massive increase in run-time from 16 to 64 processes. This is reflected in a corresponding drop in the parallel efficiency of the program components (figure 12.6, right). However, a close analysis of the absolute times needed for different components of the code (table 12.6) shows, that there has to be an additional reason. While the time needed for the calculation of the defect and the assembly of the Jacobian stays more or less the same from one to sixteen processes, increases by 40% on 64 processes and then remains again nearly constant, the time for one step of the linear solver already increases considerably with 16 processes and rises to 2.5 times the original value with 64 and 256 processes. t_{it} then drops to 1.8 times the sequential value with 1024 processes and afterwards remains there up to 917'504 processes. For the coarsening the situation is similar but less pronounced. The jobs up to 256 processes are started on a different partition than the larger jobs, which suggests that there is a technical reason for this effect. Up to know even after consultation of the technical staff in Jülich it was not possible to find a detailed explanation. The number of linear iterations was the same as in the scalability test on JUGENE and was nearly constant as well. The run-time with hyper-threading was only a little bit higher than the run-time on JUGENE which is a result of the faster processor cores on JUQUEEN. However, as the problem is bounded by memory bandwidth, the 4-fold SIMD unit could not be exploited.

As 917’504 and 524’288 are no square numbers, the rectangular grids with a size of $57344 \times 65536 \times 128$ and $32768 \times 57344 \times 128$ were used for the two largest simulations. For all other simulations the grid was quadratic in the horizontal. While the grid was always partitioned in the horizontal only, the partitioner chose a partition of $512 \times 896 \times 2$ processes for the full machine run. This also resulted in a different shape of the heterogeneity as the same structure was now stretched to $0.64 \times 1.28 \times 0.64$ metres instead of $0.64 \times 0.64 \times 1.28$ metres. Neither the change of the domain shape nor of the partitioning did significantly affect the simulation time, which is an indicator for the reliability of the scalability data.

With this data $\mu\varphi$ was shown to scale up to the full machine of JUQUEEN and was admitted as ninth program into the High-Q club of best scaling software on JUQUEEN.

12.2. Scalability Tests for Solute Transport

12.2.1. Setup

The flux fields produced by the scalability test of $\mu\varphi$ on JUQUEEN were used to simulate solute transport and assess the weak scalability of the transient solute transport model. A Dirichlet boundary condition of 1 was used at the upper boundary. No-flow boundary conditions are used at the sides and for the diffusive flux at the bottom. The same grid and domain decomposition as in the flow calculations was used for the solute transport as well.

12.2.2. Results on JUQUEEN

12.2.2.1. Hyperthreading

For a more thorough testing of the efficiency of the hyper-threading more thoroughly, solute transport was simulated with the flow field calculated with 4096 processes. The number of threads per node was increased from 4 to 64 while the number of compute nodes was decreased from 1024 to 64. The rise of the parallel run-time from 4 to 16 threads (figure 12.7, left) is caused solely by an increase in the time needed for file I/O while the actual computation time stays constant. With less compute nodes there are also less I/O nodes available and thus the bandwidth is reduced. There is a slight increase in the computation time with 2-fold hyper-threading and a marked increase with 4-fold hyper-threading which is partially compensated by a decrease of the time needed for file I/O. The small jobs are run on a partition with more I/O nodes per compute node and thus file transfer is more efficient. The 10 % increase in the time needed for the actual calculations from 16 to 32 threads is much smaller than the 40 % increase measured for $\mu\varphi$. This is unexpected as an explicit time-stepping scheme should more likely be memory bandwidth-limited, but on the other hand shows that the measures taken to make the calculations vectorisable and to optimise the memory layout have been successful. There is a 63 % increase in calculation time from 32 to 64 threads

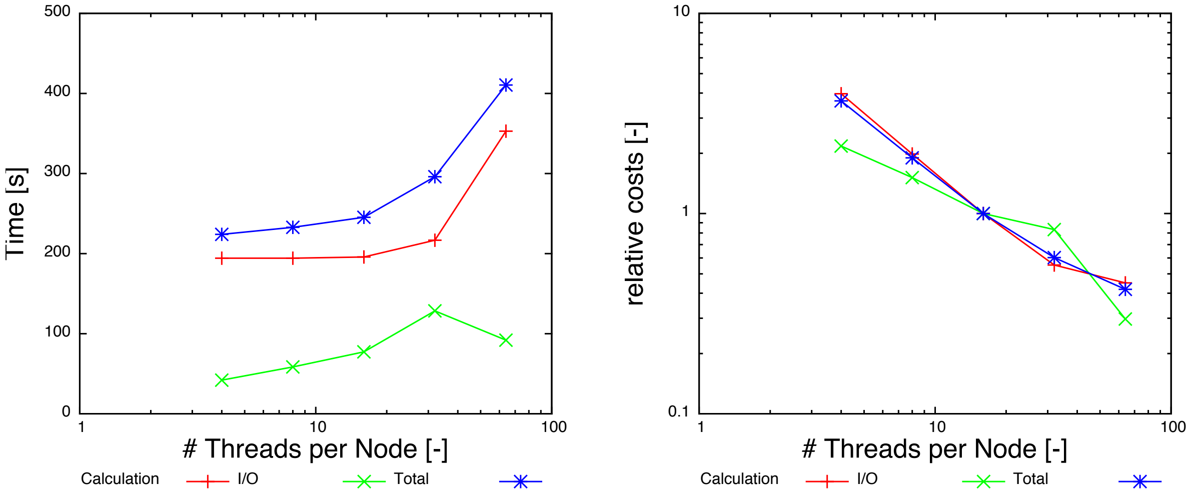

Figure 12.7.: Change of computation time (left) and the relative computational costs (right) with the number of threads per node while keeping the total problem size constant.

and an 80 % increase from 16 to 64 threads. Thus 4-fold hyper-threading is still most efficient in terms of relative computational costs (figure 12.7, right), defined as the time needed with X-threads compared to the time needed with 16 threads. While the use of 64 threads thus is optimal for solute transport, 32 threads seem to be more reasonable for the water transport. As the scalability test of the water and solute transport had to be run in one batch, the number of nodes had to be the same. Thus 32 threads were used for the whole scalability test.

12.2.2.2. Scalability

There is a small increase in total computation time without I/O time from 1 to 16 processes (figure 12.8, left and table 12.6). The 2-fold hyper-threading results in an additional increase, but afterwards the time stays essentially constant. The time with 917'504 is much shorter, but this is due to the different structure resulting from the changed grid partitioning. This results in a different flux field and a larger time step calculated by the CFL condition and thus a smaller number of time steps. The time for a single time step is still the same. The parallel efficiency for a single time step is 85 % (compared to a sequential computation) up to 917'504 processes (figure 12.8, right). If the 10 % loss by hyper-threading is taken into account it is even 95 %. While this is to be expected for an explicit scheme in principle it is still surprisingly good.

However, computation time for the largest simulations is only in the order or below the time needed for file transfer. This is partly due to the fact that only 900 seconds of computed time have been simulated. As an hourly time step was used for the water transport calculation there would only be a need to read one set of flux densities and volumetric water content and write one set of results every hour. Thus for longer simulations the ratio of file transfer to calculation for the largest simulation would be

P	steps	total	input	output	calculation	t_{step}
1	355	650.7	2.4	0.9	648.2	1.8
4	368	677.5	3.1	1.1	674.4	1.8
16	372	700.5	7.3	1.3	693.2	1.9
64	372	762.7	11.5	7.4	751.3	2.0
256	372	774.3	16.6	6.5	757.7	2.0
1024	376	782.3	25.5	15.1	756.8	2.0
4096	376	793.7	27.2	11.7	766.5	2.0
16’384	372	959.8	199.7	127.8	760.2	2.0
65’536	372	1015.8	244.2	233.8	771.6	2.1
262’144	376	1386.0	597.2	379.0	788.8	2.1
524’288	376	1000.6	222.9	138.0	777.7	2.1
917’504	216	805.1	368.6	166.8	436.5	2.0

Table 12.7.: Number of time steps and total parallel run time, time for file input and output, the pure calculation and time needed for one time step (t_{step}) for the block test case of the parabolic scenario on JUQUEEN.

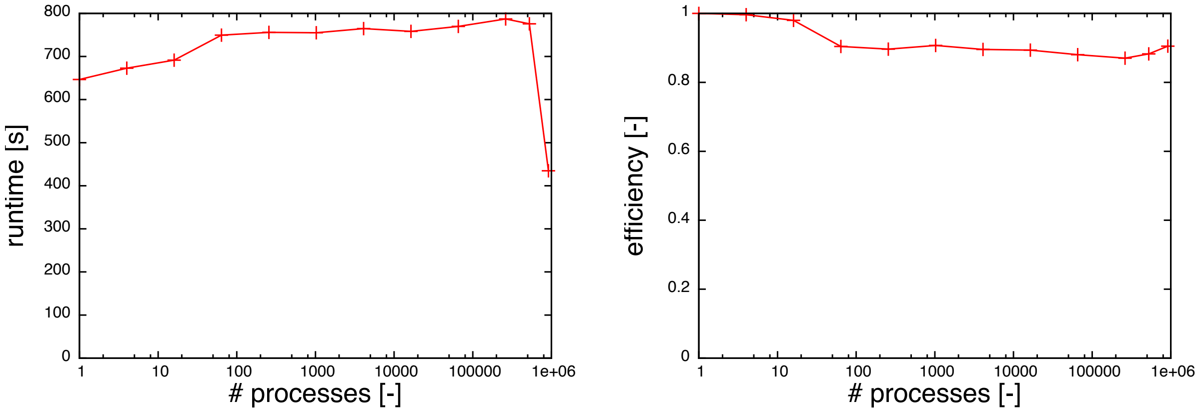

Figure 12.8.: Total computation time and parallel efficiency for solute transport with up to 917’504 processes.

more like 1 : 5. File transfer is faster for the two largest simulations (table 12.7) as some parameters for SIONlib have not been set optimally for the smaller test cases.

12.3. Parallel File Transfer on JUGENE and JUQUEEN

As file transfer can be a bottleneck for large-scale simulations it will be analysed more closely in this section using data from all scalability test performed.

12.3.1. File Input

As mentioned in section 12.1.1.5 the input of the structure used in the large test cases was realised with two different libraries: HDF5 as well as SIONlib. Whereas for the large test case of the parabolic scenario file I/O with HDF5 was used, for the large test case of the elliptic scenario the file I/O was based on SIONlib as the former was too slow for large problems. The dependency of the read file transfer rate on the number of processes is shown in figure 12.9 on the left. With up to 512 processes the file transfer rate on JUGENE increased with the number of processes following a power law both for HDF5 and for SIONlib, with the latter being faster by about a factor of three. However, the file transfer rate with HDF5 stays more or less constant from 4096 to 294'849 processes, while the file transfer rate with SIONlib increases. For the largest problems SIONlib is, with a maximal transfer rate of 6.28 GB/s, faster by a factor 15 (table 12.8). On JUQUEEN there was no need for parallel input in the water transport simulation as the block-structure was read by the rank zero process and broadcasted to the other processes. The solute transport code had to read fields of porosity, water content and flux density. SIONlib was used for this purpose. File transfer was nearly four times faster than measured on JUGENE with the same number of processes and increased up to 47.4 GB/s with 917'504 processes.

12.3.2. File Output

For the output of the results HDF5 was also tested initially. However, output to three-dimensional data formats was already prohibitively slow on JUGENE with several hundred processes. This is most probably due to the necessary reordering of the data during writing. A modified version, where each process wrote its data consecutively in a one-dimensional array, performed better, but was not competitive with SIONlib and also required a post-processing. The better performance of SIONlib probably is a consequence of the prevention of access to the same (large) file system block.

The measured write file transfer with SIONlib also follows a power law (figure 12.9). While the file transfer rate for small processes shows some spreading, the file transfer rate for the large simulations is rather similar. However, this effect is also exaggerated by the double logarithmic plot. The slow output for $\mu\varphi$ on JUQUEEN up to 262'144 processes and for solute from 16384 to 262144 processes was caused by the use of a wrong communicator for SIONlib. For the simulations with the corrected code (solute with

		transfer rate [GB/s]		*transfer data* [TB]	
	processes	*input*	*output*	*input*	*output*
JUGENE					
HDF5	4’096	0.18		0.009	
	294’849	0.40		0.62	
SIONlib	4’096	0.63	1.4/4.8	0.008	0.05/0.11
	287’496	6.28	13.5	0.59	7.7
	294’849		10.1		1.24
JUQUEEN					
$\mu\varphi$	4’096	0.76		0.1	
	917’504		77.0		21.3
solute	4’960	2.09	5.19	0.08	0.06
	917’504	47.4	81.2	17.5	13.5

Table 12.8.: Average transfer rates of file I/O on JUGENE and JUQUEEN and total amount of transferred data. Here 1 TB = 1000 GB=10^{12} byte.

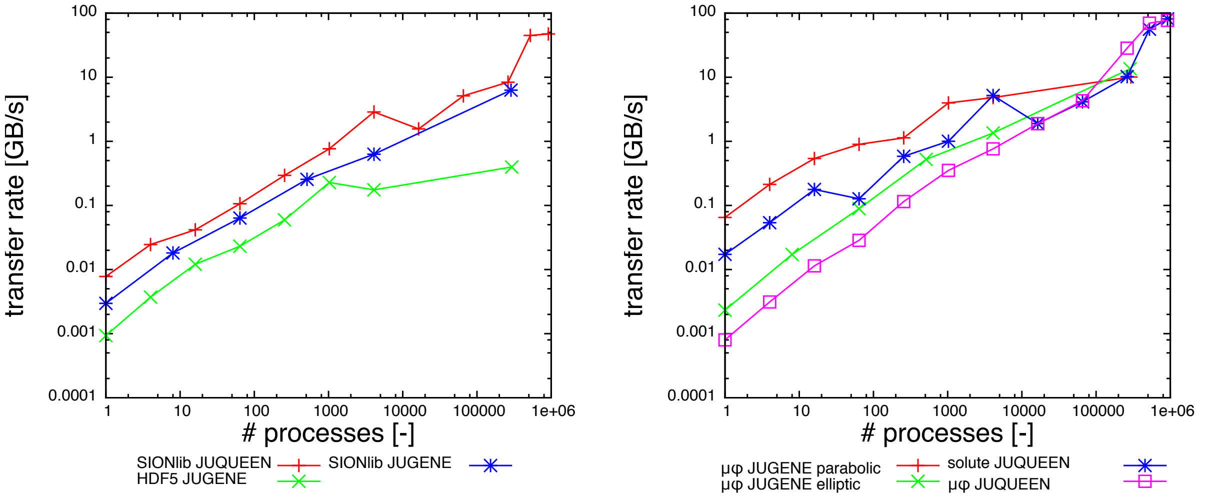

Figure 12.9.: File transfer rates in GB/s on JUGENE and JUQUEEN with different libraries for input (left) and output (right) in a double logarithmic plot. Here 1 TB = 1000 GB=10^{12} byte.

1 to 65’536 and the simulations with 524’288 and 917’504 processes) the transfer rates were also above the values measured on JUGENE with the same number of processes. The maximal measured write rate was 81.2 GB/s. Wolfgang Frings, the developer of SIONlib, measured maximal write rates of 100-120 GB/s and read rates of ≈ 65 GB/s with 1.8 million processes in a favourable configuration (personal communication). As the transfer rates given in table 12.8 are net transfer rates with the file system blocks assigned to each process not always filled completely, this is very close to the maximal bandwidth of the file system.

With the same number of processes the transfer rates for writing were always higher than for reading. While for the structures only a single file could be used as the sequential version of SIONlib does not support the writing of multiple files, this does not explain the lower rates during the input of the solute transport code, where the data had been already written to multiple files. As it was also measured by Wolfgang Frings it seems to be a hardware effect.

The scalability tests of file I/O showed that efficient parallel file transfer with the huge data sets involved in the simulation of large systems (up to 21 TB in the tests) is possible. Data I/O can, however, still consume a significant fraction of the total run-time. To obtain a high performance it is mandatory that each process stores its data in a consecutive block. In the best case each process gets a file system block of its own.

Part IV.

Conclusions and Outlook

13. Conclusions

In this work powerful software tools for the sequential and parallel simulation of water and solute transport in strongly heterogeneous porous media have been developed. They have been shown to scale extremely good on BlueGene/P and /Q supercomputers. The efficient use of petascale computers is thus already possible. If massively parallel computers like JUQUEEN become available to a wider public this would open up a new perspective for high-resolution large-scale land-surface simulations. Together with a handful of other models $\mu\varphi$ and its transient solute transport solver are members of the High-Q club of best scaling codes on JUQUEEN.

A bottleneck especially for the solution of elliptic problems is the repartitioning of unknowns during the coarsening of the algebraic multi-grid solver. This is already addressed as part of the EXADUNE project funded by the German Research Association within the framework of the German Priority Programme "Software for Exascale Computing".

The input and output of structures and results can require a major part of the run-time of massively parallel simulations if frequent output is required or the simulated time is rather short. Thus it is reassuring that parallel file systems can be used efficiently with transfer rates close to the technical maximum. A careful choice of the libraries and of the I/O strategy is a necessary precondition for this.

The developed software has been applied to several problems of practical relevance. Solute transport in heterogeneous unsaturated systems is still a problem with many open questions. The fitting of breakthrough curves only gives limited insight and the obtained parameters are hardly transferable. Thus the inclusion of structural information is a promising approach. While structural information today is easy to measure e.g. with X-ray micro-tomography, the hydraulic parameters of the different materials are not. The comparison of two different approaches for parameter estimation showed that the direct usage of the structure in a three-dimensional optimisation is not an easy task. The estimation of effective parameters from a one-dimensional inversion with subsequent down-scaling of the properties in contrast yielded a surprisingly good prediction of the shape of the breakthrough curve when the measured structure was included into the simulation. For saturated porous media three-dimensional inversion is more advanced as the system is linear and less over-parametrised. Here new parallel geostatistic inversion schemes are currently developed in our group.

The estimation of parameters for heterogeneous soils remains, however, an important task. Virtual soil systems can be useful tools in the optimisation of measurement strategies, the development of better effective models and the analysis of model errors. Three-dimensional virtual soil systems with a realistic structure and a high spatial resolution have been presented in this work. Water and solute transport have been simulated with unprecedented resolution in space and time. The results gave interesting insights in the influence of sub-surface properties on effective fluxes into atmosphere and groundwater. While evaporation fluxes depended mostly on the near-surface structure of the soil, the subsoil boundary in 1.5 metre depth still had a significant influence. Seepage rates on the other hand were hardly influenced by this factors and depended more on the amount of water already stored in the sub-surface. Soil heterogeneity had a much higher influence on solute transport than on water flow. The solute simulations also showed, that all water reaching the groundwater was “old” water. The simulation results are currently being processed and will be made available for public access via a web front-end.

One approach to develop effective models for heterogeneous soils is the inclusion of macroscopic hydraulic non-equilibrium. The model of Ross and Smettem (which is very similar to the model of Barenblatt) was implemented and tested with simulations of infiltration into heterogeneous soils with different correlation length. Effective hydraulic functions were derived for the heterogeneous systems. The qualitative agreement between the heterogeneous simulations and homogeneous simulations with the effective properties and different relaxation times was good. However, there are systematic differences in the shape of the water content profiles behind the infiltration front. Further research is necessary as (macroscopic) non-equilibrium is an issue during fast flow processes.

Bibliography

Barenblatt, G. I., T. W. Patzek, and D. B. Silin (2003). “The Mathematical Model of Nonequilibrium Effects in Water-Oil Displacement”. In: *SPE J.* 8.4, pp. 409–416. DOI: 10.2118/87329-PA.

Bastian, P. (1999). *Numerical Computation of Multiphase Flows in Porous Media.*

Bastian, P. et al. (1997). “UG: A flexible software toolbox for solving partial differential equations”. In: *Computing and Visualization in Science* 1, pp. 27–40.

Bastian, Peter et al. (2008a). “A generic grid interface for parallel and adaptive scientific computing. Part I: abstract framework”. In: *Computing* 82.2–3, pp. 103–119. DOI: 10.1007/s00607-008-0003-x.

Bastian, Peter et al. (2008b). “A generic grid interface for parallel and adaptive scientific computing. Part II: implementation and test in DUNE”. In: *Computing* 82.2–3, pp. 121–138. DOI: 10.1007/s00607-008-0004-9.

Bayer, Peter et al. (2013). “Review on life cycle environmental effects of geothermal power generation”. In: *Renewable and Sustainable Energy Reviews* 26, pp. 446–463.

Bear, J. (1961). “On the tensor form of dispersion in porous media”. In: *Geophysical Prospecting* 66.4, pp. 1185–1197.

— (1972). *Dynamics of Fluids in Porous Media.* New York: Elsevier.

Bear, J. and Y. Bachmat (1991). *Introduction to Modeling of Transport Phenomena in Porous Media.* Kluwer Academic Publishers.

Bechtold, M et al. (2011). “Efficient random walk particle tracking algorithm for advective-dispersive transport in media with discontinuous dispersion coefficients and water contents”. In: *Water Resources Research* 47.10, W10526–.

Biggar, J. W. and S. A. Taylor (1960). “Some aspects of the kinetics of moisture flow into unsaturated soil”. In: *Soil Sci. Soc. Am. Proc.* 24, pp. 81–85.

Binning, Philip and Michael A Celia (1996). “A Finite Volume Eulerian-Lagrangian Localized Adjoint Method for Solution of the Contaminant Transport Equations in Two-Dimensional Multiphase flow Systems”. In: *Water Resources Research* 32.1, pp. 103–114.

Bitterlich, S. et al. (2004). “Inverse Estimation of the Unsaturated Soil Hydraulic Properties from Column Outflow Experiments Using Free-Form Parameterizations”. In: *Vadose Zone Journal* 3, pp. 971–981.

Blatt, Markus and Peter Bastian (2007). “The Iterative Solver Template Library”. In: *Applied Parallel Computing. State of the Art in Scientific Computing.* Ed. by Bo Kågström et al. Vol. 4699. Lecture Notes in Computer Science. Springer, pp. 666–675. DOI: 10.1007/978-3-540-75755-9_82.

Blatt, Markus and Peter Bastian (2008). “On the Generic Parallelisation of Iterative Solvers for the Finite Element Method”. In: *Int. J. Comput. Sci. Engrg.* 4.1, pp. 56–69. DOI: `10.1504/IJCSE.2008.021112`.

Blatt, Markus, Olaf Ippisch, and Peter Bastian (2013). “A Massively Parallel Algebraic Multigrid Preconditioner based on Aggregation for Elliptic Problems with Heterogeneous Coefficients”.

Bock, Hans Georg (1987). *Randwertproblemmethoden zur Parameteridentifizierung in Systemen nichtlinearer Differentialgleichungen.* Bonner Mathematische Schriften 183. Math.-Naturwiss. Fakultät der Universität Bonn.

Boopathy, R (2000). “Factors limiting bioremediation technologies”. In: *Bioresource Technology* 74.1, pp. 63–67.

Boutraa, Tahar (2010). “Improvement of water use efficiency in Irrigated agriculture: A Review”. In: *Journal of Agronomy* 9.1, pp. 1–8.

Braess, D (1992). *Finite elemente.* Springer, Berlin.

Braess, Dietrich (1995). “Towards algebraic multigrid for elliptic problems of second order”. In: *Computing* 55.4, pp. 379–393.

Brodlie, KW (1980). “A review of methods for curve and function drawing”. In: *Mathematical methods in computer graphics and design*, pp. 1–37.

Brooks, Alexander N and Thomas JR Hughes (1982). “Streamline upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations”. In: *Computer methods in applied mechanics and engineering* 32.1, pp. 199–259.

Brooks, R. H. and A. T. Corey (1966). “Properties of porous media affecting fluid flow”. In: *J. Irrigation and Drainage Div., Proc. Am. Soc. Civil Eng. (IR2)* 92, pp. 61–88.

Buckingham, E. (1907). *Studies on the movement of soil moisture.* Bulletin 38. Washington, DC: U.S. Department of Agriculture, Bureau of Soils.

Burdine, N T (1953). “Relative permeability calculations from pore size distribution data”. In: *Journal of Petroleum Technology* 5.3, pp. 71–78.

Camps-Roach, G. et al. (2010). “Experimental investigation of dynamic effects in capillary pressure: Grain size dependency and upscaling”. In: *Water Resour. Res.* 46.W08544, pp. 1–13. DOI: `10.1029/2009WR008881`.

Chardaire, C. Chavent G. Jaffré J. and J. Liu (1992). “Relative Permeabilities and Capillary Pressure Estimation through Least Square Fitting”. In: *The Mathematics of Oil Recovery.* Ed. by P. R. King. Oxford: Clarendon Press.

Class, Holger and Rainer Helmig (2002). “Numerical simulation of non-isothermal multiphase multicomponent processes in porous media.: 2. Applications for the injection of steam and air”. In: *Advances in water resources* 25.5, pp. 551–564.

Darcy, H. (1856). *Les Fontaines de la Ville de Dijon.* Dalmont, Paris.

Demmel, James W. et al. (1999). “A supernodal approach to sparse partial pivoting”. In: *SIAM J. Matrix Analysis and Applications* 20.3, pp. 720–755.

Di Pietro, Daniele Antonio and Alexandre Ern (2012). *Mathematical aspects of discontinuous Galerkin methods.* Vol. 69. Springer.

Doster, F., P. A. Zegeling, and R. Hilfer (2010). “Numerical solutions of a generalized theory for macroscopic capillarity”. In: *Physical Review E* 81, pp. 036307–1 –036307–13. DOI: `10.1103/PhysRevE.81.036307`.

Douglas Jr, Jim and Thomas F Russell (1982). “Numerical methods for convection-dominated diffusion problems based on combining the method of characteristics with finite element or finite difference procedures”. In: *SIAM Journal on Numerical Analysis* 19.5, pp. 871–885.

Durner, W. (1994). “Hydraulic conductivity estimation for soils with heterogeneous pore structure”. In: *Water Resour. Res.* 30, pp. 211–223.

Durner, W and SC Iden (2011). “Extended multistep outflow method for the accurate determination of soil hydraulic properties near water saturation”. In: *Water Resources Research* 47.8, W08526.

EXASTEEL (2014). *Bridging Scales for Multiphase Steels (EXASTEEL)*. `http://www.mi.uni-koeln.de/numerik/de/left/research/exasteel/`.

Farthing, M. W. Kees C. E. Coffey T. S. Kelley C. T. and C. T. Miller (2003). “Efficient steady-state solution techniques for variably saturated groundwater flow”. In: *Adv. Water Res.* 26. copied, pdf, pp. 833–849.

Feddes, Reinder A et al. (1976). “Simulation of field water uptake by plants using a soil water dependent root extraction function”. In: *Journal of Hydrology* 31.1, pp. 13–26.

Federal Institute for Geosciences and Natural Resources (BGR) (2014). *Groundwater*. `http://www.bgr.bund.de/EN/Themen/Wasser/wasser_node_en.html`.

Frings, Wolfgang (2009–2013). *SIONlib*. `http://www.fz-juelich.de/jsc/sionlib`.

Frings, Wolfgang, Felix Wolf, and Ventsislav Petkov (2009). “Scalable massively parallel I/O to task-local files”. In: *High Performance Computing Networking, Storage and Analysis, Proceedings of the Conference on*. IEEE, pp. 1–11.

Fritsch, FN and Ji Butland (1984). “A method for constructing local monotone piecewise cubic interpolants”. In: *SIAM journal on scientific and statistical computing* 5.2, pp. 300–304.

Fritsch, F.N. and R.E. Carlson (1980). “Monotone piecewise cubic interpolation”. In: *SIAM Journal on Numerical Analysis*, pp. 238–246.

Gassiat, Claire et al. (2013). “Hydraulic fracturing in faulted sedimentary basins: Numerical simulation of potential contamination of shallow aquifers over long time scales”. In: *Water Resources Research*.

Gebbers, Robin and Viacheslav I Adamchuk (2010). “Precision agriculture and food security”. In: *Science* 327.5967, pp. 828–831.

Gerke, H. H. and M. Th. van Genuchten (1993). “A dual-porosity model for simulating the preferential movement of water and solutes in structured porous media”. In: *Water Resour. Res.* 29, pp. 305–319.

Gmeiner, Björn et al. (2012). “Highly parallel geometric multigrid algorithm for hierarchical hybrid grids”. In: *Hierarchival Methods for Dynamics in Complex Molecular Systems, J. Grotendorst, G. Sutmann, G. Gompper, and D. Marx, Eds. Forschungszentrum Jülich*, p. 323.

Guswa, A J (2005). “Soil-moisture limits on plant uptake: An upscaled relationship for water-limited ecosystems”. In: *Advances in water resources* 28.6, pp. 543–552.

Haines, W. B. (1930). “Studies in the physical properties of soils: V. The hysteresis effect in capillary properties and the modes of moisture distribution associated therewith”. In: *J. Agric. Sci.* 20, pp. 97–116.

Haines, William B (1927). “Studies in the physical properties of soils: IV. A further contribution to the theory of capillary phenomena in soil”. In: *The Journal of Agricultural Science* 17.02, pp. 264–290.

Hardelauf, H et al. (2007). “PARSWMS: A Parallelized Model for Simulating Three-Dimensional Water Flow and Solute Transport in Variably Saturated Soils”. In: *Vadose Zone Journal* 6.2, p. 255.

Hassanizadeh, S. M. and W. G. Gray (1990). “Mechanics and thermodynamics of multiphase flow in porous media including interphase boundaries”. In: *Adv. Water Res.* 13.e, pp. 169–186.

— (1993). “Toward an improved description of the physics of two-phase flow”. In: *Adv. Water Res.* 16, pp. 53–67.

Hassanizadeh, S. M., M. A. Celia, and H. K. Dahle (2002). “Dynamic Effect in the Capillary Pressure-Saturation Relationship and its Impacts on Unsaturated Flow”. In: *Vadose Zone Journal* 1. copied, pdf, pp. 38–57.

Helmig, Rainer and Ralf Huber (1998). “Comparison of Galerkin-type discretization techniques for two-phase flow in heterogeneous porous media”. In: *Advances in water resources* 21.8, pp. 697–711.

Herzog, Alexandra (2005). “Numerische Modellierung der Bodenwassercharakteristik in porösen Medien”. MA thesis. Ruprecht-Karls-Universität Heidelberg.

Hilfer, R. (2006a). “Capillary pressure, hysteresis and residual saturation in porous media”. In: *Physica A* 359, pp. 119–128. DOI: 10.1016/j.physa.2005.05.086.

Hilfer, R (2006b). “Macroscopic capillarity and hysteresis for flow in porous media”. In: *Physical Review E* 73.1, p. 016307.

Hopmans, J. W. et al. (2002). “Simultaneous determination of water transmission and retention properties. Inverse methods”. In: *Methods of Soil analysis. Part 4*. Ed. by J. H. Dane and G. C. Topp. SSSA, Madison, WI.

Inoue, Yoshio, Kataryna Dabrowska-Zierinska, and Jiaguo Qi (2012). “Synoptic assessment of environmental impact of agricultural management: a case study on nitrogen fertiliser impact on groundwater quality, using a fine-scale geoinformation system”. In: *International Journal of Environmental Studies* 69.3, pp. 443–460.

Ippisch, O (2001). “Coupled transport in natural porous media”. In: *Combined faculties forthe natural sciences and for mathematics. Heidelberg, Germany: Rupertus Carola University*, p. 123.

Ippisch, O. (2003). “Coupled Transport in Natural Porous Media”. PhD thesis. University of Heidelberg. URL: http://www.ub.uni-heidelberg.de/archiv/3220.

Ippisch, O., H.-J. Vogel, and P. Bastian (2006). “Validity Limits for the van Genuchten-Mualem Model and Implications for Parameter Estimation and Numerical Simulation.” In: *Advances in Water Resources* 29.12, pp. 1780–1789. DOI: 10.1016/j.advwatres.2005.12.011.

Ippisch, Olaf (2013). *Muphi/Fitphi Guide*. http://conan.iwr.uni-heidelberg.de/muphi.

Jarvis, N J (1989). “A simple empirical model of root water uptake”. In: *Journal of Hydrology* 107.1, pp. 57–72.

Jenny, P, S.H Lee, and H.A Tchelepi (2003). “Multi-scale finite-volume method for elliptic problems in subsurface flow simulation”. In: *Journal of Computational Physics* 187.1, pp. 47 –67. ISSN: 0021-9991. DOI: http://dx.doi.org/10.1016/S0021-9991(03)00075-5. URL: http://www.sciencedirect.com/science/article/pii/S0021999103000755.

Jin, Y. and W. A. Jury (1996). “Characterizing the Dependence of Gas Diffusion Coefficient on Soil Properties”. In: *Soil Sci. Soc. Am. J.* 60.1, pp. 66–71.

Joekar-Niasar, V. and S. M. Hassanizadeh (2011). “Specific interfacial area: The missing state variable in two-phase flow equations”. In: *Water Resour. Res.* 47.W05513, pp. 1–14. DOI: 10.1029/2010WR00929.

Kastanek, Ferdinand J and Donald R Nielsen (2001). “Description of soil water characteristics using cubic spline interpolation”. In: *Soil Science Society of America Journal* 65.2, pp. 279–283.

Kasteel, R., H.-J. Vogel, and K. Roth (2000). “From local hydraulic properties to effective transport in soil”. In: *European J. Soil Sci.* 51. copied, pp. 81–91.

Kollet, Stefan J et al. (2010). “Proof of concept of regional scale hydrologic simulations at hydrologic resolution utilizing massively parallel computer resources”. In: *Water Resources Research* 46.4.

Kool, J. B., J. C. Parker, and M. T. van Genuchten (1987). “Parameter estimation for unsaturated flow and transport models - a review”. In: *J. Hydrol.* 91. pdf, copied, pp. 255–293.

Kuhlmann, A et al. (2012). “Influence of soil structure and root water uptake strategy on unsaturated flow in heterogeneous media”. In: *Water Resources Research* 48.2, W02534.

Kulli, Beatrice (2004). *”Durchblick in Bodenphysik – Webbasierte Hilfsmittel zur Vorlesung “Einführung in die Bodenphysik” von Prof. H. Flühler Umweltwissenschaften, ETH Zürich.”* http://www.soil-physics.ch/.

Le Guen, S. S. and A. R. Kovscek (2006). “Nonequilibirum Effects During Spontaneous Imbibition”. In: *Transport in Porous Media* 63, pp. 127–146. DOI: 10.1007/s11242-005-3327-4.

Lenhard, R J and J C Parker (1987). “A model for hysteretic constitutive relations governing multiphase flow: 2. Permeability-saturation relations”. In: *Water Resources Research* 23.12, pp. 2197–2206.

Leveque, R. J. (2002). *Finite Volume Methods for Hyperbolic Problems.* Cambridge University Press.

Leverett, M C (1940a). “Capillary behavior in porous solids”. In: *Trans. Am. Inst.*

Leverett, M. C. (1940b). “Capillary behavior in porous solids”. In: *Trans. Am. Inst*, pp. 152–169.

Li, K Y, R de Jong, and J B Boisvert (2001). “An exponential root-water-uptake model with water stress compensation”. In: *Journal of Hydrology* 252.1, pp. 189–204.

Li, X.S. et al. (1999). *SuperLU Users' Guide.* Tech. rep. LBNL-44289. `http://crd.lbl.gov/~xiaoye/SuperLU/`. Last update: August 2011. Lawrence Berkeley National Laboratory.

Lier, Q de Jong van et al. (2008). "Macroscopic Root Water Uptake Distribution Using a Matric Flux Potential Approach". In: *Vadose Zone Journal* 7.3, p. 1065.

Lorite, Ignacio J et al. (2012). "Assessment of the Irrigation Advisory Services' Recommendations and Farmers' Irrigation Management: A Case Study in Southern Spain". In: *Water resources management* 26.8, pp. 2397–2419.

Marquardt, D. W. (1963). "An Algorithm for Least-Squares Estimation of Nonlinear Parameters". In: *Journal of the Socity for Industrial and Applied Mathematics* 11.2, pp. 431–441.

Matheron, Georges et al. (1967). *Eléments pour une théorie des milieux poreux.* Masson Paris.

McCarthy, Alison C, Nigel H Hancock, and Steven R Raine (2013). "Advanced process control of irrigation: the current state and an analysis to aid future development". In: *Irrigation Science*, pp. 1–10.

Miller, E. E. and R. D. Miller (1956). "Physical theory for capillary flow phenomena". In: *J. Appl. Phys.* 27, pp. 324–332.

Millington, R. J. (1959). "Gas diffusion in porous media". In: *Science* 130, pp. 100–102.

Millington, R. J. and J. P. Quirk (1961). "Permeability of porous solids". In: *Trans. Faraday. Soc.* 57, pp. 1200–1207.

Moore, W. J. (1990). *Physical Chemistry.* Fifth Edition. Longman Scientific & Technical.

Mualem, Y. (1974). "A conceptual model of hysteresis". In: *Water Resour. Res.* 10, pp. 514–520.

— (1976). "A new model for predicting the hydraulic conductivity of unsaturated porous media". In: *Water Resour. Res.* 12, pp. 593–622.

— (1977). "Extension of the similarity hypothesis used for modeling the soil water characteristics". In: *Water Resources Research* 13.4, pp. 773–780.

— (1984). "A modified dependent domain theory of hysteresis". In: *Soil Sci.* 137, pp. 283–291.

Mualem, Y. and G. Dagan (1975). "A dependent domain model of capillary hysteresis". In: *Water Resources Research* 11.3, pp. 452–460.

Mualem, Y. and HJ Morel-Seytoux (1978). "Analysis of a capillary hysteresis model based on a one-variable distribution function". In: *Water Resources Research* 14.4, pp. 605–610.

Mueller, E.H. and R. Scheichl (2013). "Massively Parallel Solvers for Elliptic PDEs in Numerical Weather- and Climate Prediction".

Mulligan, CN, RN Yong, and BF Gibbs (2001). "Surfactant-enhanced remediation of contaminated soil: a review". In: *Engineering Geology* 60.1, pp. 371–380.

Murray, Francis W (1967). "On the computation of saturation vapor pressure". In: *Journal of Applied Meteorology* 6.1, pp. 203–204.

Muskat, Morris and Milan W Meres (1936). "The Flow of Heterogeneous Fluids Through Porous Media". In: *Physics* 7.9, p. 346.

Napov, Artem and Yvan Notay (2012). “An Algebraic Multigrid Method with Guaranteed Convergence Rate”. In: *SIAM journal on scientific computing* 34.2, A1079–A1109.

Neumann, Rebecca, Peter Bastian, and Olaf Ippisch (2013). “Modeling and simulation of two-phase two-component flow with disappearing nonwetting phase”. In: *Computational Geosciences* 17.1, pp. 139–149.

Nielsen, D. R., J. W. Biggar, and J. M. Davidson (1962). “Experimental consideration of diffusion analysis in unsaturated flow problems”. In: *Soil Sci. Soc. Am. Proc.* 26, pp. 107–111.

O’Carroll, D. M., T. J. Phelan, and L. M. Abriola (2005). “Exploring dynamic effects in capillary pressure in multistep outflow experiments”. In: *Water Resour. Res.* 41.W11419, pp. 1–14. DOI: 10.1029/2005WR004010.

O’Carroll, D. M. et al. (2010). “Influence of wettability variations on dynamic effects in capillary pressure”. In: *Water Resour. Res.* 46.W08505, pp. 1–13. DOI: 10.1029/2009WR008712.

Parker, J C and R J Lenhard (1987). “A model for hysteretic constitutive relations governing multiphase flow: 1. Saturation-pressure relations”. In: *Water Resources Research* 23.12, pp. 2187–2196.

Peters, A and W. Durner (2008). “Simplified evaporation method for determining soil hydraulic properties”. In: *Journal of Hydrology* 356.1-2, pp. 147–162.

Philip, J. R. (1969). “Theory of infiltration”. In: *Adv. Hydrosci.* 5, pp. 215–305.

Press, W. H. et al. (1992). *Numerical Recipes in C. The Art of Scientific Computing.* Second. Cambridge: Cambridge University Press.

Priesack, E. and W. Durner (2006). “Closed-Form Expression for the Multi-Modal Unsaturated Conductivity Function”. In: *Vadose Zone Journal* 5, pp. 121–124. DOI: 10.2136/vzj2005.0066.

Raviart, P.A. and J.M. Thomas (1977). “A mixed finite element method for 2-nd order elliptic problems”. In: *Mathematical Aspects of Finite Element Methods.* Ed. by Ilio Galligani and Enrico Magenes. Vol. 606. Lecture Notes in Mathematics. Springer Berlin Heidelberg, pp. 292–315. ISBN: 978-3-540-08432-7. DOI: 10.1007/BFb0064470. URL: http://dx.doi.org/10.1007/BFb0064470.

Rawlins, S. L. and W. H. Gardner (1963). “A test of the validity of the diffusion equation for unsaturated flow of soil water”. In: *Soil Sci. Soc. Am. Proc.* 27, pp. 507–511.

Rawlins, S.L. and G.S. Campell (1986). “Water potential: Thermocouple psychrometry”. In: *Methods of Soil Analysis, Part 1. Physical and Mineralogical Methods.* Ed. by A. Klute. Second. Agronomy Series 9. Madison, WI: American Society of Agronomy, pp. 597–618.

Richards, L. A. (1931). “Capillary conduction of liquids through porous mediums”. In: *Physics* 1, pp. 318–333.

Rivière, Béatrice (2008). *Discontinuous Galerkin methods for solving elliptic and parabolic equations: theory and implementation.* Vol. 35. Siam.

Ross, P. J. and K. R. J. Smettem (2000). “A Simple Treatment of Physical Nonequilibirum Water Flows in Soils”. In: *Soil Sci. Soc. Am. J.* 64, pp. 1926–1930.

Rossi, M et al. (2007). “Solute mixing during imbibition and drainage in a macroscopically heterogeneous medium”. In: *Water Resources Research* 43.4.

Rossi, M., O. Ippisch, and H. Flühler (2008). “Solute dilution under imbibition and drainage conditions in a heterogeneous structure: modeling of a sand tank experiment,” in: *Advances in Water Resources* 31, pp. 1242–1252. DOI: 10.1016/j.advwatres_2008.04003.

Roth, K. (1996). *Lecture Notes in Soil Physics.* University of Hohenheim, D-70593 Stuttgart: Institute of Soil Science.

Roth, Kurt (2012). *Soil Physics.* v2.2beta. Institute of Environmental Physics, Heidelberg University.

Schaap, Marcel G, Feike J Leij, and Martinus Th Van Genuchten (2001). “ROSETTA: a computer program for estimating soil hydraulic parameters with hierarchical pedotransfer functions”. In: *Journal of Hydrology* 251.3, pp. 163–176.

Scheidegger, A. E. (1961). “General theory of dispersion in porous media”. In: *Geophysical Prospecting* 66.10, pp. 3273–3278.

Schelle, H, SC Iden, and W Durner (2011). “Combined transient method for determining soil hydraulic properties in a wide pressure head range”. In: *Soil Science Society of America Journal* 75.5, pp. 1681–1693.

Schembre, J. M. and A. R. Kovscek (2006). “Estimation of Dynamic Relative Permeability and Capillary Pressure from Countercurrent Imbibition Experiments”. In: *Transport in Porous Media* 65, pp. 31–51. DOI: 10.1007/s11242-005-6092-5.

Schindler, Uwe et al. (2010). “The evaporation method: Extending the measurement range of soil hydraulic properties using the air-entry pressure of the ceramic cup”. In: *Journal of Plant Nutrition and Soil Science* 173.4, pp. 563–572.

Schlüter, Steffen et al. (2012). “Virtual Soils: Assessment of the Effects of Soil Structure on the Hydraulic Behavior of Cultivated Soils”. In: *Vadose Zone Journal* 11.4, p. 0.

Schlüter, S. et al. (2014). “Combined impact of soil heterogeneity and vegetation type on the annual water balance at the field scale”. In: *vzj*, accepted.

Schneider, K., O. Ippisch, and K. Roth (2006). “Novel Evaporation Experiment to measure Soil Hydraulic Properties”. In: *Hydrol. Earth Syst. Sci.* 10.6, pp. 817–827. URL: www.hydrol-earth-syst-sci.net/10/817/2006/.

Schneider-Zapp, K and O Ippisch (2010). “Numerical study of the evaporation process and parameter estimation analysis of an evaporation experiment”. In: *Hydrology and Earth . . .*

Schultze, B. (1998). “Optimierung der Messung bodenhydraulischer Eigenschaften durch inverse Simulation von Ausflu”s- und R”uckflu”sexperimenten an Bodens”aulen”. PhD thesis. Universit”at Bayreuth.

Genuchten, M. Th. Leij F. J. van and L. Wu, eds. (1999). *Dynamic Nonequilibirum During Unsaturated Water Flow.* pdf. Riverside, CA: University of California, pp. 877–892.

Schweizer, Ben (2012). “The Richards equation with hysteresis and degenerate capillary pressure”. In: *Journal of Differential Equations* 252.10, pp. 5594–5612.

Seneviratne, Sonia I. et al. (2010). “Investigating soil moisture–climate interactions in a changing climate: A review”. In: *Earth-Science Reviews* 99.3–4, pp. 125 –161. ISSN: 0012-8252. DOI: http://dx.doi.org/10.1016/j.earscirev.2010.02.004. URL: http://www.sciencedirect.com/science/article/pii/S0012825210000139.

Simunek, J, K Huang, M Th Van Genuchten, et al. (1997). *The HYDRUS-ET Software Package for Simulating the One-Dimentional Movement of Water, Heat and Multiple Solutes in Variably-Saturated Media, Version 1.1.* Bratislava: Inst. Hydrology Slovak Acad. Sci.

Stauffer, F. (1978). "Time dependence of the relations between capillary pressure, water content and conductivity during drainage of porous media". In: *Proceedings of the IAHR conference on scale effects in porous media, Thessaloniki, Greece.*

The HDF Group (2000–2013). *Hierarchical data format version 5.* `http://www.hdfgroup.org/HDF5`.

Topp, G. C. (1971). "Soil-water hysteresis: The domain model theory extended to pore interaction conditions". In: *Soil Sci. Soc. Am. J.* 35, pp. 219–225.

Topp, G. C., A. Klute, and D. B. Peters (1967). "Comparison of water content-pressure head data obtained by equilibrium, steady-state, and unsteady-state methods". In: *Soil Sci. Soc. Am. Proc.* 31, pp. 312–314.

Tsang, Chin-Fu et al. (2012). "Coupled thermo-hydro-mechanical processes in the near field of a high-level radioactive waste repository in clay formations". In: *International Journal of Rock Mechanics and Mining Sciences* 49, pp. 31–44.

Tuller, Markus and Dani Or (2001). "Hydraulic conductivity of variably saturated porous media: Film and corner flow in angular pore space". In: *Water Resources Research* 37.5, pp. 1257–1276.

Tuller, Markus, Dani Or, and Lynn M Dudley (1999). "Adsorption and capillary condensation in porous media: Liquid retention and interfacial configurations in angular pores". In: *Water Resources Research* 35.7, pp. 1949–1964.

Vachaud, G., M. Vauclin, and M. Wakil (1972). "A study of the uniqueness of the soil moisture characteristic during desorption by vertical drainage". In: *Soil Sci. Soc. Am. Proc.* 36, pp. 531–532.

Van Dam, J. C., J. N. M. Stricker, and P. Droogers (1994). "Inverse method to determine soil hydraulic functions from multistep outflow experiments". In: *Soil Sci. Soc. Am. J.* 58, pp. 647–652.

Van Duijn, C J et al. (2013). "Travelling wave solutions for degenerate pseudo-parabolic equations modelling two-phase flow in porous media". In: *Nonlinear Analysis: Real World Applications* 14.3, pp. 1361–1383.

van Genuchten, M. Th. (1980). "A closed-form equation for predicting the hydraulic conductivity of unsaturated soils". In: *Soil Sci. Soc. Am. J.* 44, pp. 892–898.

Vanderborght, J et al. (2005). "A set of analytical benchmarks to test numerical models of flow and transport in soils". In: *Vadose Zone Journal* 4.1, pp. 206–221.

Varado, N, I Braud, and P J Ross (2006). "Development and assessment of an efficient vadose zone module solving the 1D Richards' equation and including root extraction by plants". In: *Journal of Hydrology* 323.1-4, pp. 258–275.

Vogel, H. J. (2008). "QuantIm-Some useful C/C++ functions for scientific image processing'". In: *Helmholtz Center for Environmental Research: Halle.*

Vogel, H.-J., K. Roth, and P. Bastian (2002). "Using bulk density measured by X-ray tomography to estimate the hydraulic structure of soil". In: *Transactions 17th World Congress os Soil Science, Bangkok, 2002*, pp. 1218/1–1218/2.

Vogel, H. J., A. Samouëlian, and O. Ippisch (2008). "Multi-Step and two-step experiments in heterogeneous porous media to evaluate the relevance of dynamic effects". In: *Adv. Water Res.* 31, pp. 181–188. DOI: `10.1016/j.advwatres.2007.08.003`.

Vogel, H J, U Weller, and O Ippisch (2010). "Non-equilibrium in soil hydraulic modelling". In: *Journal of Hydrology* 393.1-2, pp. 20–28.

Vogel, T., M. Th. van Genuchten, and M. Cislerova (2001). "Effect of the shape of the soil hydraulic functions near saturation on variably-saturated flow predictions". In: *Adv. Water Res.* 24. copied, pdf, pp. 133–144.

Vorst, Henk A Van der (1992). "Bi-CGSTAB: A fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems". In: *SIAM Journal on scientific and Statistical Computing* 13.2, pp. 631–644.

Vrugt, Jasper A. et al. (2004). "Inverse modeling of large-scale spatially distributed vadose zone properties using global optimization". In: *Water Resources Research* 40.6, n/a–n/a. ISSN: 1944-7973. DOI: `10.1029/2003WR002706`. URL: `http://dx.doi.org/10.1029/2003WR002706`.

Weller, U. et al. (2011a). "Direct Measurement of Unsaturated Conductivity including Hydraulic Nonequilibrium and Hysteresis". In: *Vadose Zone Journal* 10.2, pp. 654–661. DOI: `10.2136/vzj2010.0074`.

Weller, U et al. (2011b). "Direct Measurement of Unsaturated Conductivity including Hydraulic Nonequilibrium and Hysteresis". In: *Vadose Zone Journal* 10.2, pp. 654–661.

Weller, Ulrich and Hans-Jörg Vogel (2012). "Conductivity and Hydraulic Nonequilibrium across Drainage and Infiltration Fronts". In: *Vadose Zone Journal* 11.3, p. 0.

Wildenschild, D., J. W. Hopmans, and J. Simunek (2001). "Flow Rate Dependence of Soil Hydraulic Characteristics". In: *Soil Sci. Soc. Am. J.* 65, pp. 35–48.

World Water Assessment Programme (2012). *Managing Water under Uncertainty and Risk.* The United Nations World Water Development Report 4. Paris: UNESCO.

Wyckoff, R D and H G Botset (1936). "The Flow of Gas-Liquid Mixtures Through Unconsolidated Sands". In: *Physics* 7.9, p. 325.

Zhang, Wei-xian (2003). "Nanoscale iron particles for environmental remediation: an overview". In: *Journal of nanoparticle Research* 5.3-4, pp. 323–332.

Zhao, Xiaoning et al. (2012). "Effect of optimal irrigation, different fertilization, and reduced tillage on soil organic carbon storage and crop yields in the North China Plain". In: *Journal of Plant Nutrition and Soil Science.*