Numerical Simulation of Transport Processes in Porous Media

Olaf Ippisch

email: `olaf.ippisch@iwr.uni-heidelberg.de`

March 27, 2013

Contents

1 Introduction

1.1 Subject of the Lecture

Groundwater Production/Flood Prediction
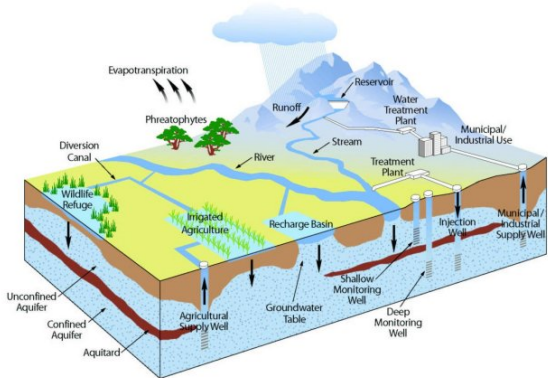
source: V. M. Ponce: Sustainable Yield of Ground Water (`http://ponce.sdsu.edu/groundwater_sustainable_yield.html`)

Agriculture

source: Myrabella [CC-BY-SA-3.0,2.5,2.0,1.0 or GFDL], from Wikimedia Commons
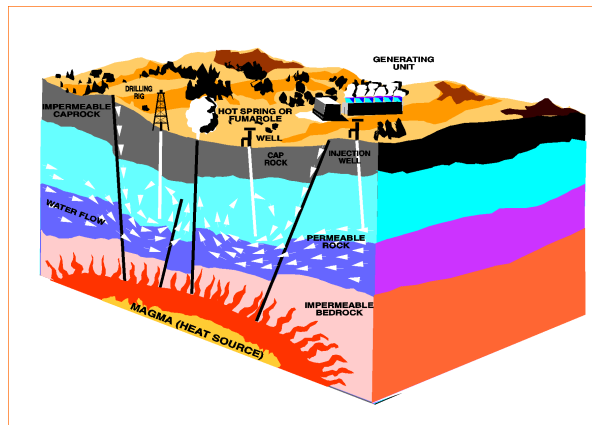
Optimization of Irrigation

source: United States Department of Agriculture (from Wikimedia Commons)

Assessment and Remediation of Contaminated Sites

Geothermal Energy

Oil Production/Reservoir Simulation

Global Climate Prediction, Reconstruction of Paleo-Climate

Global Warming Predictions

2070-2100 Prediction vs. 1960-1990 Average

Based on HadCM3

Temperature Increase (°C)
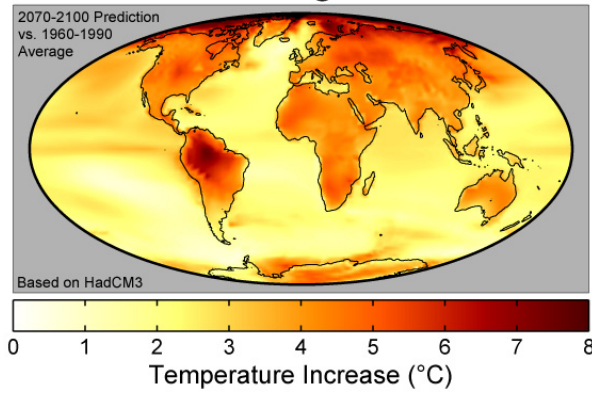
Carbon Dioxide Sequestration
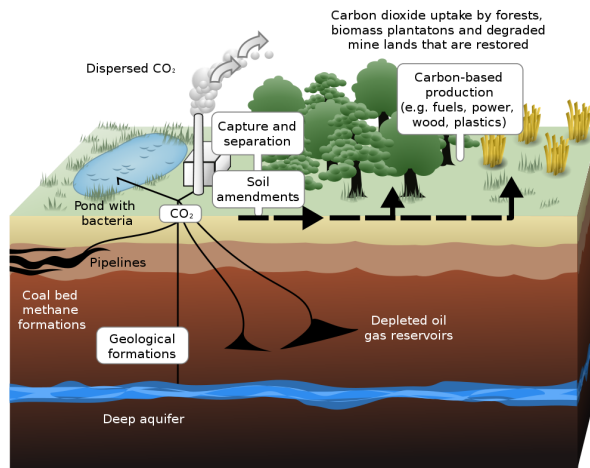
Carbon dioxide uptake by forests, biomass plantatons and degraded mine lands that are restored

Dispersed CO₂

Capture and separation

Carbon-based production (e.g. fuels, power, wood, plastics)

Soil amendments

Pond with bacteria

CO₂

Pipelines

Coal bed methane formations

Geological formations

Depleted oil gas reservoirs

Deep aquifer

Catalyst Research, Fuel Cells

$H_2 \rightarrow 2H^+ + 2e^-$ $\qquad$ $O_2 + 4H^+ + 4e^- \rightarrow 2H_2O$
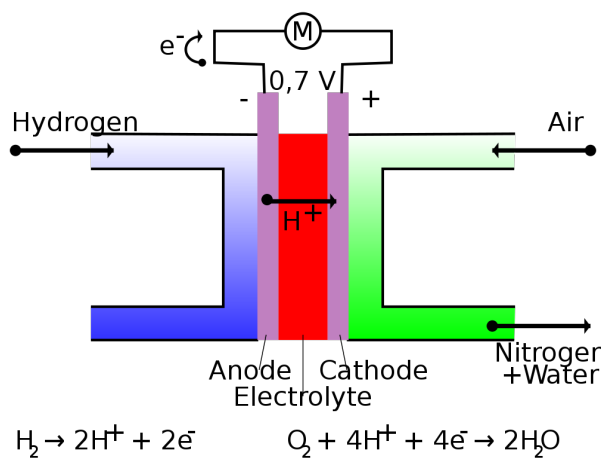
Transport in Brain Tissue
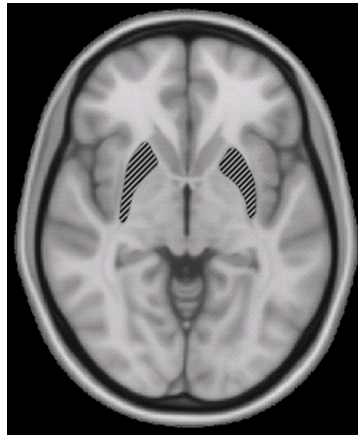
source: Woutergroen [public domain], from Wikimedia Commons

- Introduction to the physics of transport in porous media

- Learning the necessary basics on
    - Discretisation of partial differential equations, in particular the Finite-Volume method
    - Iterative solution of linear equation systems
    - Time discretisation
    - Solution of non-linear partial differential equations
    - Bottom-up implementation of numeric solvers

- Aims:
    - Get an insight in the operation of simulation programs
    - Get a better understanding for the behaviour of existing solvers for partial differential equations
    - Get a better understanding for the possible phenomena occuring in porous media flow

Prerequisites

For the lecture

- Basic knowledge of numerical mathematics

- Basic knowledge about partial differential equations

For the exercises

- Basic knowledge of object-oriented programming with C++ (Info1 lecture)

- Readiness to do some programming in the exercises

Topics

- Classification of partial differential equations

- Spatial discretization methods

- Finite-Volume methods

- Iterative solvers

- Groundwater flow / elliptic PDE

- Heat conduction / parabolic PDE

- Solute transport / hyperbolic PDE
  - Particle Tracking
  - Higher-order methods
  - Solute sorption

- Solution of non-linear equations

- Water transport in unsaturated porous media

What is not covered (thoroughly)?

- Detailed physics of flow in porous media

- Properties of natural porous media

- Analytical solutions

This is partially done in the lecture "Physics of Terrestrial Systems" by Kurt Roth, INF 229, room 108/110, Monday 16:15-17:45.

There is also a good english script for this lecture, which can be obtained at:

`http://www.iup.uni-heidelberg.de/institut/forschung/groups/ts/soil_physics/students/`
`lecture_notes05`

1.2 Example Problem: Groundwater Contamination Problem

The detection of a groundwater contamination is a typical example for the relevance of transport in porous media.

The water in several wells is contaminated with a soluble substance.

We know that there was an accident in a factory where the same substance was released to the groundwater.
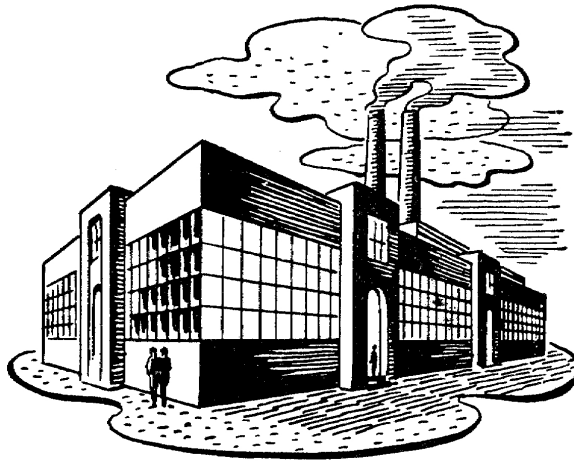
Time series of the concentration of the contaminant are available from several wells.
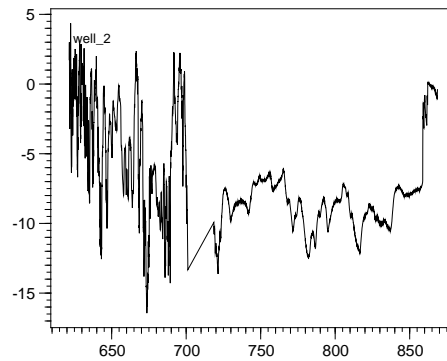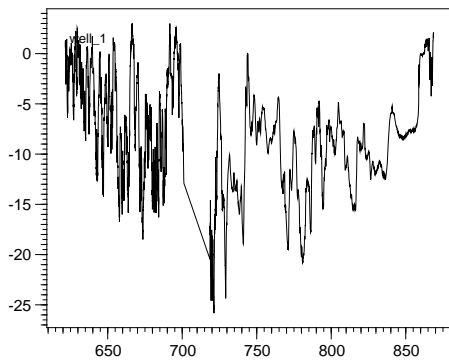
- Does this explain all the contamination?

- Can we reproduce the measurements?

- Is there another source involved?

What do we have to do to solve this problem?

- Compute the flow field for groundwater

- Determine the amount of contamination from the factory

- Solve solute transport problem

- Compare measurements at wells with the result

2 Groundwater Flow

Groundwater is subsurface water in a region where all pores are completely waterfilled. The flow of groundwater is influenced by the vertical inflow of water (groundwater recharge), the topography, and the geology of the aquifer.

Groundwater recharge depends on precipitation, evaporation of water directly from the soil, transpiration by plants and surface runoff.
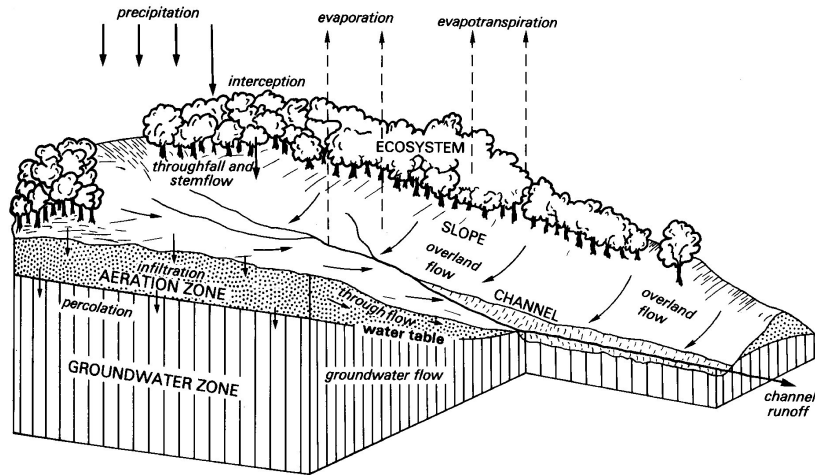
Figure 1: Schematic representation of a river catchment (from: Environmental Systems - An introductory text, I. D. White, D. N. Mottershead, S. J. Harrison, 2nd edition, Chapman & Hall).

Heterogeneity

The properties of a porous medium can depend on the position. This is called Heterogeneity and can be found on all scales from pore scale to regional scale.

Anisotropy

Natural porous media are also often anisotrope, i.e. their properties depend on the direction of flow. This can be due to geology but also caused by fractures or the compaction of regions close to the surface (e.g. plough pans)

Continuum approach

At pore scale the flux laws are well known (Navier-Stokes equations) but the pore geometry can neither be measured well enough, nor would it be possible to process the enormous amount of data which is necessary to simulate flow in a larger region.

For a locally sufficiently homogeneous porous medium it is possible to formulate a macroscopic equation at the continuum scale. The pore geometry is taken into account as a equivalent effective conductivity (similar to the transition from molecular description of a gas to the ideal gas law). This involves an averaging which should preserve the effective macroscopic behaviour (Figure 4).
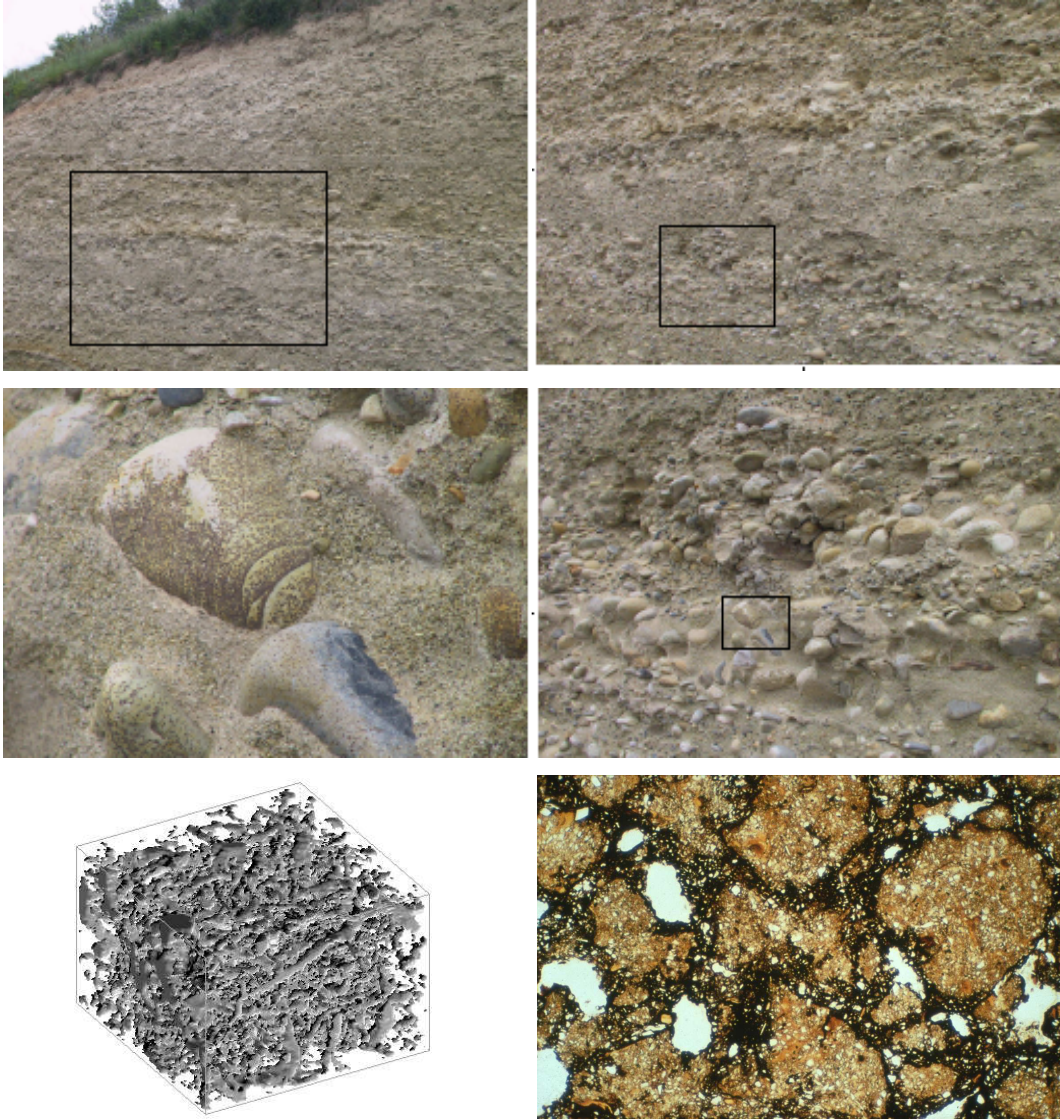
Figure 2: Heterogeneity of a natural porous medium at different scales (from: K. Roth (2005), Soil Physics - Lecture Notes v1.0, Institut für Umweltphysik, Universität Heidelberg)
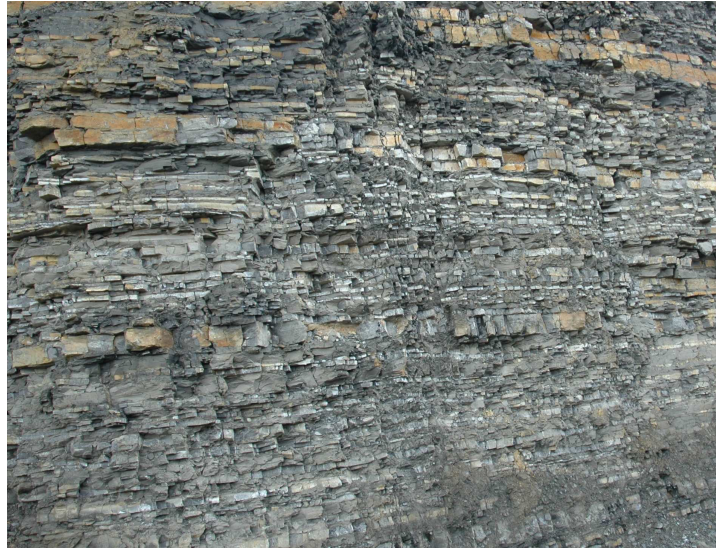
Figure 3: Natural porous media are often anisotrope (different permeability in different directions of flow) (from: K. Roth (2005), Soil Physics - Lecture Notes v1.0, Institut für Umweltphysik, Universität Heidelberg)
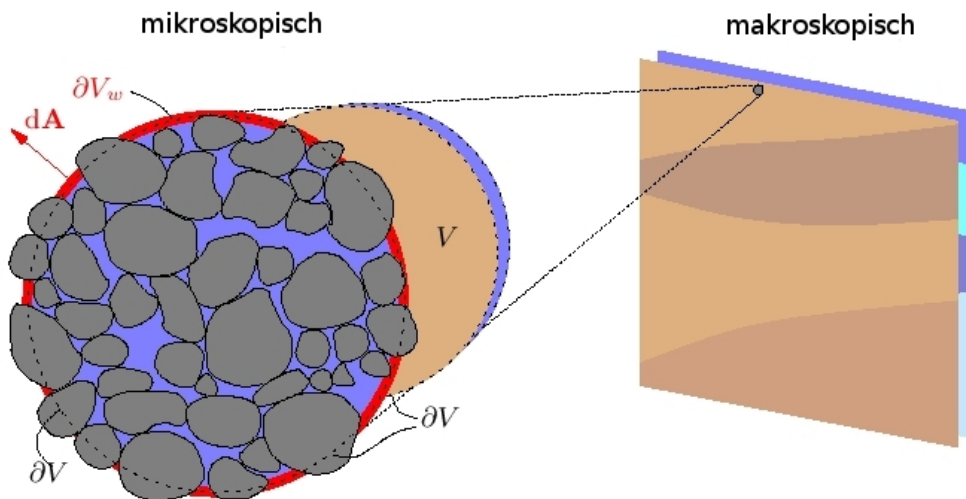
Figure 4: Transition from pore scale to continuum scale with averaged properties (from: K. Roth (2005), Soil Physics - Lecture Notes v1.0, Institut für Umweltphysik, Universität Heidelberg)

Darcy Equation

Such a flux law was proposed for the first time by Henry Darcy in 1856 (H. Darcy: Les Fontaines de la Ville de Dijon, Dalmont, Paris). According to Darcy's law the volumetric flux through a porous medium is proportional to the applied pressure gradient. The constant of proportionality is characteristic for the material and is called saturated hydraulic conductivity (K_s).

$$J_w = -K_s \cdot \frac{\Delta p_w}{\Delta x}$$

for $\Delta x \to 0$

$$J_w = -K_s \cdot \frac{\partial p_w}{\partial x}$$

in three dimensions:

$$\vec{J}_w = -\bar{K}_s \cdot \begin{pmatrix} \frac{\partial p_w}{\partial x} \\ \frac{\partial p_w}{\partial y} \\ \frac{\partial p_w}{\partial z} \end{pmatrix} = -\bar{K}_s \cdot \nabla p_w$$

Mass Conservation

The total mass has to be locally preserved during a transport process. The mass balance over a control volume of soil therefore has to add up. Components of the mass balance are the fluxes over the sides of the control volume, the change of water storage in the control volume and the water extraction or induction due to e.g. roots or wells (Figure 5). Often a volume conservation is considered instead. The water content θ_w is a dimensionless quantity which describes then the fraction of the soil which is filled by water.

Transport Equation

The combination of Darcy's equation and mass balance yields the transport equation:

$$\frac{\partial \theta_w(\vec{x})}{\partial t} + \nabla \cdot \vec{J}_w(\vec{x}) + r_w(\vec{x}) = 0$$

$$\frac{\partial \theta_w(\vec{x})}{\partial t} + \nabla \cdot \left[-\bar{K}_s(\vec{x}) \cdot \nabla p_w \right] + r_w(\vec{x}) = 0$$

$$\frac{\partial \theta_w(\vec{x})}{\partial t} - \nabla \cdot \left[\bar{K}_s(\vec{x}) \cdot \nabla p_w \right] + r_w(\vec{x}) = 0$$

The inclusion of gravity results in an additional driving force in vertical direction:

$$\frac{\partial \theta_w(\vec{x})}{\partial t} - \nabla \cdot \left[\bar{K}_s(\vec{x}) \cdot (\nabla p_w - \rho_w g \vec{e}_z) \right] + r_w(\vec{x}) = 0$$

In steady-state or if the water storage does not depend on the pressure, the time dependent terms vanish and the equations simplifies to:

$$-\nabla \cdot \left[\bar{K}_s(\vec{x}) \cdot (\nabla p_w - \rho_w g \vec{e}_z) \right] + r_w(\vec{x}) = 0$$

Water Extraction Rate
$r_w(x,y,z,t)\Delta x\Delta y\Delta z$

Water Outflow Rate
$J_w(x,y,z+\Delta z,t)\Delta x\Delta y$

Plant
root

Flow Area $A = \Delta x\,\Delta y$

Soil Water Storage
$\theta_w(x,y,\bar{z},t)\Delta x\Delta y\Delta z$

Δz

Δy

Δx

Water Inflow Rate
$J_w(x,y,z,t)\Delta x\Delta y$
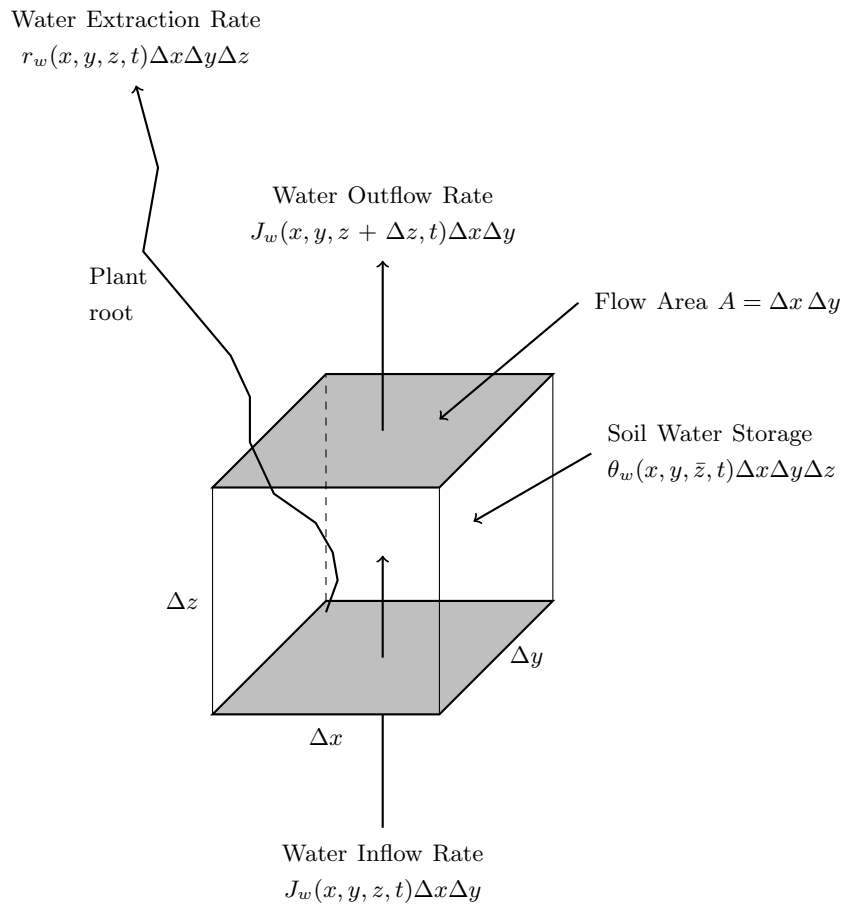
Figure 5: Mass balance for a cubic control volume (according to W. A. Jury, R. Horton (2004): Soil Physics, 6th ed, Wiley & Sons, New Jersey)

Summary: Groundwater Flow

- Groundwater flow can be described by Darcy's Law $J_w = -K_s \nabla p_w$ and the continuity equation $\frac{\partial \theta_w(\vec{x})}{\partial t} + \nabla \cdot \vec{J}_w(\vec{x}) + r_w(\vec{x}) = 0$.

- Gravity results in an addition driving force $-\rho_w g \vec{e}_z$:

$$\frac{\partial \theta_w(\vec{x})}{\partial t} - \nabla \cdot \left[\bar{K}_s(\vec{x}) \cdot (\nabla p_w - \rho_w g \vec{e}_z) \right] + r_w(\vec{x}) = 0$$

- Heterogeneity is considered by different values of K_s at different positions of $\vec{x}$

- Anisotropy is considered by using a tensor $\bar{K}_s$ instead of a scalar

- In steady state the flux equation is given by:

$$-\nabla \cdot \left[\bar{K}_s(\vec{x}) \cdot (\nabla p_w - \rho_w g \vec{e}_z) \right] + r_w(\vec{x}) = 0$$

3 Partial Differential Equations

A partial differential equation

- determines a function $u(\vec{x})$ in $n \geq 2$ variables $\vec{x} = (x_1, \ldots, x_n)^{\mathrm{T}}$.

- is a functional relation between partial derivatives (to more than one variable) of u at *one* point.

In general:

$$F\left(\frac{\partial^m u}{\partial x_1^m}(\vec{x}), \frac{\partial^{m-1} u}{\partial x_1^{m-1}}(\vec{x}), \ldots, \frac{\partial^m u}{\partial x_1^{m-1} \partial x_2}(\vec{x}), \ldots, \frac{\partial^m u}{\partial x_n^m}(\vec{x}), \frac{\partial^{m-1} u}{\partial x_n^{m-1}}(\vec{x}), \ldots, u(\vec{x}), \vec{x}\right) = 0 \quad \forall \vec{x} \in \Omega \tag{1}$$

Important:

- The highest derivative m determines the order of a PDE

PDE's are not posed on the whole $\mathbb{R}^n$ but on a subset of $\mathbb{R}^n$.

Definition 3.1 (Domain). $\Omega \subseteq \mathbb{R}^n$ is called domain if Ω is open and connected.

open: For each $\vec{x} \in \Omega$ there exists a $B_\epsilon(\vec{x}) = \{\vec{y} \in \Omega | \|\vec{x} - \vec{y}\| < \epsilon\}$ such that $B_\epsilon(\vec{x}) \subseteq \Omega$ if ϵ is small enough.

connected: if $\vec{x}, \vec{y} \in \Omega$, then there exists a steady curve $\vec{t}(s) : [0, 1] \to \Omega$ with $\vec{t}(0) = \vec{x}$, $\vec{t}(1) = \vec{y}$, $\vec{t}(s) \in \Omega$.

$\overline{\Omega}$ designates the closure of Ω, i.e. Ω plus the limit values of all sequences, which can be generated from elements of Ω.

$\partial\Omega = \overline{\Omega} \setminus \Omega$ is the boundary of Ω. Often additional conditions on the smoothness of the boundary are necessary.

Finally $\vec{\nu}(\vec{x})$ is the outer unit normal at a point $\vec{x} \in \partial\Omega$. $\qquad\square$

- $u : \Omega \to \mathbb{R}$ is called a solution of a PDE if it satisfies the PDE identically for every point $\vec{x} \in \Omega$

- Solutions of PDE's are usually not unique unless additional conditions are posed. Typically these are conditions for the function values (and/or derivatives) at the boundary

- A PDE is well posed if the solution
  - exists
  - is unique (with appropriate boundary conditions)
  - depends continuously on the data.

Linear partial PDE's of second order are a case of specific interest. For 2 dimensions and order $m = 2$ the general equation is:

$$a(x, y)\frac{\partial^2 u}{\partial x^2}(x, y) + 2b(x, y)\frac{\partial^2 u}{\partial x \partial y}(x, y) + c(x, y)\frac{\partial^2 u}{\partial y^2}(x, y)$$

$$+d(x, y)\frac{\partial u}{\partial x}(x, y) + e(x, y)\frac{\partial u}{\partial y}(x, y) + f(x, y)u(x, y)$$

$$+g(x, y) = 0$$

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

The general linear PDE of 2nd order in n space dimensions is:

$$\underbrace{\sum_{i,j=1}^{n} a_{ij}(\vec{x}) \partial_{x_i} \partial_{x_j} u}_{\text{main part}} + \sum_{i=1}^{n} a_i(\vec{x}) \partial_{x_i} u + a_0(\vec{x}) u = f(\vec{x}) \quad \text{in } \Omega.$$

without loss of generality one can set $a_{ij} = a_{ji}$ (as second derivatives are symmetric). With $(A(\vec{x}))_{ij} = a_{ij}(\vec{x})$ the PDE is at a point $\vec{x}$

elliptic if all eigenvalues of $A(\vec{x})$ have identical sign and no eigenvalue is zero.

hyperbolic if $(n-1)$ eigenvalues have identical sign, one eigenvalue the opposite sign and no eigenvalue is zero.

parabolic if one eigenvalue is zero, all other eigenvalues have identical sign and the $\mathrm{Rank}[A(\vec{x}), a(\vec{x})] = n$.

$\square$

- Why this classification? Different solution techniques are necessary for the different types of PDE's.

- The described classification is *complete* for linear PDE's with $n = m = 2$. In higher space dimensions the classification is no longer complete.

- The type is invariant under coordinate transformation $\xi = \xi(x,y)$, $\eta = \eta(x,y)$ and $u(x,y) = \tilde{u}(\xi(x,y), \eta(x,y))$, which yields a new PDE for $\tilde{u}(\xi, \eta)$ with the coefficients $\tilde{a}, \tilde{b}$, etc.. If the equation for u in (x,y) has the type t than $\tilde{u}$ in $(\xi(x,y), \eta(x,y))$ has the same type.

- The type *can* vary at different points.

- The type is only determined by the main part of the PDE (except for parabolic equations).

- Pathological cases like $\frac{\partial^2 u}{\partial x^2} + \frac{\partial u}{\partial x} = 0$ with the solution $u(x,y) = 0$ are avoided.

Definition 3.2. A linear PDE of 2nd order is called elliptic (hyperbolic, parabolic) in Ω if it is elliptic (hyperbolic, parabolic) for all points $(x,y) \in \Omega$. $\square$

Classification for first-order PDE's

Definition 3.3. An equation of the form

$$d(x,y)\frac{\partial u}{\partial x}(x,y) + e(x,y)\frac{\partial u}{\partial y}(x,y) + f(x,y)u(x,y) + g(x,y) = 0$$

is called hyperbolic if $|d(x,y)| \cdot |e(x,y)| > 0 \quad \forall (x,y) \in \Omega$.

For $n \geq 2$ the equation

$$\vec{v}(\vec{x}) \cdot \nabla u(\vec{x}) + f(\vec{x})u(\vec{x}) + g(\vec{x}) = 0$$

is called hyperbolic. $\square$

Non-linear PDE's, Systems of PDE's

- For non-linear PDE's of 2nd order (i.e. the coefficients a_{ij} and a_i can depend on the solution u) the type of the PDE can change in space *and* time.

- In this lecture we only cover scalar PDE's.

- Systems of PDE's contain several unknown functions

$$u_1, \ldots, u_n : \Omega \to \mathbb{R}$$

  and n (coupled) PDE's (e.g. in Two-Phase flow).

- There is also a classification system for systems of PDE's.

3.1 Examples for PDE types

Poisson-Equation

$$\frac{\partial^2 u}{\partial x^2}(x,y) + \frac{\partial^2 u}{\partial y^2}(x,y) = f(x,y) \qquad \forall (x,y) \in \Omega \tag{2}$$

is called Poisson-Equation.

This is the prototype of an *elliptic* PDE. The solution of equation (2) is not unique. If $u(x,y)$ is a solution, then e.g. $u(x,y) + c_1 + c_2 x + c_3 y$ is also a solution for arbitrary values of c_1, c_2, c_3. To get a unique solution u values at the boundary have to be specified (we therefore call this a "boundary value problem").

Two types of boundary values are common:

1. $u(x,y) = g(x,y)$ for $(x,y) \in \Gamma_D \subseteq \partial\Omega$ (Dirichlet[1]),

2. $\frac{\partial u}{\partial \nu}(x,y) = h(x,y)$ for $(x,y) \in \Gamma_N \subset \partial\Omega$ (Neumann[2], flux),

and $\Gamma_D \cup \Gamma_N = \partial\Omega$. It is also important that $\Gamma_N \neq \partial\Omega$, as else the solution is only defined up to a constant.

---

[1] Peter Gustav Lejeune Dirichlet, 1805-1859, German Mathematician.
[2] Carl Gottfried Neumann, 1832-1925, German Mathematician.

Complete Poisson-Equation
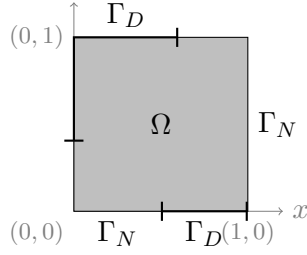
$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = f \text{ in } \Omega$$

$$u = g \text{ on } \Gamma_D \subseteq \partial\Omega$$

$$\frac{\partial u}{\partial \nu} = h \text{ on } \Gamma_N = \partial\Omega \setminus \Gamma_D \neq \partial\Omega$$

Generalisation to n space dimensions:

$$\sum_{i=1}^{n} \frac{\partial^2 u}{\partial x_i^2} =: \Delta u = f \text{ in } \Omega$$

$$u = g \text{ on } \Gamma_D \subseteq \partial\Omega$$

$$\nabla u \cdot \nu = h \text{ on } \Gamma_N = \partial\Omega \setminus \Gamma_D$$

This equation is also called elliptic. If $f \equiv 0$ it is called Laplace-Equation. $\qquad\square$

General Diffusion Equation

$K : \mathbb{R}^n \to \mathbb{R}^{n \times n}$ is a map, which relates to each point $\vec{x} \in \Omega$ a $n \times n$ matrix $K(\vec{x})$. We demand also (for all $\vec{x} \in \Omega$) that

1. $K(\vec{x}) = K^{\mathrm{T}}(\vec{x})$ and $\xi^{\mathrm{T}} K(\vec{x})\xi > 0 \ \ \forall \xi \in \mathbb{R}^n, \xi \neq 0$ (symmetric positive definite),

2. $C(\vec{x}) := \min\left\{\xi^{\mathrm{T}} K(\vec{x})\xi \,\middle|\, \|\xi\| = 1\right\} \geq C_0 > 0$ (uniform ellipticity).

$$-\nabla \cdot \left\{K(\vec{x})\nabla u(\vec{x})\right\} = f \text{ in } \Omega$$

$$u = g \text{ on } \Gamma_D \subseteq \partial\Omega \tag{3}$$

$$-\left(K(\vec{x})\nabla u(\vec{x})\right) \cdot \nu(\vec{x}) = h \text{ on } \Gamma_N = \partial\Omega \setminus \Gamma_D \neq \partial\Omega$$

is then called General Diffusion Equation (e.g. groundwater flow equation).

For strongly varying K equation (3) can be very difficult to solve. $\qquad\square$

Wave-Equation

The prototype of a hyperbolic equation of second order is the Wave-Equation:

$$\frac{\partial^2 u}{\partial x^2}(x,y) - \frac{\partial^2 u}{\partial y^2}(x,y) = 0 \qquad \text{in } \Omega \quad . \tag{4}$$

Possible boundary values for a domain $\Omega = (0,1)^2$ are e.g.:

$x \in [0, 1]$:

a) $u(x, 0) = u_0(x)$

b) $\frac{\partial u}{\partial y}(x, 0) = u_1(x)$

$y \in [0, 1]$:

c) $u(0, y) = g_0(y)$

d) $u(1, y) = g_1(y)$
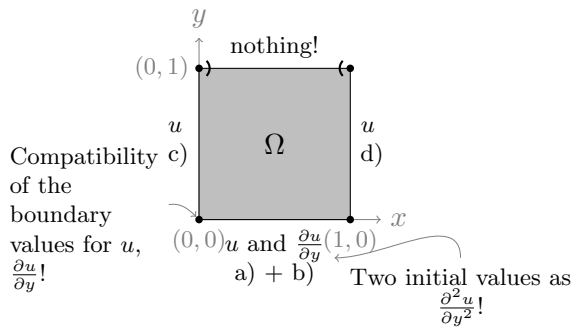
One direction (here y, usually the time) is special. a) + b) are called initial values and c) + d) boundary values (the boundary values can also be Neumann boundary conditions). It is not possible to prescribe values at the whole boundary (the future)! □

Heat-Equation

The prototype of a parabolic equation is the heat equation:

$$\frac{\partial^2 u}{\partial x^2}(x, y) - \frac{\partial u}{\partial y}(x, y) = 0 \qquad \text{in } \Omega.$$
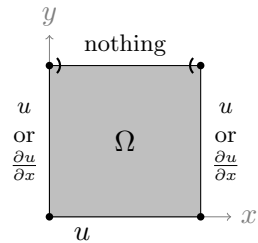
For a domain $\Omega = (0, 1)^2$ typical boundary values are (with $x \in [0, 1], y \in [0, 1]$):

$$u(x, 0) = u_0(x)$$

$$u(0, y) = g_0(y) \text{ or } \frac{\partial u}{\partial x}(0, y) = h_0(y)$$

$$u(1, y) = g_1(y) \text{ or } \frac{\partial u}{\partial x}(1, y) = h_1(y)$$

□

Transport-Equation

If $\Omega \subset \mathbb{R}^n, \vec{v}: \Omega \to \mathbb{R}^n$ is a given vector field, the equation

$$\nabla \cdot \{\vec{v}(\vec{x})u(\vec{x})\} = f(\vec{x}) \qquad \text{in } \Omega$$

is called stationary transport equation and is a hyperbolic PDE of first order. Possible boundary values are

$$u(\vec{x}) = g(\vec{x})$$
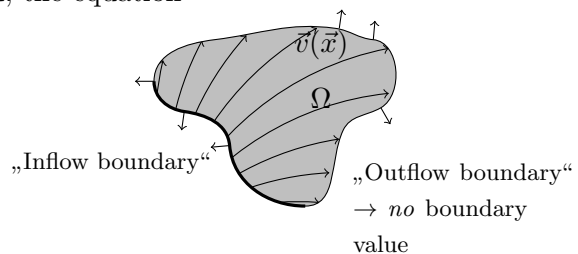
for $\vec{x} \in \partial\Omega$ with $\vec{v}(\vec{x}) \cdot \nu(\vec{x}) < 0$ (Boundary value depends on the flux field)

$\frac{\partial u}{\partial t} + \nabla \cdot \{\vec{v}(\vec{x}, t)u(\vec{x}, t)\} = f(\vec{x}, t)$ is also a hyperbolic PDE of first order. $\qquad\qquad\square$

3.2 Sphere of Influence

The type of a partial differential equation can also be illustrated with the following question:

Given $\vec{x} \in \Omega$. Which initial/boundary values influence the solution u at the point $\vec{x}$?

Elliptic $\quad u_{xx} + u_{yy} = 0$
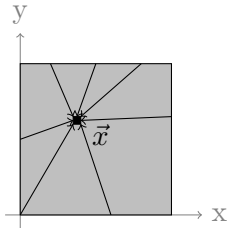
all boundary values influence $u(\vec{x})$, i.e. Change in $u(\vec{y}), \vec{y} \in \partial\Omega \Rightarrow$ Change in $u(\vec{x})$.

Parabolic $\quad u_{xx} - u_y = 0 \qquad$ Note: The $-$ is crucial, $+$ is parabolic according to the definition
but it is not well posed (stable)
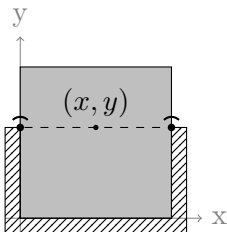
for (x, y) all (x', y') with $y' \le y$ influence the value at $\vec{x}$.
„infinite velocity of propagation"

Hyperbolic (2nd order) $\quad u_{xx} - u_{yy} = 0$
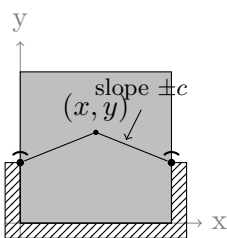
Solution at (x, y) is influenced by all boundary values below the cone

$$\{(x', y') \quad | \quad y' \le (x' - x) \cdot c + y$$
$$\wedge \quad y' \le (x - x') \cdot c + y\} \quad \cap \quad \partial\Omega$$

„finite velocity of propagation"

Hyperbolic (1st order) $\quad u_x + u_y = 0$
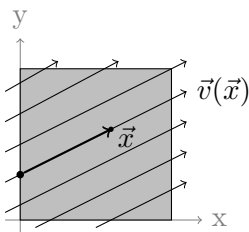
Only one boundary point influences the value.

- The steady-state groundwater flow equation $-\nabla \cdot \left[\bar{K}_s(\vec{x}) \cdot (\nabla p_w - \rho_w g \vec{e}_z) \right] + r_w(\vec{x}) = 0$ is an elliptic partial differential equation of second order.

- To get a well posed problem either Dirichlet boundary conditions (the pressure value is given) or Neumann boundary conditions (the flux is given) must be specified at each boundary point.

- At one point of the boundary a Dirichlet boundary condition should be specified (else the equation is only defined up to a constant).

- Each point in the domain is influenced by all boundary conditions.

4 Spatial-Discretisation Methods

- Partial differential equations can only be solved analytically for very special cases with a very restricted choice of domain shapes, boundary conditions and parameter fields.

- Approximations can be calculated with numerical methods

- Numerical methods usually yield approximations of
  - the solution at certain points in space (e.g. Finite Differences)
  - the solution with a parameterised function (e.g. Finite Elements, Discontinuous Galerkin . . .)
  - certain mathematical properties (mass conservation, continuity of fluxes) of the equation (e.g. Finite Volumes, Mimetic Finite Differences)

4.1 Grids

- For most discretisation schemes it is necessary to partition the domain Ω into sub-domains (elements) e with a simple geometrical structure (triangulation).

- Typical element geometries are:

  1D line segments

  2D triangle, quadrilateral

  3D tetrahedron, cuboid, pyramid, prism, hexahedron

- All the elements together are called a grid.

- It is not always possible to fill the whole domain with elements of a simple geometry, but there should be no holes in the grid and $\bigcup_{i=1}^{n} e_i \approx \bar{\Omega}$

There are different varieties of grids depending on the purpose and the numerical scheme. Grids can be

structured is constructed with regular elements from a simple construction principle. Typical examples are grids with rectangular elements with a width which is

 equidistant element width is h_i in dimension $i \in \{x, y, z\}$

 tensor product element width is $h_i = f(x_i)$ in dimension $i \in \{x, y, z\}$

unstructured can be composed of elements with different geometries and shapes

conforming there are no hanging nodes, i.e. if the intersection $e_i \cap e_j$ between two elements e_i and e_j is a
- point they have a common node
- a line they have a common edge
- a surface they have a common face

non-conforming there are hanging nodes, i.e. nodes of one element, which are not nodes of an element with which an intersection exists
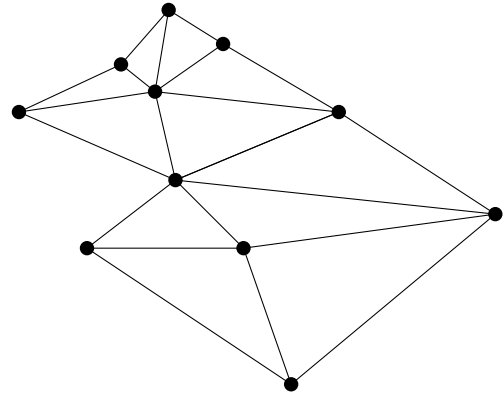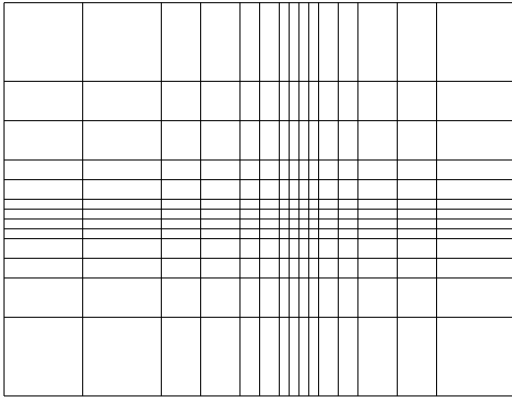
Figure 6: Structured (tensor-product) grid (left) and unstructured grid (right).
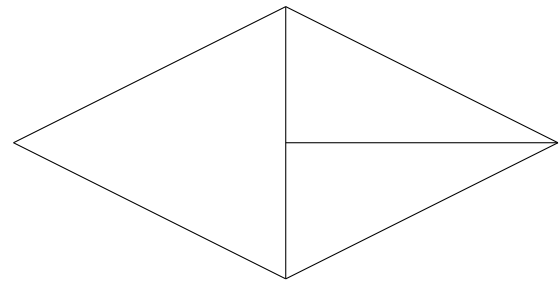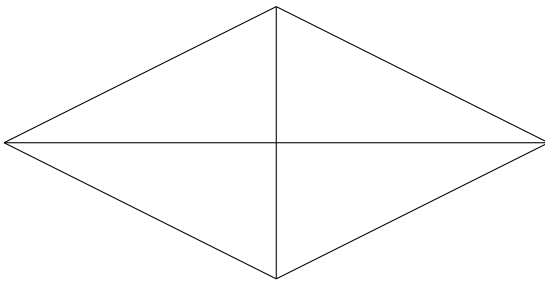
Figure 7: Conforming grid (left) and non-conforming grid (right).

4.2 The Finite Difference Method

Basic Idea: Partial derivatives are replaced with difference quotients (Taylor series expansion)
Let us use the one-dimensional Poisson equation as example:

$$-\frac{\partial^2 u}{\partial x^2} = f(x) \qquad x \in (0,1)$$
$$u(0) = \varphi_0, \quad u(1) = \varphi_1.$$

We do a Taylor expansion of $u(x + h)$:

$$u(x + h) = u(x) + hu'(x) + \frac{h^2}{2}u''(x + \vartheta^+ h) \qquad \vartheta^+ \in (0,1)$$

$$\Longleftrightarrow \; u'(x) = \frac{u(x + h) - u(x)}{h} - \underbrace{\left(\frac{h}{2}u''(x + \vartheta^+ h)\right)}_{O(h)} \qquad \vartheta^+ \in (0,1)$$

This is a first order accurate approximation of the gradient of u.

If we do an expansion up to the fourth order terms of $u(x+h)$ and $u(x-h)$

$$u(x+h) = u(x) + hu'(x) + \frac{h^2}{2}u''(x) + \frac{h^3}{6}u'''(x) + \frac{h^4}{24}u''''(x+\vartheta^+ h)$$

$$u(x-h) = u(x) - hu'(x) + \frac{h^2}{2}u''(x) - \frac{h^3}{6}u'''(x) + \frac{h^4}{24}u''''(x-\vartheta^- h)$$

we get the second order accurate formula for gradient u

$$u(x+h) - u(x-h) = 2hu'(x) + \frac{h^3}{6}\left\{u'''(x+\vartheta^+ h) + u'''(x-\vartheta^- h)\right\}$$

$$\iff u'(x) = \frac{u(x+h) - u(x-h)}{2h} \quad \underbrace{-\left(\frac{h^2}{12}\left\{u'''(x+\vartheta^+ h) + u'''(x-\vartheta^- h)\right\}\right)}_{O(h^2)}$$

and the second order accurate approximation of the second derivative of u:

$$u(x+h) + u(x-h) = 2u(x) + h^2 u''(x) + \frac{h^4}{24}\left\{u''''(x+\vartheta^+ h) + u''''(x-\vartheta^- h)\right\}$$

$$\iff u''(x) = \frac{u(x-h) - 2u(x) + u(x+h)}{h^2} \quad \underbrace{-\left(\frac{h^2}{24}\left\{\dots\right\}\right)}_{O(h^2)}$$

If we insert this in our partial differential equation we get for $x_i = i \cdot h$

$$-\frac{\partial^2 u(x_i)}{\partial x^2} \approx -\frac{u(x_{i-1}) - 2u(x_i) + u(x_{i+1})}{h^2} = f(x_i)$$

one equation per grid point. Dirichlet boundary conditions can be easily incorporated by setting $u_0 = \varphi_0$ and $u_n = \varphi_1$ and bringing the corresponding terms to the right hand side.

Application to two-dimensional Poisson Equation

In 2D the Poisson equation is

$$-\Delta u(x) = f(x)$$

and we get for $x_i = i \cdot h$ and $y_j = j \cdot h$ with

$$\Delta u(x_i, y_j) \approx \frac{u(x_{i-1}, y_j) - 2u(x_i, y_j) + u(x_{i+1}, y_j)}{h^2} + \frac{u(x_i, y_{j-1}) - 2u(x_i, y_j) + u(x_i, y_{j+1})}{h^2}$$

for each grid point the linear equation

$$\frac{4u(x_i, y_j) - u(x_{i-1}, y_j) - u(x_{i+1}, y_j) - u(x_i, y_{j-1}) - u(x_i, y_{j+1})}{h^2} = f(x_i, y_j)$$

- Dirichlet boundary conditions can easily be integrated by rearranging the equation systems and bringing them to the right side of the equation.

- Neumann boundary conditions are integrated by either replacing them with a forward difference formula or by introduction of ghost nodes

Rate of Convergence

The approximation error $e = ||u - u_h||$ of the approximated solution u_h on a grid with element width h is proportional to the size of the grid cells.

- We get a *linear grid convergence* if

$$\lim_{h \to 0} \frac{e_{i+1}}{e_i} = \lim_{h \to 0} \frac{||u - u_{h_{i+1}}||}{||u - u_{h_i}||} \leq C \cdot \frac{h_{i+1}}{h_i}$$

- We get a *grid convergence of order q* if

$$\lim_{h \to 0} \frac{e_{i+1}}{e_i} = \lim_{h \to 0} \frac{||u - u_{h_{i+1}}||}{||u - u_{h_i}||} \leq C \cdot \left(\frac{h_{i+1}}{h_i} \right)^q$$

- It can be proved that the Finite Difference Method is second-order accurate on an equidistant grid if the solution is regular enough

Properties of the Finite Difference Method

- Advantages:
  - easy to formulate and implement
  - well suited for structured grids

- Problems:
  - Only linear convergence rate on non-equidistant grids
  - What's the value between two points?
  - Representation of complex domains difficult
  - In general not (locally) mass-conservative.

4.3 The Finite Element Method

- A parameterised trial function $y(x)$ is inserted in the partial differential equation, resulting in a residual.

- The trial function is build as a sum over products of base functions times parameters $y(x) = \sum\limits_{i=1}^{n} c_i \cdot \psi_i(x)$

- We would like to choose the parameters c_i of the trial function to minimise the error between the approximation and the correct solution. As the latter is unknown this is not possible

- For the correct solution the partial differential equation is zero:

$$F \left(\frac{\partial^m u}{\partial x_1^m}(\vec{x}), \frac{\partial^{m-1} u}{\partial x_1^{m-1}}(\vec{x}), \dots, \frac{\partial^m u}{\partial x_1^{m-1} \partial x_2}(\vec{x}), \dots, \frac{\partial^m u}{\partial x_n^m}(\vec{x}), \frac{\partial^{m-1} u}{\partial x_n^{m-1}}(\vec{x}), \dots, u(\vec{x}), \vec{x} \right) = 0 \quad \forall \vec{x} \in \Omega$$

$$(5)$$

- We demand that the partial differential equation F should only be fulfilled in the integral over Ω. For generality we multiply F with a weighting function w. The weighting function has to be zero at the boundaries. One obtains:

$$\iiint_{\Omega} F\left(\frac{\partial^m u}{\partial x_1^m}(\vec{x}), \frac{\partial^{m-1} u}{\partial x_1^{m-1}}(\vec{x}), \dots, \frac{\partial^m u}{\partial x_1^{m-1}\partial x_2}(\vec{x}), \dots, \frac{\partial^m u}{\partial x_n^m}(\vec{x}), \frac{\partial^{m-1} u}{\partial x_n^{m-1}}(\vec{x}), \dots, u(\vec{x}), \vec{x}\right) \cdot w(\vec{x}) \, dV = 0$$

(6)

  For a solution of the PDE this is obviously always fulfilled. However, it is a weaker condition, as it can also be true for functions which fulfil the partial differential equation only "on an average". Thus we call this a *weak formulation*.

- To reduce the computational costs and increase the flexibility, the base functions are defined element wise, i.e. for each element there is a set of base functions, which is different from zero on this element, but zero on almost all other elements. We get:

$$y(x) = \sum_{e_i} \sum_{j=1}^{n_{e_i}} c_{e_i,j} \cdot \psi_{e_i,j}(x)$$

- This allows the integral over the whole domain to be replaced with a sum over integrals over each of the elements

- Different Finite Element methods differ in the choice of the trial and weighting functions.

- Usually the trial function on an element is parameterised with the value of the trial function at certain positions (the nodes, additionally on edges or faces) and the base chosen to be one at one of these positions and zero at all others (similar to Lagrange interpolation). For a conforming grid this guarantees a solution which is steady over element boundaries.

- The base functions are defined on a reference element and scaled to the real geometry

Example: One-dimensional Poisson Equation

$$-\frac{\partial^2 y(x)}{\partial x^2} = f(x) \qquad \text{in } (0,1)$$

with the boundary conditions $y(0) = 0$ and $y(1) = 0$

We use an equidistant grid $\vdash\!\!+\!\!+\!\!+\!\!+\!\!+\!\!+\!\!+\!\!+\!\!\dashv$ $\qquad\qquad x_i = i \cdot h \quad i = 0, \dots, n, \quad h = \frac{1}{n}$

Example: One-dimensional Poisson Equation

As base functions we use the hat functions $\psi_i, \quad i = 1, \dots, n-1$
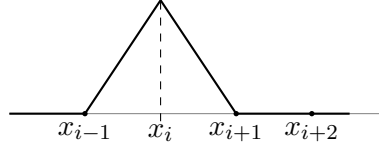
$$\psi_i(x) = \begin{cases} \frac{x-x_{i-1}}{x_i-x_{i-1}} = \frac{x-x_{i-1}}{h} & x \in (x_{i-1}, x_i) \\ \frac{x-x_{i+1}}{x_i-x_{i+1}} = -\frac{x-x_{i+1}}{h} & x \in (x_i, x_{i+1}) \\ 0 & \text{else} \end{cases}$$

with the special property:

$$\psi_i(x_j) = \begin{cases} 1 & i = j \\ 0 & \text{else} \end{cases}$$

For each weighting functions w_i we get:

$$-\int_0^1 \frac{\partial^2 y}{\partial x^2} \cdot w_i \, dx = \int_0^1 f \cdot w_i \, dx$$

with partial integration:

$$-\left[-\int_0^1 \frac{\partial y}{\partial x} \cdot \frac{\partial w_i}{\partial x} \, dx + \frac{\partial y}{\partial x}(1) w_i(1) - \frac{\partial y}{\partial x}(0) w_i(0) \right] = \int_0^1 f \cdot w_i \, dx$$

as $w_i(x)$ is choosen to be zero at the boundary we get

$$\int_0^1 \frac{\partial y}{\partial x} \cdot \frac{\partial w_i}{\partial x} \, dx = \int_0^1 f \cdot w_i \, dx$$

Galerkin Method
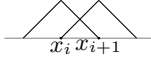
In the so called Galerkin Method we use the base functions $\psi_i(x)$ also as trial functions.

With $y(x) = \sum_i y_i \psi_i(x)$ we get for each weighting function one line of a linear equation system:

$$y_{i-1} \int_{x_{i-1}}^{x_i} \frac{\partial \psi_{i-1}}{\partial x} \cdot \frac{\partial \psi_i}{\partial x} dx + y_i \int_{x_{i-1}}^{x_{i+1}} \frac{\partial \psi_i}{\partial x} \cdot \frac{\partial \psi_i}{\partial x} dx + y_{i+1} \int_{x_i}^{x_{i+1}} \frac{\partial \psi_{i+1}}{\partial x} \cdot \frac{\partial \psi_i}{\partial x} dx = \int_{x_{i-1}}^{x_{i+1}} f \cdot \psi_i dx$$

All other terms are zero as the weighting function $\psi_i(x)$ is zero outside the interval (x_{i-1}, x_{i+1}).

The integrals can be evaluated to:

$$\int \frac{\partial \psi_i}{\partial x} \cdot \frac{\partial \psi_k}{\partial x} dx = \begin{cases} \int_{x_i-h}^{x_i} \frac{1}{h} \cdot \frac{1}{h} dx + \int_{x_i}^{x_i+h} (-\frac{1}{h}) \cdot (-\frac{1}{h}) dx = \frac{1}{h} + \frac{1}{h} = \frac{2}{h} & k = i \\[2ex] \int_{x_i}^{x_i+h} (-\frac{1}{h}) \cdot \frac{1}{h} dx = -\frac{1}{h} & k = i \pm 1 \\[4ex] & \text{\includegraphics} \quad \overline{x_i \; x_{i+1}} \\[2ex] 0 & \text{else} \end{cases}$$

If we integrate the right hand side with the trapezoidal rule we get

$$\int_{x_{i-1}}^{x_{i+1}} f \cdot \psi_i dx \approx \frac{h}{2} \cdot (0 \cdot f_{i-1} + f_i + f_i + 0 \cdot f_{i+1}) = h \cdot f_i$$

and finally the linear equation

$$\frac{1}{h}(-y_{i-1} + 2y_i - y_{i+1}) = h \cdot f_i$$

For one-dimensional equidistant grids we get exactly the same solution for the Finite-Difference and the Finite-Element discretisation:

$$-\frac{u(x_{i-1}) - 2u(x_i) + u(x_{i+1})}{h^2} = f(x_i)$$

$$\frac{1}{h}(-y_{i-1} + 2y_i - y_{i+1}) = h \cdot f_i$$

- Dirichlet boundary conditions can be directly incorporated into the trial functions.

- Neumann boundary conditions are handled in the integrals and result in terms on the right hand side.

- Convergence order depends on the choice of weight and trial functions.

- Often the integrations are performed with numerical integration. If an integration rule of sufficient order is used, the full convergence order of the Finite-Element method is achieved.

Properties of the Finite Element Method

- Advantages:
  - can be used for domains with complicated shape
  - yields function values everywhere
  - well suited for unstructured grids
  - local adaptivity possible

- Problems:
  - grid generation can be complicated (must often fullfill certain conditions)
  - more computationally expensive for simple problems
  - not always (locally) mass-conservative
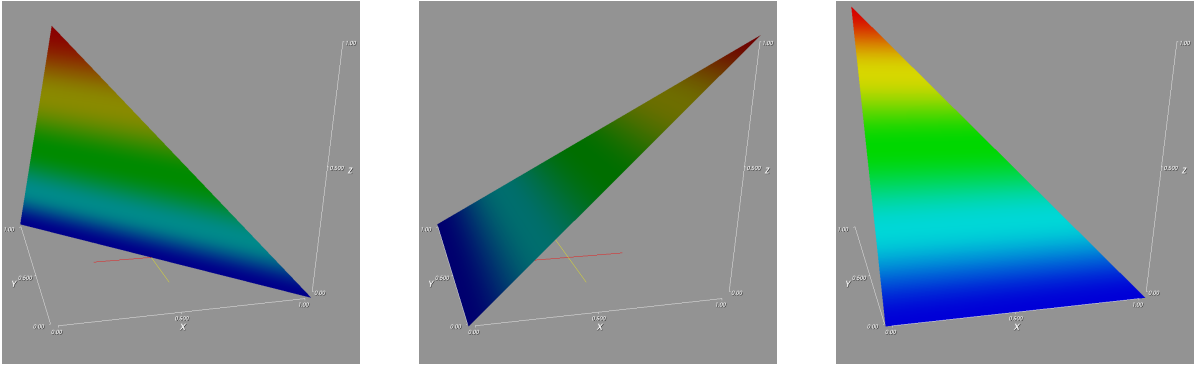
Figure 8: First Order base functions on the reference element.
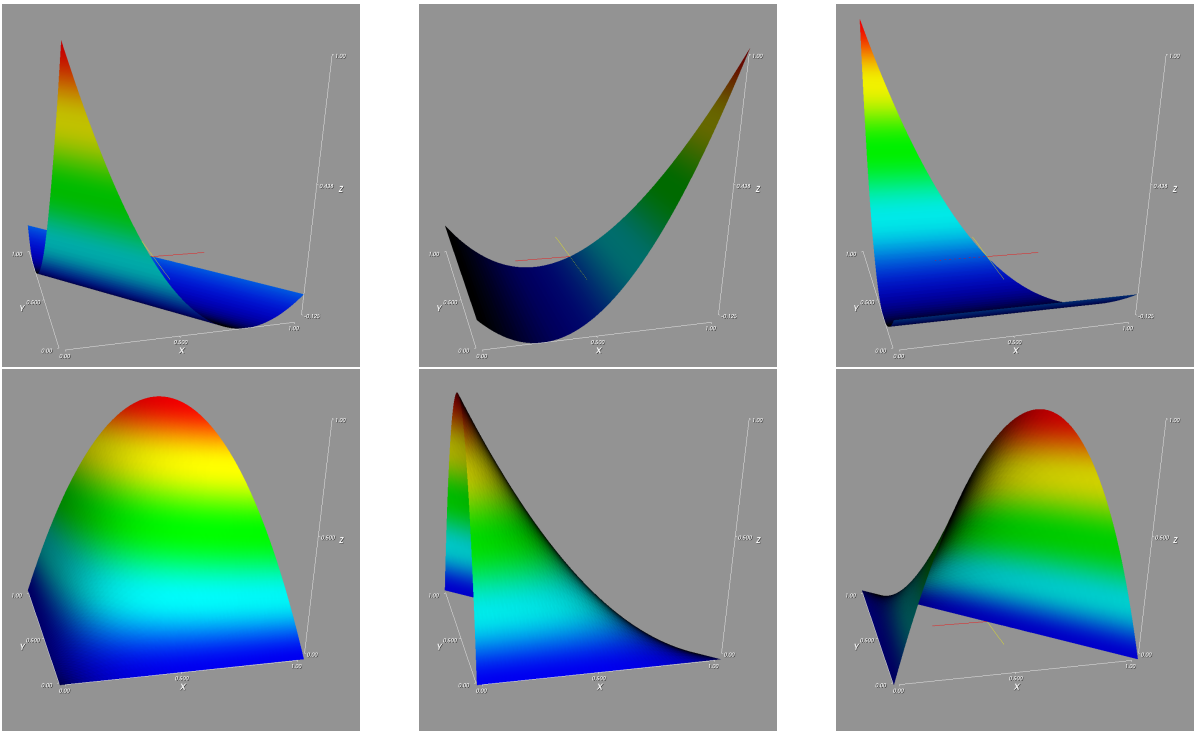
Figure 9: Second order base functions on the reference element.

4.4 Discontinuous Galerkin Scheme

Discontinuous Galerkin Scheme

- In the same way as in the Finite-Element method we formulate a weak solution and replace it by a sum of integrals over the elements. For the Poisson equation with zero Dirichlet boundary conditions we get with $\vec{j} = -\nabla u$:

$$\int_\Omega w \nabla \cdot \vec{j}\, dV \approx \sum_{e_i \in \Omega} \int_{e_i} w \nabla \cdot \vec{j}\, dV = \sum_{e_i \in \Omega} \int_{e_i} wf\, dV$$

- We allow the trial functions v and test functions w to be discontinuous at element boundaries

- Cell-wise integration by parts yields

$$\sum_{e_i \in \Omega} \left(-\int_{e_i} \vec{j} \cdot \nabla w\, dV + \int_{\partial e_i} w\vec{j} \cdot \vec{n}\, ds \right) =$$
$$-\sum_{e_i \in \Omega} \int_{e_i} \vec{j} \cdot \nabla w\, dV + \sum_{\Gamma_{\mathsf{ef}} \subseteq \Gamma_{\mathsf{int}}} \int_{\Gamma_{\mathsf{ef}}} \left[w\vec{j} \cdot \vec{n}_{\mathsf{ef}} \right] ds$$

Jumps and Averages

- We introduce the jump and the average of a function over a boundary:

$$\begin{aligned}
[v]_{ef}(\vec{x}) &= v|_{(\partial e \cap \Gamma_{\mathsf{ef}})}(\vec{x}) - v|_{(\partial f \cap \Gamma_{\mathsf{ef}})}(\vec{x}), & e > f \\
\langle v \rangle_{ef}(\vec{x}) &= {}^1\!/_2 \left(v|_{(\partial e \cap \Gamma_{\mathsf{ef}})}(\vec{x}) + v|_{(\partial f \cap \Gamma_{\mathsf{ef}})}(\vec{x}) \right), & e > f
\end{aligned}$$

- It can be shown that for the product of two functions the following equality holds:

$$\left[w\vec{j} \cdot \vec{n}_{\mathsf{ef}} \right] = [w]\langle \vec{j} \cdot \vec{n}_{\mathsf{ef}} \rangle + \langle w \rangle [\vec{j} \cdot \vec{n}_{\mathsf{ef}}]$$

- The second term is zero for the exact solution of the PDE. Thus we can subtract it without changing the result for the exact solution.

- We get

$$\int_\Omega w \nabla \cdot \vec{j}\, dV \approx \sum_{e_i \in \Omega} \int_{e_i} \nabla u \cdot \nabla w\, dV - \sum_{\Gamma_{\mathsf{ef}} \subseteq \Gamma_{\mathsf{int}}} \int_{\Gamma_{\mathsf{ef}}} [w]\langle \nabla u \cdot \vec{n}_{\mathsf{ef}} \rangle\, ds$$

Interior Penalty Terms

- We can add an additional term to penalize jumps in the solution. It is also zero for the exact solution and can make the result symmetric:

$$\pm \sum_{\Gamma_{\mathsf{ef}} \subseteq \Gamma_{\mathsf{int}}} \int_{\Gamma_{\mathsf{ef}}} \langle \nabla w \cdot \vec{n}_{\mathsf{ef}} \rangle [u]\, ds$$

- With the positive sign we get the Oden Babuschka Baumann Discontinuous Galerkin scheme (OBB)

$$\sum_{e_i \in \Omega} \int_{e_i} \nabla u \cdot \nabla w \, dV$$

$$- \sum_{\Gamma_{\mathsf{ef}} \subseteq \Gamma_{\mathsf{int}}} \int_{\Gamma_{\mathsf{ef}}} \Big([w] \langle \nabla u \cdot \vec{n}_{\mathsf{ef}} \rangle - \langle \nabla w \cdot \vec{n}_{\mathsf{ef}} \rangle [u] \Big) ds$$

$$= \sum_{e_i \in \Omega} \int_{e_i} w f \, dV$$

Stabilisation Term

- To get coercivity which is a sufficient condition for the well-posedness of the problem we add another term which is zero for the exact solution (σ_{ef} is a element size dependent positive scalar). We then get the Non-symmetric Interior Penalty Discontinuous Galerkin scheme (NIPG):

$$\sum_{e_i \in \Omega} \int_{e_i} \nabla u \cdot \nabla w \, dV$$

$$- \sum_{\Gamma_{\mathsf{ef}} \subseteq \Gamma_{\mathsf{int}}} \int_{\Gamma_{\mathsf{ef}}} \Big([w] \langle \nabla u \cdot \vec{n}_{\mathsf{ef}} \rangle - \langle \nabla w \cdot \vec{n}_{\mathsf{ef}} \rangle [u] - \sigma_{ef} [w][u] \, ds \Big) ds$$

$$= \sum_{e_i \in \Omega} \int_{e_i} w f \, dV$$

- If we take the interior penalty term with the negative sign, we get the Symmetric Interior Penalty Discontinuous Galerkin scheme (SIPG). The additional stabilisation term is then mandatory (i.e. ($\sigma_{ef} > 0$)):

$$\sum_{e_i \in \Omega} \int_{e_i} \nabla u \cdot \nabla w \, dV$$

$$- \sum_{\Gamma_{\mathsf{ef}} \subseteq \Gamma_{\mathsf{int}}} \int_{\Gamma_{\mathsf{ef}}} \Big([w] \langle \nabla u \cdot \vec{n}_{\mathsf{ef}} \rangle + \langle \nabla w \cdot \vec{n}_{\mathsf{ef}} \rangle [u] - \sigma_{ef} [w][u] \Big) ds$$

$$= \sum_{e_i \in \Omega} \int_{e_i} w f \, dV$$

Additional Remarks

- Dirichlet boundary conditions are only weakly enforced by penalty terms

- Neumann boundary conditions can be handled in the boundary integrals

- Different Discontinuous Galerkin Schemes can be obtained with different trial and test functions

- DG schemes usually have more degrees of freedom than Standard Finite Element schemes, as there is an independent set of of parameters for the base functions for each element. Each element edge(2D)/face(3D) produces two (dense) blocks of coefficients in the matrix

- The order of convergence depends on the variant of the DG scheme and the selection of the test and trial functions

Properties of the Discontinuous Galerkin Scheme

- Advantages:
  - can be used for domains with complicated shape
  - yield function values everywhere
  - well suited for unstructured grids
  - local adaptivity possible
  - locally mass-conservative

- Problems:
  - grid generation can be complicated (must often fullfill certain conditions)
  - more computationally expensive for simple problems
  - usually produce non-steady solution

4.5 Cell-Centred Finite-Volume Method

We want to discretise the steady-state ground-water equation

$$\nabla \cdot \vec{J}_w(\vec{x}) + r_w(\vec{x}) = 0$$

with

$$J_w = -K_s(\vec{x}) \nabla p_w$$

with the Cell-Centred Finite-Volume method.

First we divide grid into rectangular grid cells g_{ij}
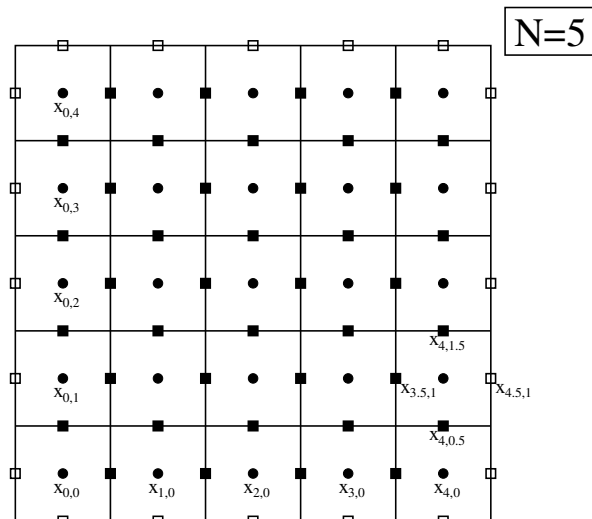
We demand that the integral of the partial differential equation over each grid cell is fulfilled:

$$\int\limits_{g_{ij}} \nabla \cdot \vec{J}_w \ dx \, dy = \int\limits_{g_{ij}} r(\vec{x}) \ dx \, dy$$

and use the Satz of Gauss to transform the volume integral over the divergence of the flux into a boundary integral over the flux normal to the boundary:

$$\underbrace{\Leftrightarrow}_{Satz\,of\,Gauss} \int\limits_{\partial g_{ij}} \vec{J}_w \cdot \vec{n} \ ds = \int\limits_{g_{ij}} r(\vec{x}) \ dx \, dy$$

Let us look at an inner grid cell:
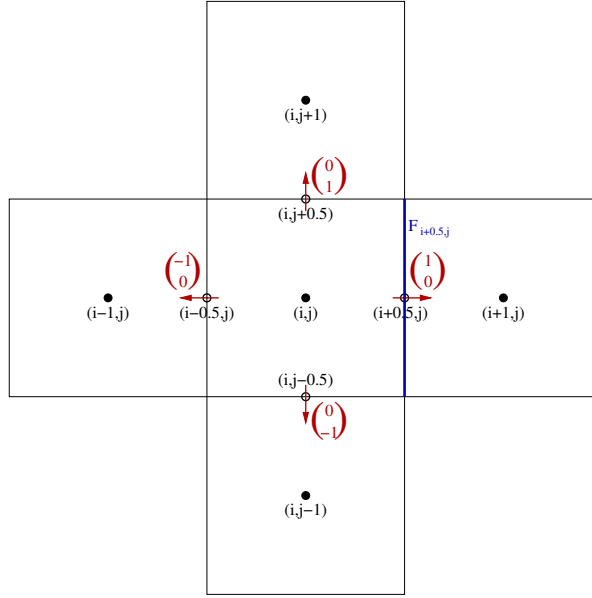
For our rectangular cell, we can split the integral over the boundary of the cell into integrals over each face

$$\int\limits_{\partial g_{ij}} \vec{J}_w \cdot \vec{n} \ ds = \sum_{k=i\pm0.5} \int\limits_{F_{kj}} \vec{J}_w \cdot \vec{n} \ ds + \sum_{l=j\pm0.5} \int\limits_{F_{il}} \vec{J}_w \cdot \vec{n} \ ds$$

and approximate the integral over each face with the Midpoint rule

$$\underbrace{\approx}_{\text{Midpointrule}} \sum_{k=i\pm0.5} \vec{J}_w(\vec{x}_{k,j}) \cdot \vec{n} \cdot \underbrace{h}_{\text{Face Area}} + \sum_{l=j\pm0.5} \vec{J}_w(\vec{x}_{i,l}) \cdot \vec{n} \cdot \underbrace{h}_{\text{Face Area}}$$

If the permeability is a diagonal matrix the flux over a face depends only on the gradient in the normal direction

$$\vec{J}_w(\vec{x}) = - \begin{pmatrix} K_{xx}(\vec{x}) & 0 \\ 0 & K_{yy}(\vec{x}) \end{pmatrix} \cdot \begin{pmatrix} \frac{\partial p}{\partial x}(\vec{x}) \\ \frac{\partial p}{\partial y}(\vec{x}) \end{pmatrix}$$

The multiplication with the (normalised) normal vector only influences the sign of the flux integral

$$\sum_{k=i\pm0.5} \vec{J}_w(\vec{x}_{k,j}) \cdot \vec{n} \cdot h + \sum_{l=j\pm0.5} \vec{J}_w(\vec{x}_{i,l}) \cdot \vec{n} \cdot h =$$

$$-K_{xx}(\vec{x}_{i-0.5,j}) \cdot \frac{\partial p}{\partial x}(\vec{x}_{i-0.5,j}) \cdot \underbrace{(-1)}_{\text{from } n_x} \cdot h$$

$$-K_{xx}(\vec{x}_{i+0.5,j}) \cdot \frac{\partial p}{\partial x}(\vec{x}_{i+0.5,j}) \cdot \underbrace{(1)}_{\text{from } n_x} \cdot h$$

$$-K_{yy}(\vec{x}_{i,j-0.5}) \cdot \frac{\partial p}{\partial x}(\vec{x}_{i,j-0.5}) \cdot \underbrace{(-1)}_{\text{from } n_y} \cdot h$$

$$-K_{yy}(\vec{x}_{i,j+0.5}) \cdot \frac{\partial p}{\partial x}(\vec{x}_{i,j+0.5}) \cdot \underbrace{(1)}_{\text{from } n_y} \cdot h$$

The gradient at the face midpoint is approximated by a central difference quotient

$$\underbrace{\approx}_{\text{approx.Derivative}} \quad +K_{xx}(\vec{x}_{i-0.5,j}) \cdot \frac{p(\vec{x}_{i,j}) - p(\vec{x}_{i-1,j})}{h} \cdot h$$

$$-K_{xx}(\vec{x}_{i+0.5,j}) \cdot \frac{p(\vec{x}_{i+1,j}) - p(\vec{x}_{i,j})}{h} \cdot h$$

$$+K_{yy}(\vec{x}_{i,j-0.5}) \cdot \frac{p(\vec{x}_{i,j}) - p(\vec{x}_{i,j-1})}{h} \cdot h$$

$$-K_{yy}(\vec{x}_{i,j+0.5}) \cdot \frac{p(\vec{x}_{i,j+1}) - p(\vec{x}_{i,j})}{h} \cdot h$$

The integration of the source/sink term is also done with the midpoint rule:

$$\int_{g_{ij}} r(\vec{x}) dx\, dy \approx h^2 r(\vec{x}_{i,j})$$

We get one line of a linear equation system for each grid cell:

$$-K_{xx}(\vec{x}_{i-0.5,j}) \cdot p_{i-1,j} - K_{xx}(\vec{x}_{i+0.5,j}) \cdot p_{i+1,j}$$
$$-K_{yy}(\vec{x}_{i,j-0.5}) \cdot p_{i,j-1} - K_{yy}(\vec{x}_{i,j+0.5}) \cdot p_{i,j+1}$$
$$+[K_{xx}(\vec{x}_{i-0.5,j}) + K_{xx}(\vec{x}_{i+0.5,j}) + K_{yy}(\vec{x}_{i,j-0.5}) + K_{yy}(\vec{x}_{i,j+0.5})] \cdot p_{i,j} = h^2 r(\vec{x}_{i,j})$$

Dirichlet Boundary Conditions

Let us assume that at $x = 0$ there is a Dirichlet boundary:
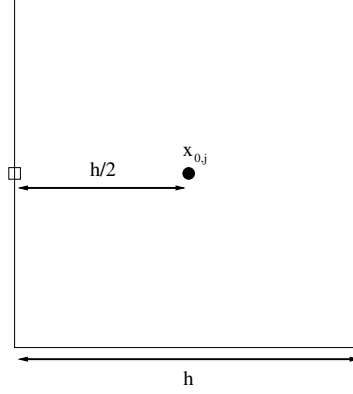
The derivative between the face midpoint and the element midpoint can be approximated by a difference quotient (only first order):

$$\frac{\partial p}{\partial x}(\vec{x}_{-0.5,j}) \approx \frac{p(\vec{x}_{0,j}) - p_d(0, y_j)}{h/2}$$

The constant term $-K_{xx}(\vec{x}_{i-0.5,j}) \cdot p_d(0, y_j)$ is brought to the right-hand side of the equation:

$$
\begin{aligned}
-K_{xx}(\vec{x}_{i+0.5,j}) \cdot p_{i+1,j} & \\
-K_{yy}(\vec{x}_{i,j-0.5}) \cdot p_{i,j-1} - K_{yy}(\vec{x}_{i,j+0.5}) \cdot p_{i,j+1} & \\
+ [2K_{xx}(\vec{x}_{i-0.5,j}) + K_{xx}(\vec{x}_{i+0.5,j}) & \\
+ K_{yy}(\vec{x}_{i,j-0.5}) + K_{yy}(\vec{x}_{i,j+0.5})] \cdot p_{i,j} & = h^2 r(\vec{x}_{i,j}) + 2K_{xx}(\vec{x}_{i-0.5,j}) \cdot p_d(0, y_j)
\end{aligned}
$$

Neumann Boundary Conditions

To integrate Neumann boundary conditions we go back to the point before the integration of the face fluxes with the midpoint rule. For each face we had to determine

$$\int_F \vec{J}_w \cdot \vec{n} \; ds$$

At a Neumann boundary $\vec{J}_w \cdot \vec{n}$ is given directly by the boundary condition $\phi_n(\vec{x})$, we can therefore use

$$\int_{F_{kl}} \vec{J}_w \cdot \vec{n} \; ds \underbrace{\approx}_{\text{Midpoint rule}} h \cdot \vec{\phi}_N(\vec{x}) \cdot \vec{n}$$

at each Neumann boundary.

We transfer the constant term $h \cdot \phi_N(\vec{x}_{-0.5,j})$ to the right hand side:

$$
\begin{aligned}
-K_{xx}(\vec{x}_{i+0.5,j}) \cdot p_{i+1,j} & \\
-K_{yy}(\vec{x}_{i,j-0.5}) \cdot p_{i,j-1} - K_{yy}(\vec{x}_{i,j+0.5}) \cdot p_{i,j+1} & \\
+ [K_{xx}(\vec{x}_{i+0.5,j}) + K_{yy}(\vec{x}_{i,j-0.5}) + K_{yy}(\vec{x}_{i,j+0.5})] \cdot p_{i,j} & = h^2 r(\vec{x}_{i,j}) - h \cdot \phi_N(\vec{x}_{-0.5,j})
\end{aligned}
$$

Different Grid Spacing in x and y direction

If the grid spacing in x and y direction is different, the h factors can not vanish:

$$-\frac{h_y}{h_x}\left(K_{xx}(\vec{x}_{i-0.5,j})\cdot p_{i-1,j} - K_{xx}(\vec{x}_{i+0.5,j})\cdot p_{i+1,j}\right)$$

$$-\frac{h_x}{h_y}\left(K_{yy}(\vec{x}_{i,j-0.5})\cdot p_{i,j-1} - K_{yy}(\vec{x}_{i,j+0.5})\cdot p_{i,j+1}\right)$$

$$+\left[\frac{h_y}{h_x}\left(K_{xx}(\vec{x}_{i-0.5,j}) + K_{xx}(\vec{x}_{i+0.5,j})\right)\right.$$

$$\left.+\frac{h_x}{h_y}\left(K_{yy}(\vec{x}_{i,j-0.5}) + K_{yy}(\vec{x}_{i,j+0.5})\right)\right]\cdot p_{i,j} \quad = \quad h_x h_y r(\vec{x}_{i,j})$$

Example: 3×3 Grid

Let us perform a Finite-Volume discretisation of the steady-state groundwater equation on a 3×3 grid with a homogeneous permeability field and Dirichlet boundary condition on the north and south side and no-flux boundary conditions at the left and right:

dirichlet north

6	7	8
3	4	5
0	1	2

noflux (left) **noflux** (right)

dirichlet south
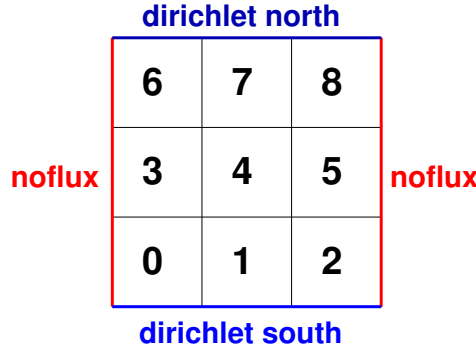
$$K(\vec{x}) = \begin{pmatrix} K & 0 \\ 0 & K \end{pmatrix}$$

The resulting linear equation system is:

$$\begin{pmatrix} 4K & -K & 0 & -K & 0 & 0 & 0 & 0 & 0 \\ -K & 5K & -K & 0 & -K & 0 & 0 & 0 & 0 \\ 0 & -K & 4K & 0 & 0 & -K & 0 & 0 & 0 \\ -K & 0 & 0 & 3K & -K & 0 & -K & 0 & 0 \\ 0 & -K & 0 & -K & 4K & -K & 0 & -K & 0 \\ 0 & 0 & -K & 0 & -K & 3K & 0 & 0 & -K \\ 0 & 0 & 0 & -K & 0 & 0 & 4K & -K & 0 \\ 0 & 0 & 0 & 0 & -K & 0 & -K & 5K & -K \\ 0 & 0 & 0 & 0 & 0 & -K & 0 & -K & 4K \end{pmatrix} \begin{pmatrix} p_0 \\ p_1 \\ p_2 \\ p_3 \\ p_4 \\ p_5 \\ p_6 \\ p_7 \\ p_8 \end{pmatrix} = \begin{pmatrix} 2Kp_{d_{\text{south}}} \\ 2Kp_{d_{\text{south}}} \\ 2Kp_{d_{\text{south}}} \\ 0 \\ 0 \\ 0 \\ 2Kp_{d_{\text{north}}} \\ 2Kp_{d_{\text{north}}} \\ 2Kp_{d_{\text{north}}} \end{pmatrix}$$

Effective Permeability

We assume that the permeability is a diagonal Tensor, which is depending on the position, but constant on each grid cell g_{ij}.

We need to evaluate K at the cell boundaries $x_{i\pm0.5,j\pm0.5}$.

What is the correct value of K if it is not homogeneous but element-wise constant?

To derive the correct effective permeability from an analysis of the one-dimensional problem. The steady-state groundwater flow equation is:

$$\frac{dJ_w}{dx} = 0 \quad \text{in } \Omega = (0, \underbrace{\ell}_{\text{length}})$$

$$J_w = -K(x)\frac{dp}{dx}$$
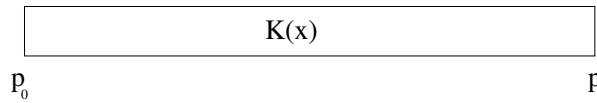
with the Dirichlet boundary conditions

$$p(0) = p_0$$
$$p(\ell) = p_\ell$$

because of $\frac{dJ_w}{dx} = 0$ in $\Omega \Leftrightarrow J_w(x) = J_0 \in \mathbb{R}$ this means

$$J_0 = -K(x)\frac{dp}{dx} \Leftrightarrow \frac{dp}{dx} = -\frac{J_0}{K(x)}$$

By integration of both sides over the domain

$$\frac{dp}{dx} = -\frac{J_0}{K(x)}$$

$$\Leftrightarrow \int_0^\ell \frac{dp}{dx}dx = [p(x)]_0^\ell = p_\ell - p_0 = -J_0 \int_0^\ell \frac{1}{K(x)}dx$$

we get the flux depending on the boundary conditions and the permeability distribution:

$$\Leftrightarrow J_0 = - \underbrace{\frac{\ell}{\int_0^\ell \frac{1}{K(x)}dx}}_{\text{eff.permeability}} \cdot \underbrace{\frac{p_\ell - p_0}{\ell}}_{\text{approx.gradient}}$$

If we divide the domain into two halves with constant permeability if $K(x) = \begin{cases} K_l & x \leq \frac{\ell}{2} \\ K_r & x > \frac{\ell}{2} \end{cases}$

we can perform the integration and get the effective permeability

$$K_{\text{eff}} = \frac{\ell}{\int\limits_0^\ell \frac{1}{K(x)}dx} = \frac{\ell}{\frac{\ell}{2}\frac{1}{K_l} + \frac{\ell}{2}\frac{1}{K_r}} = \frac{2}{\frac{1}{K_l} + \frac{1}{K_r}}$$

We therefore choose for cell-wise constant permeabilities the harmonic mean

$$K(\vec{x}_{i\pm0.5,j}) = \frac{2}{\frac{1}{K(\vec{x}_{i,j})} + \frac{1}{K(\vec{x}_{i\pm1,j})}}$$

Finite-Volume Method for tensor-product Grids
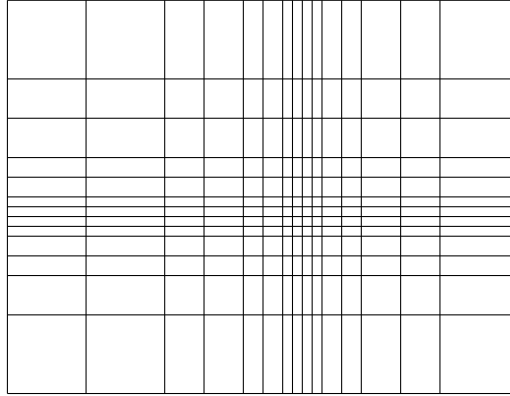
$$-\frac{2h_{y_j}K_{xx}(\vec{x}_{i-0.5,j})}{h_{x_{i-1}} + h_{x_i}} \cdot p_{i-1,j} - \frac{2h_{y_j}K_{xx}(\vec{x}_{i+0.5,j})}{h_{x_i} + h_{x_{i+1}}} \cdot p_{i+1,j}$$
$$-\frac{2h_{x_i}K_{yy}(\vec{x}_{i,j-0.5})}{h_{y_{j-1}} + h_{y_j}} \cdot p_{i,j-1} - \frac{2h_{x_i}K_{yy}(\vec{x}_{i,j+0.5})}{h_{y_j} + h_{y_{j+1}}} \cdot p_{i,j+1}$$
$$+ \left[\frac{2h_{y_j}K_{xx}(\vec{x}_{i-0.5,j})}{h_{x_{i-1}} + h_{x_i}} + \frac{2h_{y_j}K_{xx}(\vec{x}_{i+0.5,j})}{h_{x_i} + h_{x_{i+1}}} \right.$$
$$\left. + \frac{2h_{x_i}K_{yy}(\vec{x}_{i,j-0.5})}{h_{y_{j-1}} + h_{y_j}} + \frac{2h_{x_i}K_{yy}(\vec{x}_{i,j+0.5})}{h_{y_j} + h_{y_{j+1}}} \right] \cdot p_{i,j} = h_{x_i}h_{y_j}r(\vec{x}_{i,j})$$

Complexer Grids with Cell-Centred Finite Volumes

With the Cell-Centred Finite Volume Method it is also possible to use some kind of unstructured grids:

Nested Grids

Voronoi Grids

x_k

G

F

B

x_i

C_i

A

C

E

D

B

M• •J

•I

•K

A

γ

x
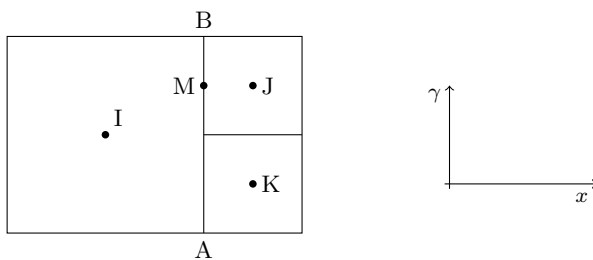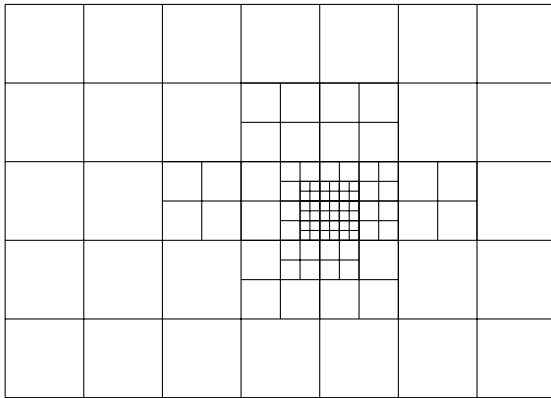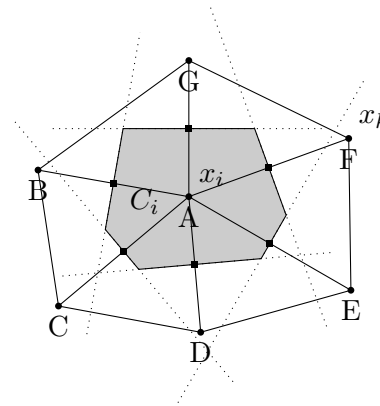
Summary Cell-Centred Finite-Volume Method

- Only the integral of the partial differential equation over each grid cell must fullfill the equation.

- Implementation of Dirichlet Boundary and Neumann Boundary conditions straight forward

- Structured and unstructured grids possible

- Dirichlet boundary conditions can easily be integrated by rearranging the equation systems and bringing them to the right side of the equation.

- Neumann boundary conditions can easily be integrated in the flux integrals

- Convergence order can differ dependent on the concrete method.

Properties of the Cell-Centred Finite-Volume Method

- Advantages:
  - well suited for structured grids
  - locally mass conservative
  - good approximation of average permeability
  - limited variety of unstructured grids possible
  - limited local adaptivity possible

– cheap for simple problems

- Problems:
  - Only linear convergence rate on non-equidistant grids
  - grid generation can be complicated (must fulfil rather strong conditions)

4.6 Vertex-Centred Finite-Volume Method
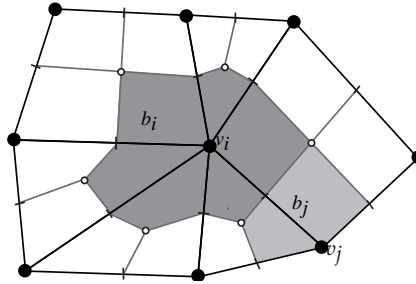
- The unknowns are located at the edges of the elements (vertices)

- Base functions are used on each element, which are parameterised with the values at the vertices

- A secondary mesh is constructed connecting the face centres and the barycenter of the element

- The flux balance is not calculated over the original grid, but over the secondary mesh, the elements of the secondary mesh are called control-volumes, the parts of a control volume belonging to a specific element of the primary mesh are called subcontrol-volumes.
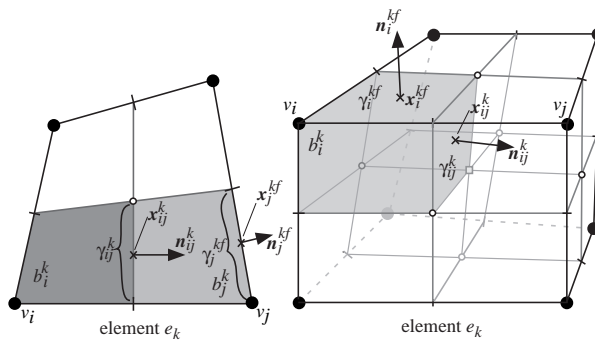
- Material properties are assumed to be constant for each element

- The volume integrals are calculated as a sum over the subcontrol-volumes using the midpoint rule and the material properties valid for the specific control-volume. $\sum_i b_i^k \cdot r_i^k$

- The face integrals are calculated as a sum over all subcontrol-volume faces with the midpoint rule $\sum_{ij} \gamma_{ij}^k \vec{J}_{ij}^k \vec{n}_{ij}^k$

- The gradient at the face centres is given by the base functions.

Properties of the Vertex-Centred Finite-Volume Method

- Advantages:
  - can be used for domains with complicated shape
  - well suited for unstructured grids
  - local adaptivity possible
  - locally mass conservative

- Problems:
  - grid generation can be complicated (must often fullfill certain conditions)
  - more computationally expensive for simple problems
  - bad approximation of average permeability

4.7 Influence of discretisations on estimated effective conductivity

Natural porous can be strongly heterogeneous at very different scales (Figure 10).
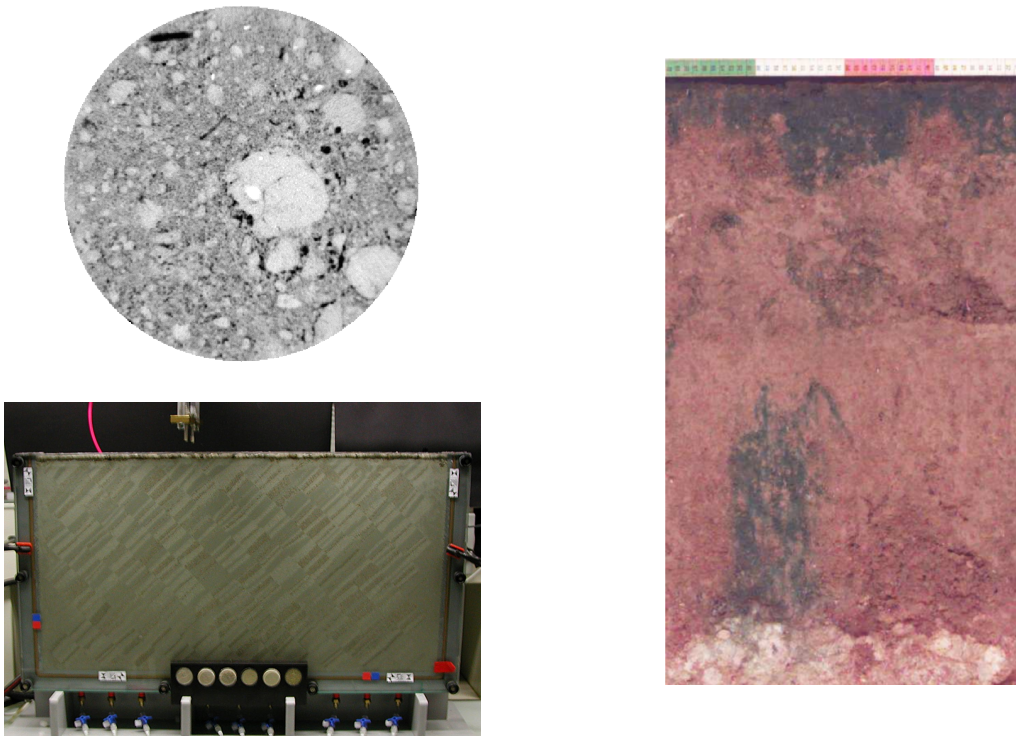
Figure 10: Strong heterogeneity in soils occurs on all scales: at the pore scale (upper left), the lab scale (lower left) and in the field (right).

- Numerical models are often used to determine the effective properties of heterogeneous porous media

- Numerical models perform an internal averaging of properties of conductivities between elements/grid cells

43

- This averaging influences the estimated parameters

- While all reasonable discretisation schemes converge to the correct solution if the grid size goes to zero, the convergence speed and the starting position can be quite different

The effective permeability is determined by applying a constant pressure at the top and bottom boundary and no-flux boundary conditions at the side boundaries. The cummulated flux over a horizontal line is divided by the macroscopic pressure gradient to get the effective conductivity.

The cell-centred Finite Volume scheme calculates the fluxes over faces and uses an harmonic mean of the conductivities. In the case of a checkerboard conductivity with one element per conductivity unit it therefore produces an harmonic average of the conductivities.
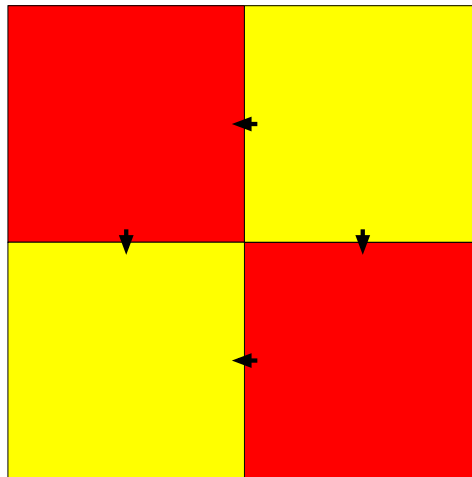
The standard Finite Element and the vertex-centred Finite Volume scheme integrate over the fluxes inside one element or at the sub-control volume faces. This directly influences the value at the vertex and this again the fluxes in the elements attached to the vertex. The heterogeneity is only taken implicitly into account and leads to an arithmetic averaging of the conductivities. In the case of a checkerboard conductivity with one element per conductivity unit these schemes therefore produce an arithmetic average of the conductivities.
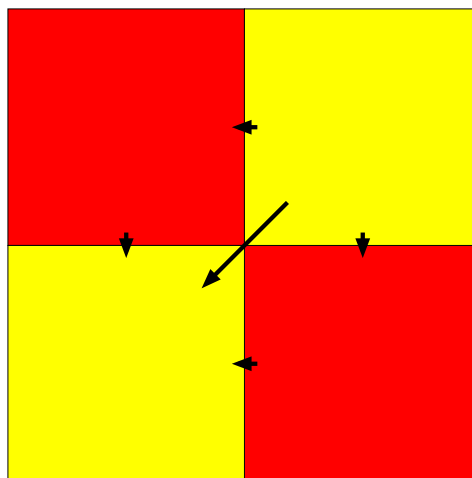
The cell-centred Finite Volume scheme tends to underestimate the effective permeability, the standard Finite Element and the vertex-centred Finite Volume scheme tend to overestimate the effective permeability.

To investigate the effects of more complex permeability distributions Durlofsky (1994) proposed a model problem:

$$
\begin{aligned}
\nabla \cdot \vec{J}_w &= 0 & &\text{in } \Omega = (0,1) \times (0,1) \\
\vec{J}_w &= -\mathbb{K}\nabla p & & \\
p &= 1 & &\text{on left boundary} \\
p &= 0 & &\text{on right boundary} \\
\vec{J}_w \cdot \vec{n} &= 0 & &\text{on upper and} \\
& & &\text{lower boundary}
\end{aligned}
$$

permeability $\mathbb{K} = 1$ permeability $\mathbb{K} = 10^{-6}$
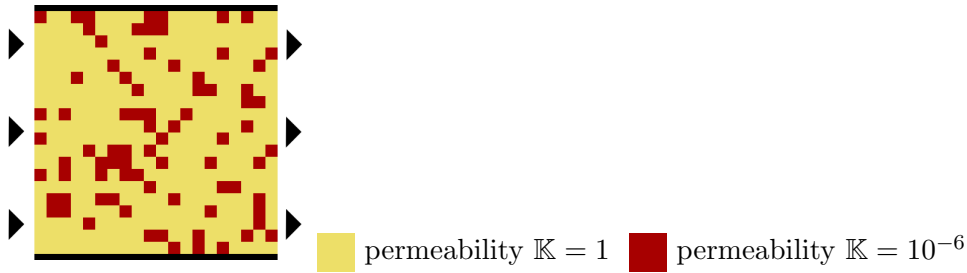
L. J. Durlofsky, *Accuracy of mixed and control volume finite element approximations to Darcy velocity and related quantities*, Water Resources Research **30** (1994), no. 4, 965-973.

The result (Figure 11) shows that all discretisations tend to the same value, but that the cell-centred Finite Volume scheme and the Mimetic Finite Difference scheme converge much faster. The convergence of the vertex-centered Finite Volume scheme slows down considerably so that one might accept a wrong value as the improvement between two grid refinements is "small enough".
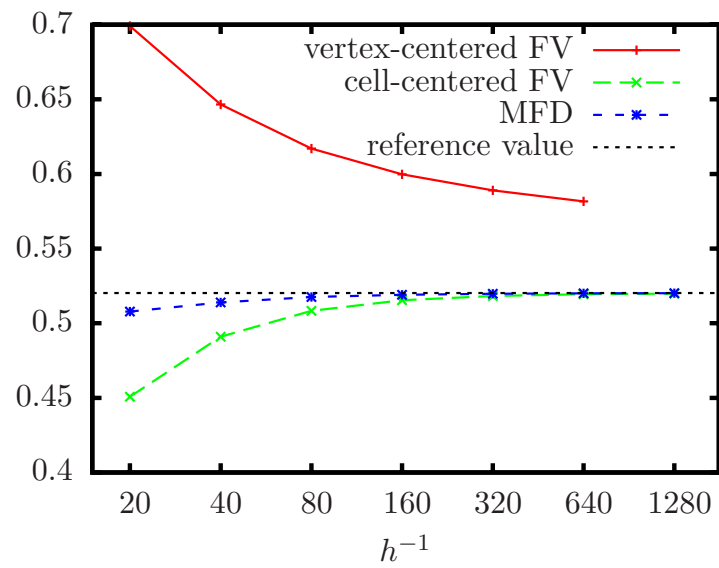
Figure 11: Effective conductivity for the Durlovsky problem calculated with different discretisation schemes and successive grid refinement.

5 Solution of Linear Equation Systems

5.1 Direct Solution of Sparse Linear Equation Systems

We do a Gaussian eliminitation for $A \cdot \vec{x} = \vec{b}$ with $A \in \mathbb{R}^{N \times N}$ regular, $\vec{x}, \vec{b} \in \mathbb{R}^N$ and A is a matrix assembled by the Finite-Volume method.
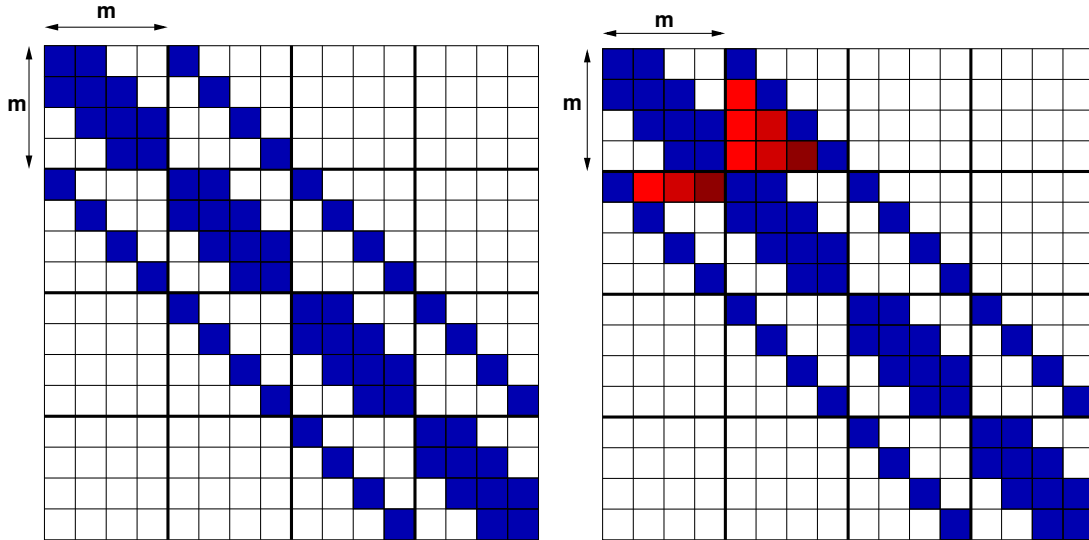
- As A is symmetric and positive definite the elimination can be done without pivoting

- New non-zero elements are created during the eliminitation ("fill in")

- The "fill in" is created within the outer diagonals

Complexity of the Elimination

Due to the "fill in" '$O(N) = O(n \cdot m)$ matrix entries become $O(n \cdot m \cdot m) = O(n \cdot m^2)$ matrix entries after the elimination.

The complexity of the elimination is:

$$\text{Complexity} \leq \sum_{i=1}^{N} \underbrace{m}_{\substack{\# \text{elements} \\ \text{to eliminate} \\ \text{until diagonal} \\ \text{in line } i}} \cdot \underbrace{m}_{\substack{\text{lower limit} \\ \text{for} \\ \text{elimination} \\ \text{of one} \\ \text{element}}} = N \cdot m^2 = n \cdot m^3$$

If $n = m$ the complexity of the elimination is $O(N^2)$, with optimal numbering of the nodes $O(N^{3/2})$, compared to $O(N^3)$ with a fully occupied matrix.

In three dimensions: The elimination has a complexity of $O(N^{7/3})$

In one dimension: The elimination has an optimal complexity of $O(N)$

5.2 Iterative Solution of Sparse Linear Equation Systems

As typical matrices from the discretisation of partial differential equations are sparse (i.e. they contain only $k \cdot n$ elements, where k is a small integer usually up to 7) we want to find more efficient solution methods than Gaussian elimination, which exploit the sparsity of the matrix. This is done by iterative solution methods.

Starting from an initial value $\vec{x}^{(0)} \in \mathbb{R}^N$, iterative solution methods create a sequence

$$\vec{x}^{(0)}, \vec{x}^{(1)}, \ldots, \vec{x}^{(k)}, \ldots$$

with the characteristic

$$\lim_{k \to \infty} \vec{x}^{(k)} = \vec{x}.$$

5.2.1 Relaxation Methods

The ith equation in $A\vec{x} = \vec{b}$ is:

$$\sum_{j=1}^{N} a_{ij} x_j = b_i$$

solve for x_i:

$$x_i = \frac{1}{a_{ii}} \left(b_i - \sum_{j \neq i} a_{ij} x_j \right)$$

Precondition: $a_{ii} \neq 0 \quad \forall i = 1 \ldots N$. This is not true for all matrices

Gauß-Seidel Iteration: Algorithm

Update all columns one after the other:

$$
\begin{aligned}
&\text{given } \vec{x}^{(k)} \\
&\text{for } (i = 1; i \leq N; i = i + 1) \\
&\quad x_i^{(k+1)} = \frac{1}{a_{ii}} \left(b_i - \sum_{j < i} a_{ij} x_j^{(k+1)} - \sum_{j > i} a_{ij} x_j^{(k)} \right) \\
&\text{yields } \vec{x}^{(k+1)}
\end{aligned}
$$

This scheme is called Gauß-Seidel Iteration.

Complexity for calculation of $\vec{x}^{(k+1)}$ from $\vec{x}^{(k)}$ proportional to number of non-zero elements of the matrix, therefore $O(N)$ for sparse matrices.

Open Questions

- Under which conditions is the sequence converging with $\lim_{k \to \infty} \vec{x}^{(k)} = \vec{x}$.

- How many iterations are necessary to reach $||\vec{x}^{(k)} - \vec{x}|| \leq \epsilon$ for a given precision ϵ?

- How can one determine efficiently if $||\vec{x}^{(k)} - \vec{x}|| \leq \epsilon$ is reached? (we don't know the exact solution $\vec{x}$)

Other Relaxation Methods

Jacobi Iteration:

$$x_i^{(k+1)} = \frac{1}{a_{ii}} \left(b_i - \sum_{j \neq i} a_{ij} x_j^{(k)} \right)$$

Damped Jacobi Iteration:

$$x_i^{(k+1)} = (1 - \omega) x_i^{(k)} + \frac{\omega}{a_{ii}} \left(b_i - \sum_{j \neq i} a_{ij} x_j^{(k)} \right)$$

special case: $\omega = 1 \Rightarrow$ Jacobi Iteration

SOR (successive overrelaxation) Iteration:

$$x_i^{(k+1)} = (1 - \omega) x_i^{(k)} + \frac{\omega}{a_{ii}} \left(b_i - \sum_{j < i} a_{ij} x_j^{(k+1)} - \sum_{j > i} a_{ij} x_j^{(k)} \right)$$

$0 < \omega < 1$: underrelaxation $1 < \omega < 2$: overrelaxation   special case: $\omega = 1 \Rightarrow$ Gauß-Seidel Iteration

Damped Richardson Iteration:

$$x_i^{(k+1)} = (1 - a_{ii}\omega) x_i^{(k)} + \omega \left(b_i - \sum_{j \neq i} a_{ij} x_j^{(k)} \right)$$

Matrix Notation of Relaxation Methods

For an analysis of the convergence behavior it is more convenient to write the iteration schemes as matrix operations:

As $\vec{x} = \vec{x}^{(k)} + \vec{e}^{(k)}$ and $A\vec{e}^{(k)} = \vec{b} - A\vec{x}^{(k)}$ we could calculate $\vec{x}$ from

$$\vec{x} = \vec{x}^{(k)} + A^{-1} \left(\vec{b} - A\vec{x}^{(k)} \right)$$

However inverting A is at least as expensive as calculating the solution of $A\vec{x} = \vec{b}$ with a direct method. We therefore approximate the matrix A^{-1} with a matrix M^{-1}, where M is an approximation of A, which is easy to invert, and get the new formula

$$\vec{x}^{(k+1)} = \vec{x}^{(k)} + M^{-1} \left(\vec{b} - A\vec{x}^{(k)} \right)$$

$\vec{x}^{(k+1)}$ is no longer the exact solution but (hopefully) an improvement to $\vec{x}^{(k)}$

We split $A = L + D + U$ into a strictly lower diagonal matrix L, a strictly upper diagonal matrix U and a diagonal Matrix D.

Now we can get the iteration methods described above by

$M = \omega^{-1}I$	damped Richardson iteration
$M = D$	Jacobi iteration
$M = \omega^{-1}D$	damped Jacobi iteration
$M = L + D$	Gauß-Seidel iteration
$M = L + \omega^{-1}D$	SOR iteration

Now we want to analyse the change of the error $\vec{e}^{(k)}$ in one iteration.
For the general iteration scheme we get:

$$
\begin{aligned}
\vec{x}^{(k+1)} &= \vec{x}^{(k)} + M^{-1}\left(\vec{b} - A\vec{x}^{(k)}\right) \\
\Leftrightarrow \underbrace{\vec{x} - \vec{x}^{(k+1)}}_{\vec{e}^{(k+1)}} &= \underbrace{\vec{x} - \vec{x}^{(k)}}_{\vec{e}^{(k)}} - M^{-1}\left(\vec{b} - A\vec{x}^{(k)}\right) \\
\vec{e}^{(k+1)} &= \vec{e}^{(k)} - M^{-1}\left(A\vec{x} - A\vec{x}^{(k)}\right) \\
&= \vec{e}^{(k)} - M^{-1}A\left(\vec{x} - \vec{x}^{(k)}\right) \\
&= \underbrace{\left(I - M^{-1}A\right)}_{=:S}\vec{e}^{(k)}
\end{aligned}
$$

We call $S = I - M^{-1}A$ the iteration matrix.
The error propagation is therefore:

$$
\vec{e}^{(k+1)} = S \cdot \vec{e}^{(k)}
$$

with the iteration matrix $S = I - M^{-1}A$.

Recursive insertion yields:

$$
\vec{e}^{(k)} = S \cdot \vec{e}^{(k-1)} = S^2 \cdot \vec{e}^{(k-2)} = \cdots = S^k \cdot \vec{e}^{(0)}
$$

If $\lim\limits_{k\to\infty} S^k = 0$ (zero matrix) the scheme converges independently of $\vec{e}^{(0)}$.

This is guaranteed if $\rho(S) < 1$, where $\rho(S) = \max\{|\lambda| \mid \lambda \text{ is eigenvalue of } S\}$ is called spectral radius of S.

Eigenvalues and Eigenvectors

- If A is symmetric and positive definite (and often if it is not) $\Rightarrow$ there exists a set of N linearly independent eigenvectors $\vec{z}_1, \vec{z}_2, \ldots, \vec{z}_N$.

- If $\vec{z}_i$ is eigenvector of A, $\alpha\vec{z}_i$ with $\alpha \in \mathbb{R}$ is also eigenvector of A.

- The product of A and z_i is equal to z_i times the scalar eigenvalue λ_i:

$$
A\vec{z}_i = \lambda_i\vec{z}_i
$$

- As the N eigenvectors are linearly independent, they form a basis of $\mathbb{R}^N$, i.e. every vector $\vec{x}$ can be expressed as a linear combination of the eigenvectors.

$$
\vec{x} = \sum_{i=1}^{N} \xi_i \vec{z}_i
$$

- As matrix-vector multiplication is distributive:

$$
A\vec{x} = \sum_{i=1}^{N} \xi_i A\vec{z}_i = \sum_{i=1}^{N} \xi_i \lambda_i \vec{z}_i
$$

Matrix multiplication with eigenvector if eigenvalue < 1
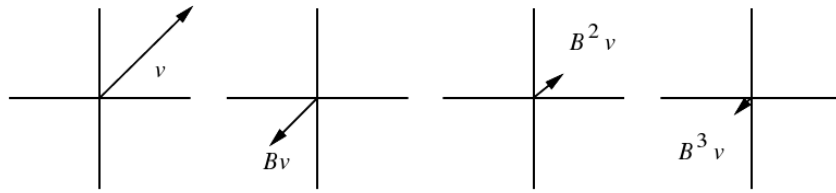
Matrix multiplication with eigenvector if eigenvalue > 1
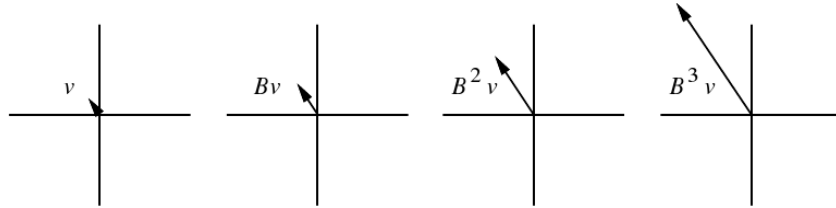
Matrix multiplication with vector which is sum of two eigenvectors
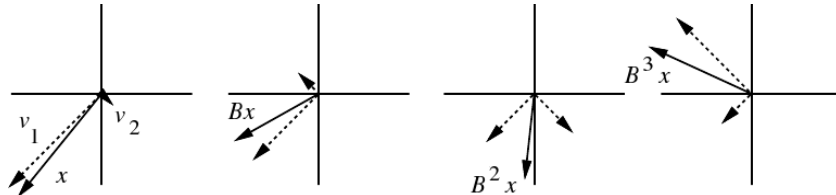
figures from J. R. Shewchuk (1994): "An Introduction to the Conjugate Gradient Method Without the Agonizing Pain"

Convergence Analysis for the Damped Richardson Iteration

If we assume that A is symmetric and positive definite $\Rightarrow$ all eigenvalues of A are real and positive. A convergence analysis can then easily be made for the damped Richardson iteration $(M = \omega^{-1} I)$ with the iteration formula:

$$\vec{x}^{(k+1)} = \vec{x}^{(k)} + \omega \left(\vec{b} - A\vec{x}^{(k)} \right)$$

The iteration matrix S is $S = I - M^{-1}A = I - \omega A$ and has the eigenvalues $\mu_i = 1 - \omega \lambda_i$ where λ_i is an eigenvalue of A.

If we use $\omega = \frac{1}{\lambda_{\mathsf{max}}(A)}$, we get:

$$0 = 1 - \frac{\lambda_{\mathsf{max}}(A)}{\lambda_{\mathsf{max}}(A)} \leq \mu_i \leq 1 - \frac{\lambda_{\mathsf{min}}(A)}{\lambda_{\mathsf{max}}(A)} = 1 - \frac{1}{\kappa(A)}$$

We call $\kappa(A) = \frac{\lambda_{\mathsf{max}}(A)}{\lambda_{\mathsf{min}}(A)}$ the spectral condition of A.

The spectral Radius of S is:

$$\rho(S) \leq 1 - \frac{1}{\kappa(A)}, \text{ with } \kappa(A) \geq 1$$

Convergence for Matrices from PDE-Discretizations

For the solution of the Laplace equation

$$\Delta p = \nabla \cdot (\nabla p) = 0$$

in $\Omega \subset \mathbb{R}^d$ with the Finite-Difference discretisation (i.e. $A \in \mathbb{R}^{N \times N}$) we get $\kappa(A) = O(N^{2/d})$.

$\Rightarrow$ the error reduction is decreasing with increasing matrix size.

Similar results can be obtained for other relaxation methods like Jacobi or Gauss-Seidel iteration.

Further Convergence Results

- If A and $2 \cdot D - A$ are both positive definite the Jacobi iteration converges.

- If A is strictly diagonally dominant ($a_{ii} > \sum_{j \neq i} |a_{ij}| \forall i$) the Jacobi and Gauß-Seidel iterations converge.

- SOR can only converge if $0 < \omega < 2$.

- If A is positive definite, both SOR and Gauß-Seidel converge.

- For many problems occurring in practical applications no convergence proofs exist.

Terminating Condition

We call $\vec{e}^{(k)} := \vec{x} - \vec{x}^{(k)}$ the error of the kth iterate. As we do not know the exact solution $\vec{x}$ the error is hard to determine.

With

$$A\vec{e}^{(k)} = A\left(\vec{x} - \vec{x}^{(k)}\right) = A\vec{x} - A\vec{x}^{(k)} = \vec{b} - A\vec{x}^{(k)} =: \vec{d}^{(k)}$$

we derive the defect vector $\vec{d}^{(k)} := \vec{b} - A\vec{x}^{(k)}$, which can be computed easily.

Because of $A\vec{e}^{(k)} = \vec{d}^{(k)} \Leftrightarrow \vec{e}^{(k)} = A^{-1}\vec{d}^{(k)}$ and therefore $||\vec{e}^{(k)}|| \leq ||A^{-1}|| \cdot ||\vec{d}^{(k)}||$ we can use the norm of the defect $||\vec{d}^{(k)}||$ as terminating condition.

As $||A^{-1}||$ can be very large, we use a relative termination criterium: $||\vec{d}^{(k)}|| < \varepsilon ||\vec{d}^{(0)}||$ with a suitable ε.

The new defect is better not calculated from $\vec{d}^{(k+1)} = \vec{b} - A\vec{x}^{(k+1)}$ as with this formulation cancelation errors are increasing if the defect gets smaller.

The defect in step $k + 1$ is:

$$
\begin{aligned}
\vec{d}^{(k+1)} &= \vec{b} - A\vec{x}^{(k+1)} = \vec{b} - A\left(\vec{x}^{(k)} + \vec{v}^{(k)}\right) \\
&= \vec{b} - A\vec{x}^{(k)} - A\vec{v}^{(k)} = \vec{d}^{(k)} - A\vec{v}^{(k)}
\end{aligned}
$$

$\vec{d}^{(k+1)} = \vec{d}^{(k)} - A\vec{v}^{(k)}$ is therefore an equivalent reformulation which reduces the cancelation errors.

The iteration scheme can also be reformulated in terms of the defect and the correction:

$$x_i^{(k+1)} = (1 - \omega)\, x_i^{(k)} + \frac{\omega}{a_{ii}} \left(b_i - \sum_{j<i} a_{ij} x_j^{(k+1)} - \sum_{j>i} a_{ij} x_j^{(k)} \right)$$

$$x_i^{(k+1)} - x_i^{(k)} = \frac{\omega}{a_{ii}} \left(b_i - \sum_{j<i} a_{ij} x_j^{(k+1)} - \sum_{j\geq i} a_{ij} x_j^{(k)} \right)$$

$$v_i^{(k)} = \frac{\omega}{a_{ii}} \left(b_i - \sum_{j<i} a_{ij} \left(x_j^{(k)} + v_j^{(k)} \right) - \sum_{j\geq i} a_{ij} x_j^{(k)} \right)$$

$$v_i^{(k)} = \frac{\omega}{a_{ii}} \left(b_i - \sum_{j} a_{ij} x_j^{(k)} - \sum_{j<i} a_{ij} v_j^{(k)} \right)$$

$$v_i^{(k)} = \frac{\omega}{a_{ii}} \left(d_i^{(k)} - \sum_{j<i} a_{ij} v_j^{(k)} \right)$$

Damped Richardson Iteration:
$$v_i^{(k)} = \omega d_i^{(k)}$$

Damped Jacobi Iteration:
$$v_i^{(k)} = \frac{\omega}{a_{ii}} d_i^{(k)}$$

special case: $\omega = 1 \Rightarrow$ Jacobi Iteration

SOR (successive overrelaxation) Iteration:

$$v_i^{(k)} = \frac{\omega}{a_{ii}} \left(d_i^{(k)} - \sum_{j<i} a_{ij} v_j^{(k)} \right)$$

$0 < \omega < 1$: underrelaxation $1 < \omega < 2$: overrelaxation special case: $\omega = 1 \Rightarrow$ Gauß-Seidel Iteration

The usage of the defect formulation allows it in theory to reduce the defect to an arbitry fraction of the initial defect. However, in practice there is no further change of the solution if the correction is too small compared to the current solution.

Example Algorithm

The initial guess $\vec{x}$, the matrix A and the right side $\vec{b}$ are given.

$$\vec{d} = \vec{b} - A\vec{x};$$
$$d_0 = ||\vec{d}||;$$
$$d_k = d_0;$$
while $(d_k \geq \varepsilon \cdot d_0)$
{
    Solve $M \cdot \vec{v} = \vec{d}$
$$\vec{x} = \vec{x} + \vec{v};$$
$$\vec{d} = \vec{d} - A\vec{v};$$
$$d_k = ||\vec{d}||;$$
}

5.2.2 Data Structures for Sparse Matrices

To save memory A should not be stored as ordinary two-dimensional array.

One of the alternatives is called "compressed row storage" (CRS).

If $A \in \mathbb{R}^{N \times N}$ and s with $N < s < N^2$ is the total number of non-zero elements of A.

- All non-zero elements are stored line by line in a one-dimensional floating-point array `a` of size s.

- The corresponding column indices are stored line by line in a one-dimensional integer array `j` of size s.

- The start indices of each line are stored in an one-dimensional integer array `r` of size $N + 1$, where the total number of non-zero elements s is stored as last element of `r` (`r[N]=s`).

Example Matrix

$$
A = \begin{pmatrix}
2.1 & 0 & 3.4 & 0 & 0 \\
0 & 1.3 & 0 & 2 & 6.4 \\
1.1 & 0 & 5.3 & 0 & 0 \\
0 & 7.8 & 0 & 3.9 & 2.3 \\
5.8 & 0 & 0 & 3.1 & 6
\end{pmatrix}
$$

$$
\begin{aligned}
a &= \{2.1, 3.4, 1.3, 2, 6.4, 1.1, 5.3, 7.8, 3.9, 2.3, 5.8, 3.1, 6\} \\
j &= \{0, 2, 1, 3, 4, 0, 2, 1, 3, 4, 0, 3, 4\} \\
r &= \{0, 2, 5, 7, 10, 13\}
\end{aligned}
$$

Memory consumption: if `double` arrays are used for the floating point variables and `int` for the integer arrays: 200 bytes for storing the full matrix, 180 bytes for the CRS matrix (The gain is much larger if the size of the matrix increases).

Access an Element in a CRS-Matrix

```
double &GetA(int row, int column)
{
    for(k=r[row];k<r[row+1];++k)
    {
        if (j[k]==column)
            return(a[k]);
    }
    return(0.);
}
```

Computing $y = A \times x$ for a CRS-Matrix

```
1 for (i=0;i<N;++i)
  {
3     y[i]=0.;
      for(k=r[i];k<r[i+1];++k)
5         y[i] = y[i] + a[k] * x[j[k]];
  }
```

Improved CRS

- Assume that diagonal element does always exist

- Store diagonal element at position `r[row]`

- Do not store diagonal index

- Store number of elements in the row at `j[r[row]]`

Advantages:

- The position of the diagonal element is always clear (necessary for relaxation methods)

- The structure of the matrix (sparsity pattern) can vary a bit

5.2.3 Multigrid Methods

Smoothing Property of Linear Iterative Methods

We assume again that A is symmetric and positive definite. If $\vec{z}_k$ is an eigenvector of A:

$$A\vec{z}_k = \lambda_k \vec{z}_k$$

with $0 < \lambda_{\mathsf{min}} \leq \lambda_k \leq \lambda_{\mathsf{max}}$.

For Richardson's iteration with $\omega = 1/\lambda_{\mathsf{max}}$ and $\vec{e}^{(i)} = \vec{z}_k$ we obtain

$$\vec{e}^{(i+1)} = \left(I - \frac{1}{\lambda_{\mathsf{max}}} A \right) \vec{z}_k = \left(1 - \frac{\lambda_k}{\lambda_{\mathsf{max}}} \right) \vec{e}^{(i)}.$$

This means: $\begin{array}{ll} \lambda_k \text{ close to } \lambda_{\mathsf{max}} & \Rightarrow \left(1 - \frac{\lambda_k}{\lambda_{\mathsf{max}}} \right) \approx 0 \\ \lambda_k \text{ close to } \lambda_{\mathsf{min}} & \Rightarrow \left(1 - \frac{\lambda_k}{\lambda_{\mathsf{max}}} \right) \approx 1 \end{array}$

- Error components corresponding to *large* eigenvalues are damped efficiently.

- Error components corresponding to *small* eigenvalues are damped slowly.

For second order problems we have $\lambda_{\mathsf{min}}/\lambda_{\mathsf{max}} = O(h^2)$, i.e. the asymptotic convergence factor is

$$\rho = 1 - O(h^2).$$

The (damped) Jacobi and Gauß–Seidel iteration have an asymptotically similar behavior in contrast to an optimally damped SOR. However, the optimal damping coefficient for SOR is hard to determine.

Error Smoothing Example

We discretize $-\Delta p = r$ with the cell-centered Finite-Volume method on a structured mesh.

The initial error consists of low and high frequency parts.

The graphs show the initial error and the error after 1 and 5 iterations.
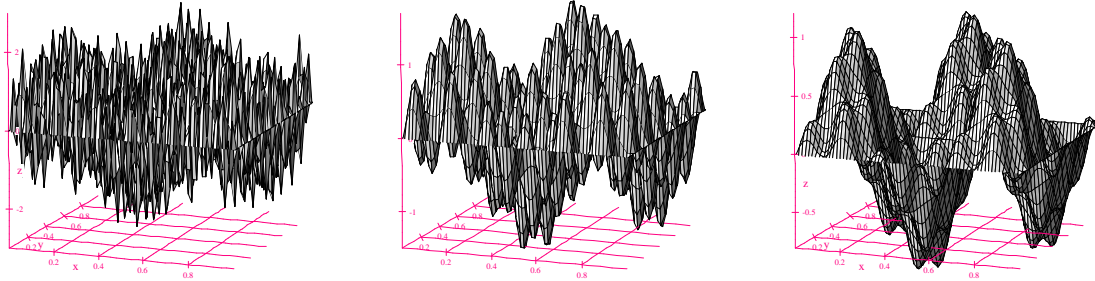
figures P. Bastian (personal communication)

Multigrid Idea

Construct an iteration that is complementary to the smoother reducing *low frequency errors.*

Idea: Low frequency errors can be represented on a coarser grid:
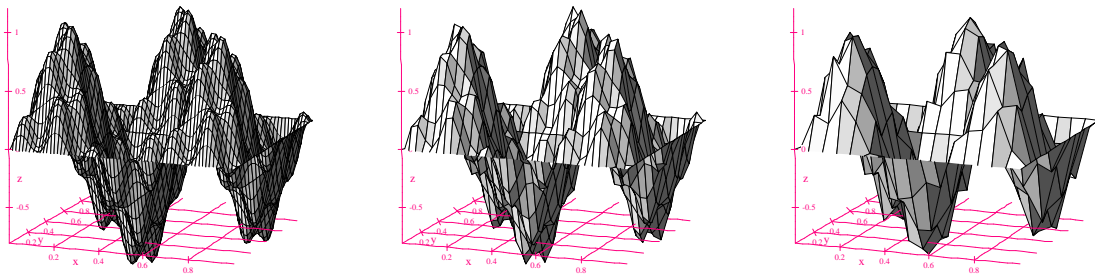
This requires a *hierarchy* of grids $\Omega_0, \Omega_1, \Omega_2, \ldots$

Correspondingly there will be a hierarchy of linear systems

$$A_l \vec{x}_l = \vec{b}_l$$

1D case

56

2D case

<table>
<tr><td>level 0</td><td>level 1</td><td>level 2</td></tr>
</table>

<table>
<tr><td>level 0</td><td>level 1</td><td>level 2</td></tr>
</table>

figures P. Bastian (personal communication)
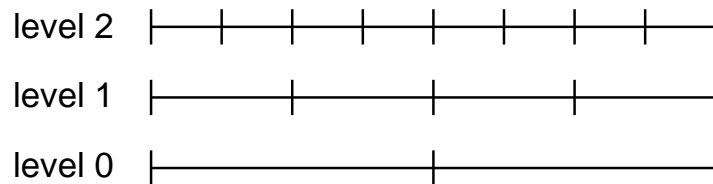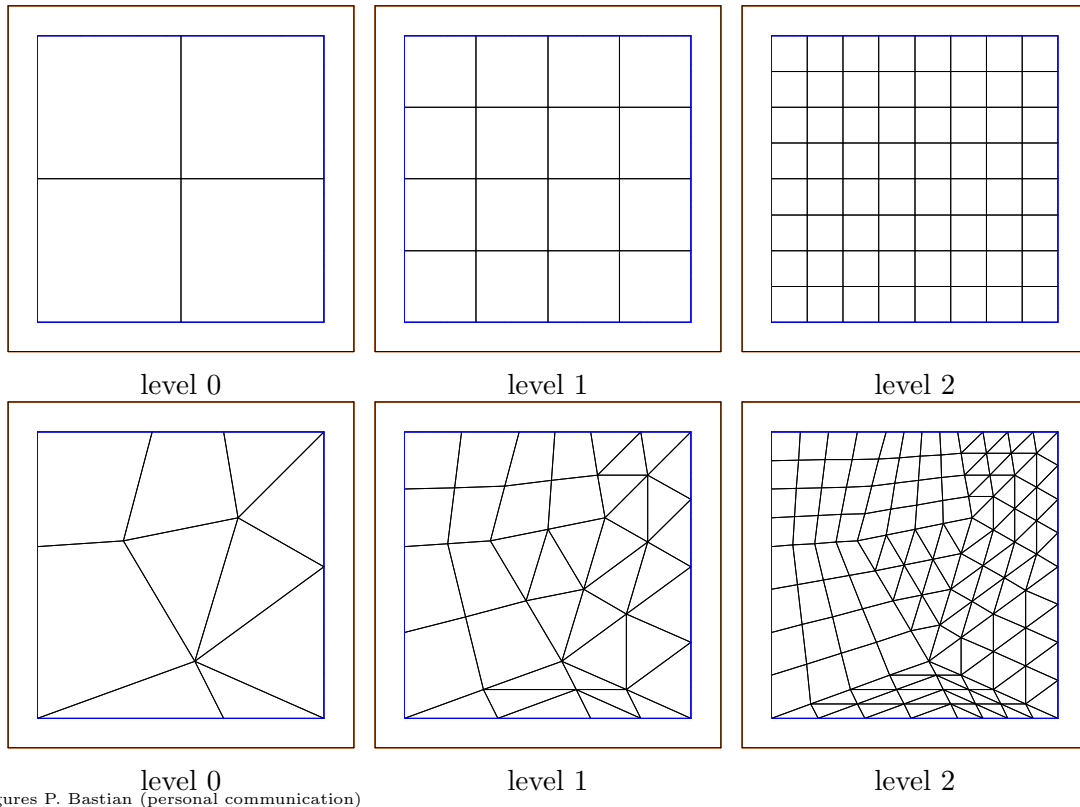
Multigrid Algorithm

- (pre)smoothing of the fine grid solution $\vec{x}_l^{(k)}$ (usually with some steps of a damped Jacobi or Gauß-Seidel iteration)

- compute defect $\vec{d}_l^{(k)}$

- restrict defect $\vec{d}_l^{(k)}$ to coarse grid $\vec{d}_{l-1}^{(k)}$ (either by just using the values at the grid points of the coarse grid or by averaging of fine grid values)

- compute solution $\vec{v}_{l-1}^{(k)}$ of $A_{l-1}\vec{v}_{l-1}^{(k)} = \vec{d}_{l-1}^{(k)}$ (with direct solution, relaxation methods or another coarse grid correction $\Rightarrow$ multigrid method)

- prolongate $\vec{v}_{l-1}^{(k)}$ to the fine grid Ω_l (interpolate $\vec{v}_l^{(k)}$ at the fine grid points)

- update fine grid solution $\vec{x}_l^{(k+1)} = \vec{x}_l^{(k)} + \vec{v}_l^{(k)}$

- sometimes (post)smoothing of the fine grid solution $\vec{x}_l^{(k+1)}$ (usually with some steps of a damped Jacobi or Gauß-Seidel iteration)

Multigrid methods

- have a overall work, which is still dominated by the finest grid. If C operations are necessary on the fine grid only $C/4$ operations in 2D and $C/8$ operations in 3D are necessary on the next coarser grid ...

57

- have a optimal complexity of $O(N)$ to solve $Ax = b$ for appropriate matrices (compared to $O(N^{3/2})$ to $O(N^2)$ with Gaussian elimination for banded matrices with bandwidth optimisation)

- there are also "Algebraic Multigrid" (AMG) solvers, which do not really construct a coarse grid, but use empirical schemes to generate coarser matrices from the fine-scale matrix. They have a complexity of $O(N \cdot ln(N))$.

5.2.4 Gradient based iterative methods

If A is symmetric and positive definite then $\vec{x}^T A \vec{x} > 0 \ \forall \vec{x} \neq 0$. Then $A\vec{x} = \vec{b}$ is equivalent to finding the minimum of the quadratic form

$$f(x) := \frac{1}{2}\vec{x}^T A \vec{x} - \vec{b}^T \vec{x} + c$$

where $c \in \mathbb{R}$ is an arbitrary scalar. As A is positive definite, the hypersurface defined by $f(\vec{x})$ forms a paraboloid in $\mathbb{R}^{N+1}$. The minimum $\vec{x}$ is unique and global.

Different gradient based methods depend on the strategy to find this minimum.
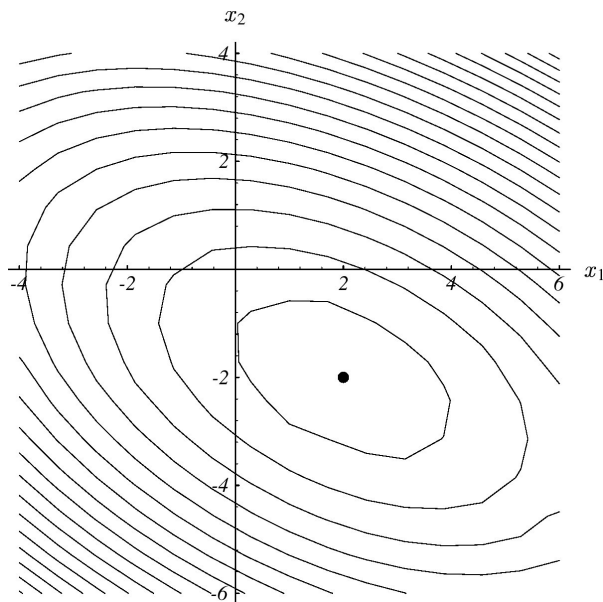
from J. R. Shewchuk (1994): "An Introduction to the Conjugate Gradient Method Without the Agonizing Pain"

Proof of Correspondence

The gradient of $f(\vec{x})$ is

$$f'(\vec{x}) := \frac{1}{2}A^T \vec{x} + \frac{1}{2}A\vec{x} - \vec{b}$$

for symmetric matrices this reduces to

$$f'(\vec{x}) := A\vec{x} - \vec{b}$$

At the minimum the gradient vanishes

$$f'(\vec{x}) := A\vec{x} - \vec{b} = 0$$

Therefore $\vec{x}$ at the minimum solves $A\vec{x} - \vec{b}$

Shape of the quadratic form $f(\vec{x})$

(a)   (b)

(c)   (d)
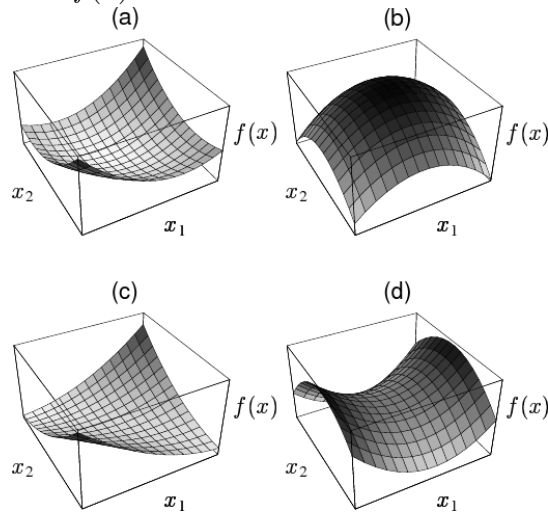
Quadratic form $f(\vec{x})$ for

- (a) a positive-definite matrix

- (b) a negative-definite matrix

- (c) a singular (and positive-definite) matrix

- (d) an indefinite matrix

from J. R. Shewchuk (1994): "An Introduction to the Conjugate Gradient Method Without the Agonizing Pain"

Method of Steepest Descent

Steepest Descend uses the direction of the negative gradient $-f'(\vec{x}^{(k)})$.

The improved solution is calculated from $\vec{x}^{(k+1)} = \vec{x}^{(k)} - \alpha f'(\vec{x}^{(k)})$.

The optimal step width α is chosen such that the minimum along the search direction is obtained. This results in the next descend being orthogonal to the search direction.
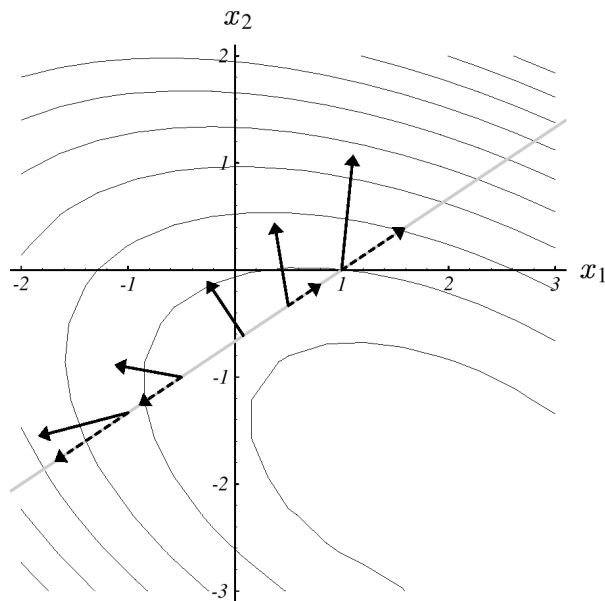
Optimal step size α

$$f'(\vec{x}^{(k)}) = A\vec{x}^{(k)} - \vec{b} = -\left(\vec{b} - A\vec{x}^{(k)}\right) = -\vec{d}^{(k)}$$

$$\vec{x}^{(k+1)} = \vec{x}^{(k)} + \alpha\vec{d}^{(k)}$$

We want to find a minimum along the search direction $\vec{v}^{(k)} = \vec{d}^{(k)}$

$$\frac{d}{d\alpha}f(\vec{x}^{(k+1)}) = 0$$

$$f'(\vec{x}^{(k+1)})^T \frac{d}{d\alpha}\vec{x}^{(k+1)} = f'(\vec{x}^{(k+1)})^T \vec{v}^{(k)} = 0$$

with $f'(\vec{x}^{(k+1)}) = -\vec{d}^{(k+1)}$:

$$\vec{d}^{(k+1)^T} \vec{v}^{(k)} = 0$$

$$
\begin{aligned}
\vec{d}^{(k+1)^T} \vec{v}^{(k)} &= 0 \\
\left(\vec{b} - A\vec{x}^{(k+1)}\right)^T \vec{v}^{(k)} &= 0 \\
\left(\vec{b} - A\left(\vec{x}^{(k)} + \alpha\vec{v}^{(k)}\right)\right)^T \vec{v}^{(k)} &= 0 \\
\left(\vec{b} - A\vec{x}^{(k)}\right)^T \vec{v}^{(k)} - \alpha\left(A\vec{v}^{(k)}\right)^T \vec{v}^{(k)} &= 0 \\
\alpha\left(A\vec{v}^{(k)}\right)^T \vec{v}^{(k)} &= \left(\vec{b} - A\vec{x}^{(k)}\right)^T \vec{v}^{(k)} \\
\alpha\vec{v}^{(k)^T} A^T \vec{v}^{(k)} &= \vec{d}^{(k)^T} \vec{v}^{(k)} \\
\alpha &= \frac{\vec{d}^{(k)^T} \vec{v}^{(k)}}{\vec{v}^{(k)^T} A\vec{v}^{(k)}} \\
\alpha_{\text{steep desc}} &= \frac{\vec{d}^{(k)^T} \vec{d}^{(k)}}{\vec{d}^{(k)^T} A\vec{d}^{(k)}}
\end{aligned}
$$

Steepest Descent Algorithm

$$\vec{d} = \vec{b} - A\vec{x}$$
$$d_0 = \vec{d}^T \vec{d}$$
$$d_k = d_0;$$
while $(d_k \geq \varepsilon^2 \cdot d_0)$
{
$$\alpha = \left(\vec{d}^T \vec{d}\right) / \left(\vec{d}^T A\vec{d}\right)$$
$$\vec{x} = \vec{x} + \alpha\vec{d}$$
$$\vec{d} = \vec{d} - \alpha A\vec{d}$$
$$d_k = \vec{d}^T \vec{d}$$
}

Optimised Steepest Descent Algorithm

$$\vec{d} = \vec{b} - A\vec{x}$$
$$d_0 = \vec{d}^T \vec{d}$$
$$d_k = d_0;$$
while $(d_k \geq \varepsilon^2 \cdot d_0)$
{
$$\vec{t} = A\vec{d}$$
$$\alpha = d_k / \left(\vec{d}^T \vec{t}\right)$$
$$\vec{x} = \vec{x} + \alpha\vec{d}$$
$$\vec{d} = \vec{d} - \alpha\vec{t}$$
$$d_k = \vec{d}^T \vec{d}$$
}

Convergence of Steepest Descent

Convergence of steepest descend depends strongly on the matrix condition $\kappa(A)$ and on the initial value. Convergence is reduced by the fact that achievements of previous steps can be lost again in later steps.
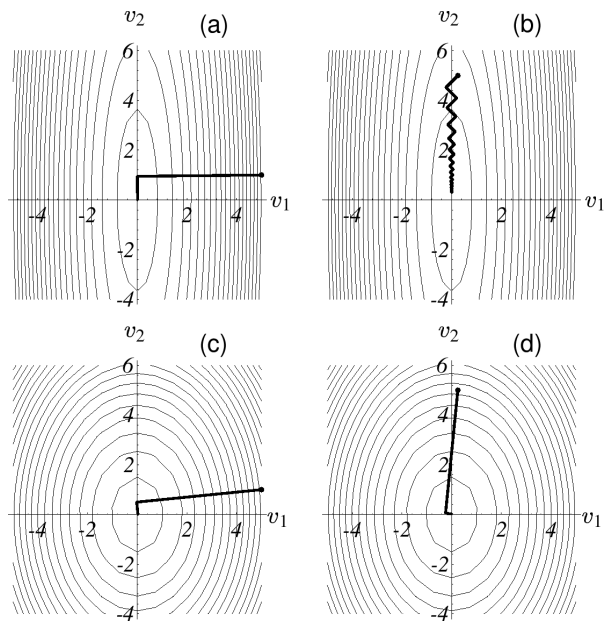
from J. R. Shewchuk (1994): "An Introduction to the Conjugate Gradient Method Without the Agonizing Pain"
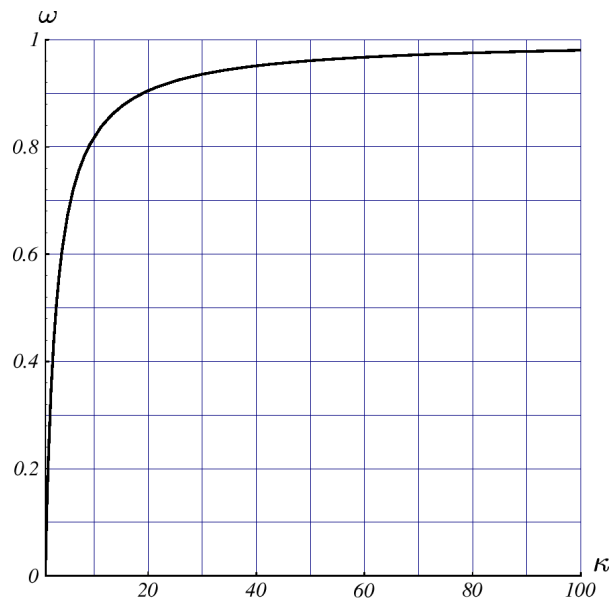
Figure 12: Convergence rate ω of the steepest descent method for different spectral conditions κ of the matrix A (from J. R. Shewchuk (1994): "An Introduction to the Conjugate Gradient Method Without the Agonizing Pain").

Convergence Rate

$$||\vec{e}^{(k)}||_A \leq \left(\frac{\kappa - 1}{\kappa + 1}\right)^k ||\vec{e}^{(0)}||_A$$

with the "energy norm"

$$||\vec{e}||_A = \sqrt{\vec{e}^T A \vec{e}}$$

Improvement of Gradient Based Methods

- Avoid loss of achievements

- Take orthogonal search directions $\vec{v}^{(i)^T} \vec{v}^{(j)} = 0 \ \forall i \neq j$

- Take one step in each search direction, which eliminates error in this direction

- Problem: The step width is obtained by

$$\begin{aligned}
\vec{v}^{(k)^T} \vec{e}^{(k+1)} &= 0 \\
\vec{v}^{(k)^T} \left(\vec{e}^{(k)} + \alpha_k \vec{v}^{(k)} \right) &= 0 \\
\alpha_k &= \frac{\vec{v}^{(k)^T} \vec{e}^{(k)}}{\vec{v}^{(k)^T} \vec{v}^{(k)}}
\end{aligned}$$

  we do not know $\vec{e}^{(k)}$

- Idea: Make search directions A-orthogonal $\vec{v}^{(i)^T} A \vec{v}^{(j)} = 0 \ \forall i \neq j$ as we know $A\vec{e}^{(k)} = \vec{d}^{(k)}$

Creation of A-orthogonal Search Directions

- A-Orthogonal vectors can be created from a set of n linearly independent vectors $\vec{u}_0, \vec{u}_1, \ldots, \vec{u}_{n-1}$ by Gram-Schmidt conjugation:
  - Take $\vec{v}_0 = \vec{u}_0$
  - In a recursive procedure take vector $\vec{u}_i$ and remove all components that are not A-orthogonal to the already previously created vectors $\vec{v}_j$.

$$\begin{aligned}
\vec{v}^{(i)^T} A \vec{v}^{(j)^T} &= \left(\vec{u}^{(i)^T} + \sum_{k=0}^{i-1} \beta_{ik} \vec{v}^{(k)^T} \right) A \vec{v}^{(j)} && (7) \\
&= \vec{u}^{(i)^T} A \vec{v}^{(j)} + \sum_{k=0}^{i-1} \beta_{ik} \vec{v}^{(k)^T} A \vec{v}^{(j)} && (8) \\
0 &= \vec{u}^{(i)^T} A \vec{v}^{(j)} + \beta_{ij} \vec{v}^{(j)^T} A \vec{v}^{(j)} && (9) \\
\beta_{ij} &= -\frac{\vec{u}^{(i)^T} A \vec{v}^{(j)}}{\vec{v}^{(j)^T} A \vec{v}^{(j)}} && (10)
\end{aligned}$$

- Yields in exact arithmetics the correct solution in n steps

- Problems:
  - We have to store all previous search directions
  - The memory consumption and the arithmetic complexity is $O(n^3)$

Conjugate Gradients (CG)

- Idea: Use the residuals as basis vectors for the Gram-Schmidt conjugation

- The residual $\vec{d}^{(k)}$ is already orthogonal to all previous search directions ($\vec{v}^{(i)T}\vec{d}^{(k)} = 0 \;\forall i < k$), because of the A-orthogonality of the search directions, so the residual gives a new linearly independent search direction unless it is zero $\Rightarrow$ the problem is already solved

- The condition for the Gram-Schmidt constants

$$\beta_{kj} = -\frac{\vec{u}^{(k)T}A\vec{v}^{(j)}}{\vec{v}^{(j)T}A\vec{v}^{(j)}}$$

gives $\beta_{kj} = 0$ for $j < (k-1)$ if we use $u^{(k)} = d^{(k)}$

- We only need to make the new search direction $\vec{v}^{(k)}$ A-orthogonal to the last search direction $\vec{v}^{(k-1)}$ and get

$$\beta_k = \frac{\vec{d}^{(k)T}\vec{d}^{(k)}}{\vec{v}^{(k-1)T}\vec{d}^{(k-1)}} = \frac{\vec{d}^{(k)T}\vec{d}^{(k)}}{\vec{d}^{(k-1)T}\vec{d}^{(k-1)}}$$
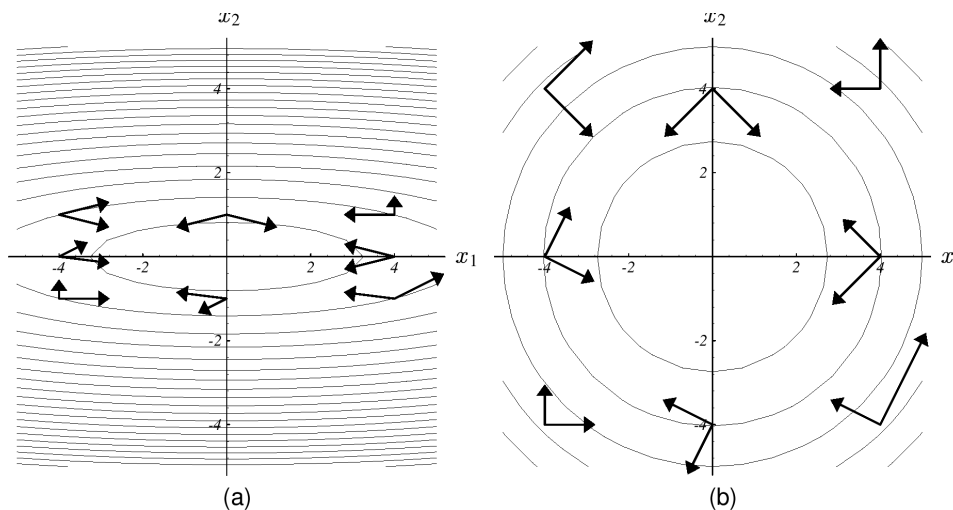
Figure 13: A-orthogonal search directions (left) are orthogonal in a stretched space where the hypersurface of $f(\vec{x})$ is spherical (from J. R. Shewchuk (1994): "An Introduction to the Conjugate Gradient Method Without the Agonizing Pain").

The Conjugate Gradient method uses a sequence of A-orthogonal search directions (Figure 13), using the residuals as basis for the creation of the search directions.

In exact arithmetic the minimum is found after at most N iterations (semi-iterative method, Figure 14). However round-off errors make CG an iterative method.

- We do not need to store previous search directions

- The memory consumption and the arithmetic complexity of one iteration is $O(n)$
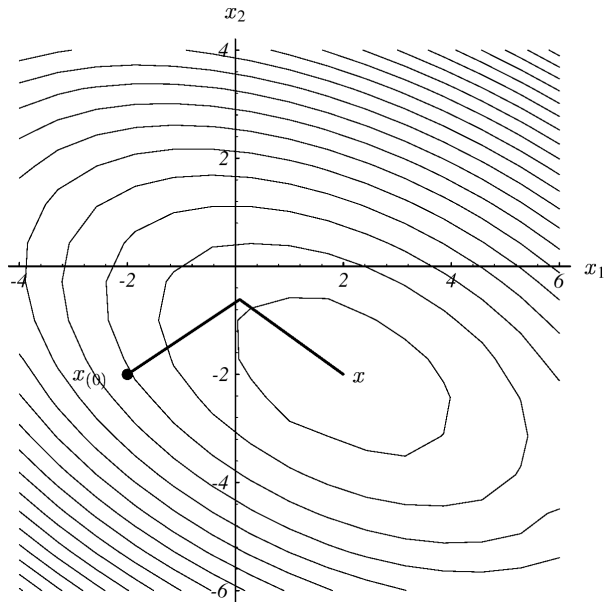
Figure 14: The method of conjugate gradients converges in exact arithmetics in n iterations (from J. R. Shewchuk (1994): "An Introduction to the Conjugate Gradient Method Without the Agonizing Pain").

CG Algorithm

$$\vec{v} = \vec{d} = \vec{b} - A\vec{x}$$
$$d_0 = \vec{d}^T \vec{d}$$
$$d_k = d_0;$$
$$\text{while } (d_k \geq \varepsilon^2 \cdot d_0)$$
$$\{$$
$$\quad \alpha = \left(\vec{d}^T \vec{d} \right) / \left(\vec{v}^T A \vec{v} \right)$$
$$\quad \vec{x} = \vec{x} + \alpha \vec{v}$$
$$\quad \vec{d}_{\text{new}} = \vec{d} - \alpha A \vec{v}$$
$$\quad \beta = \left(\vec{d}_{\text{new}}^T \vec{d}_{\text{new}} \right) / \left(\vec{d}^T \vec{d} \right)$$
$$\quad \vec{v} = \vec{d}_{\text{new}} + \beta \vec{v}$$
$$\quad \vec{d} = \vec{d}_{\text{new}}$$
$$\quad d_k = \vec{d}^T \vec{d}$$
$$\}$$

Optimised CG Algorithm

65

$$\vec{v} = \vec{d} = \vec{b} - A\vec{x}$$
$$d_0 = \vec{d}^T \vec{d};$$
$$d_k = d_0$$
while $(d_k \geq \varepsilon^2 \cdot d_0)$
$$\{$$
$$\quad \vec{t} = A\vec{v}$$
$$\quad \alpha = d_k / \left(\vec{v}^T \vec{t} \right)$$
$$\quad \vec{x} = \vec{x} + \alpha \vec{v}$$
$$\quad \vec{d} = \vec{d} - \alpha \vec{t}$$
$$\quad d_{k_{\text{old}}} = d_k;$$
$$\quad d_k = \vec{d}^T \vec{d}$$
$$\quad \beta = d_k / d_{k_{\text{old}}}$$
$$\quad \vec{v} = \vec{d} + \beta \vec{v}$$
$$\}$$
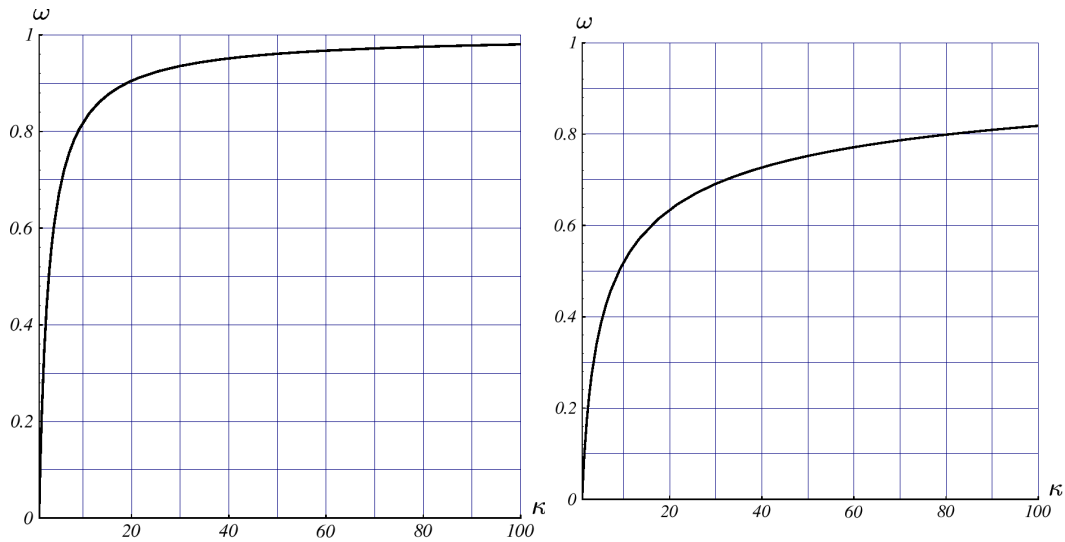
Figure 15: Condition dependent convergence rate of steepest descend (left) and conjugate gradients (right) (from J. R. Shewchuk (1994): "An Introduction to the Conjugate Gradient Method Without the Agonizing Pain").

Convergence of Conjugate Gradients

Convergence depends on the condition $\kappa(A)$ of the matrix, but less than in steepest descend (Figure 15). It also depends on the distribution of eigenvalues.

Complexity for discretisations of second-order elliptic PDE's

	two-dimensional	three-dimensional
Steepest Descent	$O(N^2)$	$O(N^{3/2})$
Conjugate Gradients	$O(N^{5/3})$	$O(N^{4/3})$

Convergence Rate

Steepest Descent

$$||\vec{e}^{(k)}||_A \le \left(\frac{\kappa - 1}{\kappa + 1}\right)^k ||\vec{e}^{(0)}||_A$$

Conjugate Gradients

$$||\vec{e}^{(k)}||_A \le 2 \left(\frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1}\right)^k ||\vec{e}^{(0)}||_A$$

with the "energy norm"

$$||\vec{e}||_A = \sqrt{\vec{e}^T A \vec{e}}$$
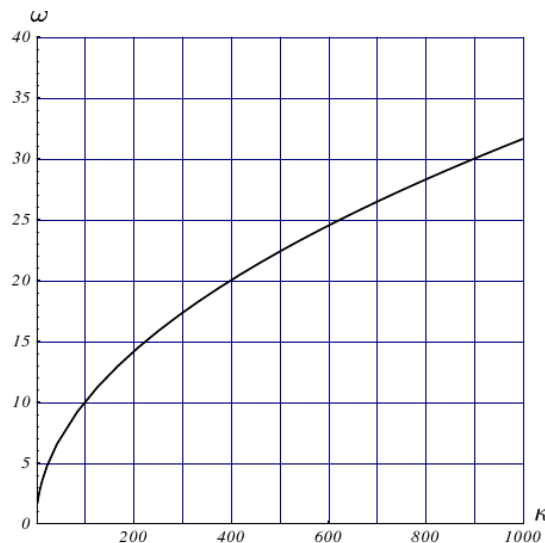
Figure 16: Quotient between convergence rate of steepest descend and conjugate gradients dependend on the condition (from J. R. Shewchuk (1994): "An Introduction to the Conjugate Gradient Method Without the Agonizing Pain").

Preconditioning

- While CG-methods usually have a better convergence than simple relaxation methods, the convergence still depends on the grid size for matrices generated by discretisations of partial differential equations.

- CG-methods therefore are often improved by using so-called preconditioning.

- Instead of $Ax = b$ we solve a system $M^{-1}Ax = M^{-1}b$, where the preconditioner M improves the distribution of eigenvalues or the condition of the matrix and thus provides an improved convergence behaviour.

- A^{-1} would be the optimal preconditioner as the eigenvalues of the resulting identity matrix I would all be identical and thus the system could be solved in one step, but it is of course too expensive to calculate.

- A simple possible choice is $M = D$ (so-called Jacobi preconditioning).

- The best choice is often a multigrid scheme for the coarse grid corrections.

- As the CG-method requires symmetric matrices, the SOR scheme can not be used. However, there is a variant called SSOR (symmetric SOR) which consists of a SOR step followed by a backward SOR step where we start with the last unknown and then decrement the indices.

$$M^{T^{-1}} M^{-1} A x = M^{T^{-1}} M^{-1} b$$

with $M = L + \omega^{-1} D$

Preconditioned CG Algorithm

$$\vec{d} = \vec{b} - A\vec{x}$$
$$\text{solve } M\vec{z} = \vec{d}$$
$$\vec{v} = \vec{z}$$
$$\rho_k = \rho_0 = \vec{d}^T \vec{z}$$
$$\text{while } (\rho_k \geq \varepsilon^2 \cdot \rho_0)$$
$$\{$$
$$\quad \vec{t} = A\vec{v}$$
$$\quad \alpha = \rho_k / \left(\vec{v}^T \vec{t} \right)$$
$$\quad \vec{x} = \vec{x} + \alpha \vec{v}$$
$$\quad \vec{d} = \vec{d} - \alpha \vec{t}$$
$$\quad \text{solve } M\vec{z} = \vec{d}$$
$$\quad \rho_{k_{\text{old}}} = \rho_k$$
$$\quad \rho_k = \vec{d}^T \vec{z}$$
$$\quad \beta = \rho_k / \rho_{k_{\text{old}}}$$
$$\quad \vec{v} = \vec{z} + \beta \vec{v}$$
$$\}$$

SSOR-Preconditioner

For the SSOR-preconditioner the step solve $M\vec{z} = \vec{d}$ is:

$$\vec{v} = 0$$
$$\text{for } (i = 0; i < n; ++i)$$
$$v_i = \omega \left(d_i - \sum_{j < i} a_{ij} v_j \right) / a_{ii}$$
$$\vec{\tau} = \vec{d} - A\vec{v}$$
$$\vec{\sigma} = 0$$
$$\text{for } (i = n - 1; i >= 0; --i)$$
$$\sigma_i = \omega \left(\tau_i - \sum_{j > i} a_{ij} \sigma_j \right) / a_{ii}$$
$$\vec{z} = \vec{v} + \vec{\sigma}$$
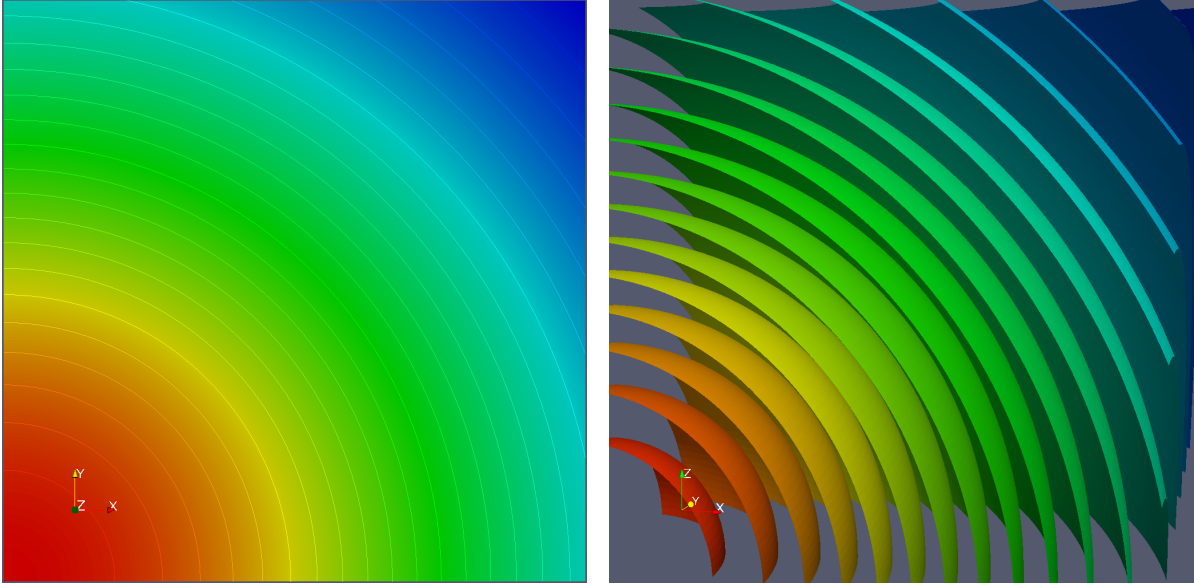
Figure 17: Solution of test case A in 2d and 3d.

More information on gradient based methods can be found in

J. R. Shewchuk (1994): "An Introduction to the Conjugate Gradient Method Without the Agonizing Pain"
  http://www.cs.cmu.edu/∼quake-papers/painless-conjugate-gradient.pdf

5.2.5 Numerical Results

The following examples show the convergence of some of the covered iteration schemes for test cases with varying difficulty in two and three space dimensions.

The tables contain the number of iterations necessary to reduce the initial defect by a factor of 10^8. The computation time is given in seconds (Core 2 Duo processor with 2.5 GHz, gcc-4.2 with -O2 optimisation). If an entry is missing the desired reduction could not be reached in 20000 iterations.

Test Case A

$$-\Delta u = (2d - 4\|x\|^2)e^{-\|x\|^2} \qquad \text{in } \Omega = (0,1)^d,$$
$$u = e^{-\|x\|^2} \qquad \text{on } \partial\Omega.$$

The exact solution is:
$$u(x) = e^{-\|x\|^2}.$$

Table 1: Convergence Results for Test Case A for triangles (P_1), rectangles (Q_1, 2d) and cuboids (Q_1, 3d).

	Test Case A, P_1, 2d													
h	Jacobi		Gauß-Seidel		Steepest Descent		SD+SSOR		CG		CG+SSOR		CG+ILU0	
	IT	Time	IT	Time	IT	Time	IT	Time	IT	Time	IT	Time	IT	Time
1/8	218		112		220		51		22		13		13	
1/16	840	0.02	427		854		177		48		26		24	
1/32	3165	0.21	1607	0.11	3230	0.12	645	0.07	98		49		45	
1/64	11820	3.04	6004	1.57	12096	1.74	2403	0.95	193	0.03	95	0.04	88	0.02
1/128							8955	13.9	378	0.24	184	0.30	172	0.20
1/256									739	2.25	359	2.58	336	2.18
	Test Case A, Q_1, 2d													
h	Jacobi		Gauß-Seidel		Steepest Descent		SD+SSOR		CG		CG+SSOR		CG+ILU0	
	IT	Time	IT	Time	IT	Time	IT	Time	IT	Time	IT	Time	IT	Time
1/8	147		75		113		24		16		10		8	
1/16	562	0.01	282		431		79		35		18		14	
1/32	2113	0.15	1056	0.08	1621	0.06	275	0.03	69		34		25	
1/64	7886	2.18	3939	1.10	6059	0.94	1011	0.43	136	0.03	64	0.03	46	0.01
1/128			14615	16.1			3741	6.42	266	0.18	120	0.22	87	0.10
1/256							13823	115	521	1.94	217	1.89	162	1.23
	Test Case A, Q_1, 3d													
h	Jacobi		Gauß-Seidel		Steepest Descent		SD+SSOR		CG		CG+SSOR		CG+ILU0	
	IT	Time	IT	Time	IT	Time	IT	Time	IT	Time	IT	Time	IT	Time
1/8	98	0.01	51		77		18		16		9		8	
1/16	376	0.24	189	0.12	290	0.10	55	0.05	34	0.01	17	0.02	15	0.01
1/32	1416	10.1	708	4.87	1087	4.10	187	1.95	67	0.26	32	0.34	27	0.25
1/64	5287	304.	2641	152.	4063	129.	681	65.6	132	4.43	59	5.86	51	4.18

Test Case B

$$-\Delta u = f \qquad\qquad \text{in } \Omega = (0,1)^d,$$
$$u = g \qquad\qquad \text{on } \Gamma_D,$$
$$-\nabla u \cdot \nu = j \qquad\qquad \text{on } \Gamma_N,$$

with

$$f(x) = \begin{cases} 50 & 0.25 \le x_0, x_1 \le 0.375 \\ 0 & \text{else} \end{cases},$$

$$\Gamma_N = \{x \mid x_1 = 0 \vee x_1 = 1 \vee (x_0 = 1 \wedge x_1 > 1/2)\} \qquad \Gamma_D = \partial\Omega \setminus \Gamma_N,$$

$$g(x) = e^{-\|x-x_0\|^2}, \qquad x_0 = (1/2, \ldots, 1/2)^T,$$

$$j(x) = \begin{cases} -5 & x_0 = 1 \wedge x_1 > 1/2 \\ 0 & \text{else} \end{cases}.$$

Test Case C

$$-\nabla \cdot \{k(x)\nabla u\} = 1 \qquad\qquad \text{in } \Omega = (0,1)^d,$$
$$u = 0 \qquad\qquad \text{on } \partial\Omega,$$

with

$$k(x) = \begin{cases}
20.0 & \lfloor x_0/H \rfloor \text{ even}, \lfloor x_1/H \rfloor \text{ even}, \lfloor x_2/H \rfloor \text{ even} \\
0.002 & \lfloor x_0/H \rfloor \text{ odd}, \lfloor x_1/H \rfloor \text{ even}, \lfloor x_2/H \rfloor \text{ even} \\
0.2 & \lfloor x_0/H \rfloor \text{ even}, \lfloor x_1/H \rfloor \text{ odd}, \lfloor x_2/H \rfloor \text{ even} \\
2000.0 & \lfloor x_0/H \rfloor \text{ odd}, \lfloor x_1/H \rfloor \text{ odd}, \lfloor x_2/H \rfloor \text{ even} \\
1000.0 & \lfloor x_0/H \rfloor \text{ even}, \lfloor x_1/H \rfloor \text{ even}, \lfloor x_2/H \rfloor \text{ odd} \\
0.001 & \lfloor x_0/H \rfloor \text{ odd}, \lfloor x_1/H \rfloor \text{ even}, \lfloor x_2/H \rfloor \text{ odd} \\
0.1 & \lfloor x_0/H \rfloor \text{ even}, \lfloor x_1/H \rfloor \text{ odd}, \lfloor x_2/H \rfloor \text{ odd} \\
10.0 & \lfloor x_0/H \rfloor \text{ odd}, \lfloor x_1/H \rfloor \text{ odd}, \lfloor x_2/H \rfloor \text{ odd}
\end{cases}.$$

Test Case D

$$-\nabla \cdot \{k(x)\nabla u\} = 0 \qquad\qquad \text{in } \Omega = (0,1)^d,$$
$$u = g \qquad\qquad \text{on } \Gamma_D,$$
$$-\nabla u \cdot \nu = 0 \qquad\qquad \text{on } \Gamma_N,$$

with

$$\Gamma_D = \{x \mid x_0 = 0 \vee x_0 = 1\} \qquad\qquad \Gamma_N = \partial\Omega \setminus \Gamma_D,$$

$$g(x) = \begin{cases} 1 & x_0 = 0 \\ 0 & x_0 = 1 \end{cases}.$$

The function $k(x)$ is log-normal distributed with a given mean of 0, a variance of 3 (i.e. the permeabilities are random variables mostly in the range 10^{-3} and 10^3) and a correlation length of 1/64 in 2d and 1/32 in 3d. Examples are shown in figure 19.

Table 2: Convergence Results for Test Case B for triangles (P_1), rectangles (Q_1, 2d) and cuboids (Q_1, 3d).

	Test Case B, P_1, 2d												
h	Jacobi		Gauß-Seidel		Steepest Descent		SD+SSOR		CG		CG+SSOR		CG+ILU0	
	IT	Time	IT	Time	IT	Time	IT	Time	IT	Time	IT	Time	IT	Time
1/8	667		338		830		138		41		18		16	
1/16	2619	0.04	1327	0.02	2969	0.03	525	0.01	82		35		32	
1/32	10009	0.60	5075	0.32	10778	0.40	2017	0.20	159		68		62	
1/64			19131	4.57			7637	2.81	306	0.05	133	0.05	124	0.04
1/128									590	0.36	259	0.39	244	0.28
1/256									1143	3.45	505	3.47	478	3.08
	Test Case B, Q_1, 2d												
h	Jacobi		Gauß-Seidel		Steepest Descent		SD+SSOR		CG		CG+SSOR		CG+ILU0	
	IT	Time	IT	Time	IT	Time	IT	Time	IT	Time	IT	Time	IT	Time
1/8	456		230		424		65		32		14		11	
1/16	1770	0.07	888	0.02	1504	0.01	237		59		24		18	
1/32	6720	0.43	3364	0.21	5436	0.22	877	0.09	112		45		32	
1/64			12614	3.20	19895	3.11	3249	1.28	215	0.04	87	0.04	61	0.02
1/128							12055	18.8	415	0.28	168	0.27	118	0.13
1/256									806	2.88	328	2.63	231	1.71
	Test Case B, Q_1, 3d												
h	Jacobi		Gauß-Seidel		Steepest Descent		SD+SSOR		CG		CG+SSOR		CG+ILU0	
	IT	Time	IT	Time	IT	Time	IT	Time	IT	Time	IT	Time	IT	Time
1/8	180	0.01	92		176		29		29		12		10	
1/16	694	0.42	349	0.21	596	0.22	95	0.09	54	0.02	22	0.02	19	0.01
1/32	2622	17.6	1313	8.74	2126	7.86	343	3.54	102	0.39	42	0.44	35	0.32
1/64	9813	531.	4908	263.	7747	240.	1269	119.	197	6.42	80	7.70	67	5.40
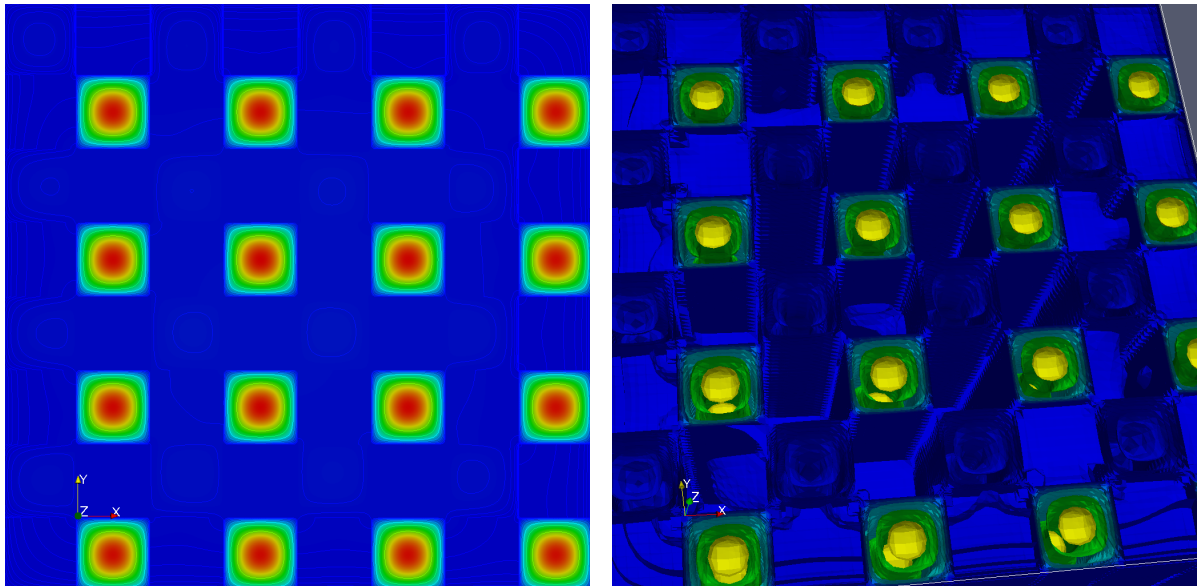
Figure 18: Solution of test case C in 2d and 3d.

72

Table 3: Convergence Results for Test Case C for rectangles (Q_1, 2d) and cuboids (Q_1, 3d).

Test Case C, Q_1, 2d														
h	Jacobi		Gauß-Seidel		Steepest Descent		SD+SSOR		CG		CG+SSOR		CG+ILU0	
	IT	Time	IT	Time	IT	Time	IT	Time	IT	Time	IT	Time	IT	Time
1/8	4665	0.06	2354	0.01	3334	0.01	724		27		17		8	
1/16			13573	0.26			4335	0.12	281		38		27	
1/32							17512	1.91	1761	0.08	73		52	
1/64									8644	1.48	142	0.06	99	0.03
1/128											282	0.49	196	0.22
1/256											577	4.82	405	2.96
Test Case C, Q_1, 3d														
h	Jacobi		Gauß-Seidel		Steepest Descent		SD+SSOR		CG		CG+SSOR		CG+ILU0	
	IT	Time	IT	Time	IT	Time	IT	Time	IT	Time	IT	Time	IT	Time
1/8	127	0.01	65		96		22		21		10		8	
1/16	1326	0.83	667	0.42			208	0.20	1179	0.45	32	0.03	23	0.02
1/32	9966	68.2	4996	34.8			1425	14.8	8594	32.9	71	0.76	56	0.51
1/64							8382	792.			151	14.6	124	9.96
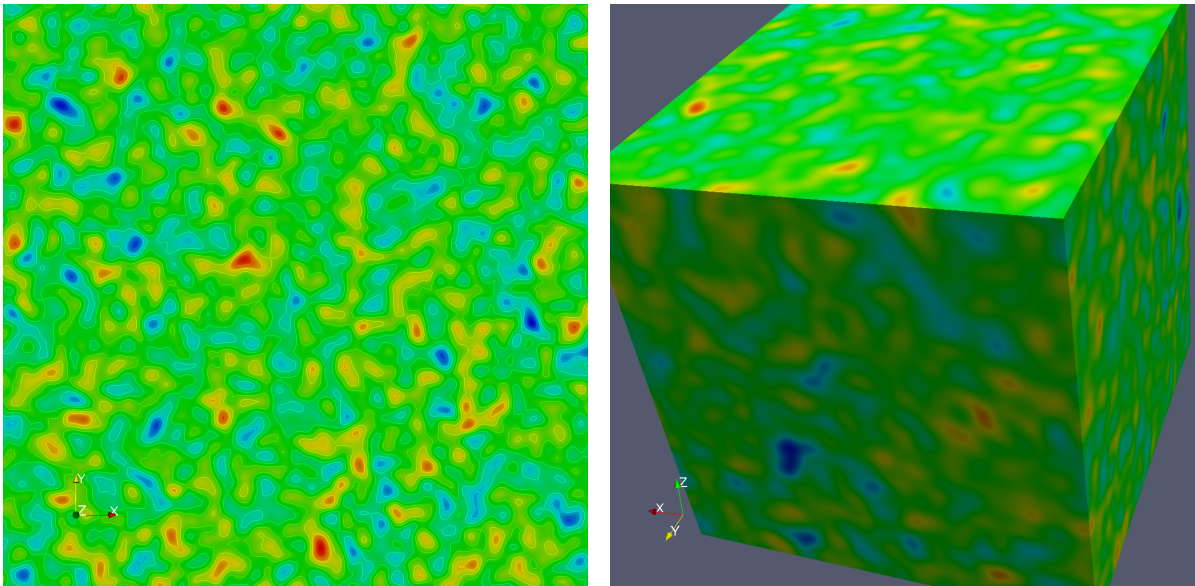
Figure 19: Log-normal distributed permeability fields in 2d and 3d.
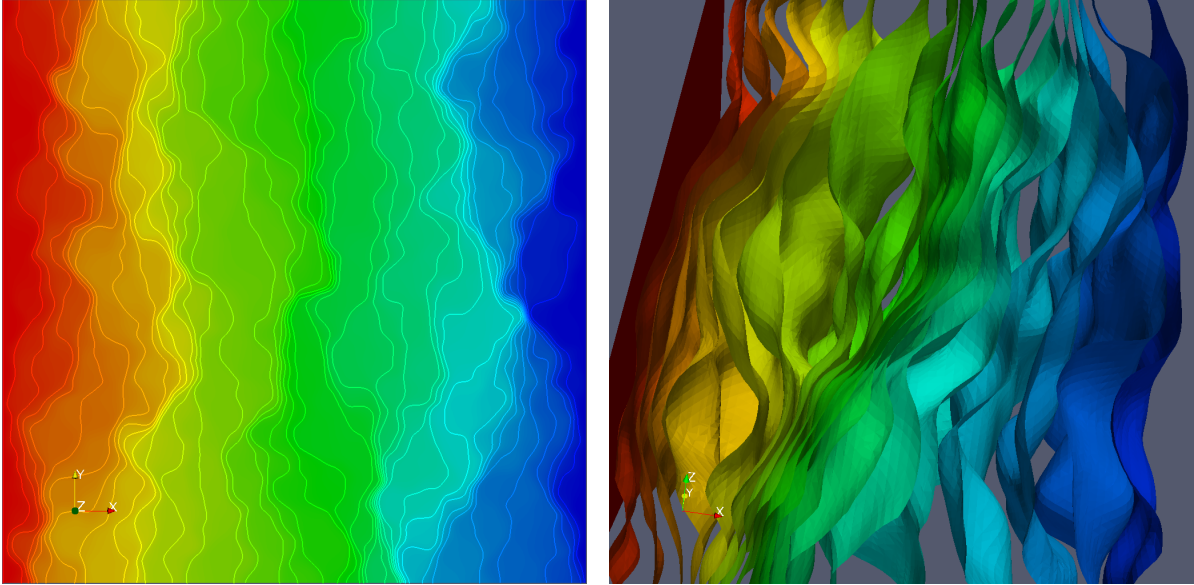
Figure 20: Solution of Test Case D in 2d and 3d.

Table 4: Convergence Results for Test Case D for rectangles (Q_1, 2d) and cuboids (Q_1, 3d).

Test Case D, Q_1, 2d														
h	Jacobi		Gauß-Seidel		Steepest Descent		SD+SSOR		CG		CG+SSOR		CG+ILU0	
	IT	Time	IT	Time	IT	Time	IT	Time	IT	Time	IT	Time	IT	Time
1/64							11307	4.58	1825	0.31	193	0.08	110	0.03
1/128									5755	3.87	375	0.62	250	0.28
1/256									15489	57.2	707	5.72	492	3.67
1/512										385.	1345	53.6	955	35.2
Test Case D, Q_1, 3d														
h	Jacobi		Gauß-Seidel		Steepest Descent		SD+SSOR		CG		CG+SSOR		CG+ILU0	
	IT	Time	IT	Time	IT	Time	IT	Time	IT	Time	IT	Time	IT	Time
1/16	2538	1.52	1280	0.78			395	0.37	452	0.17	48	0.05	36	0.03
1/32	10096	67.8	5069	34.0			1401	14.6	2190	8.48	88	0.93	73	0.69
1/64			19158	1046			4905	469.	5859	195.	166	16.3	140	11.9

Test Case E

$$-\nabla \cdot \{K(x)\nabla u\} = 1 \qquad \qquad \text{in } \Omega = (0,1)^d,$$
$$u = 0 \qquad \qquad \text{on } \partial\Omega,$$

with $K(x)$ a diagonal tensor

$$K_{ij}(x) = \begin{cases} 10^{-6} & i = j = 0 \\ 1 & i = j > 0 \\ 0 & \text{else} \end{cases}.$$

Attention: The matrix is symmetric and positive definite for Q_1 but not irreducible diagonally dominant. Jacobi iteration does not converge for every s.p.d. matrix without damping, this explains the problems with the Jacobi iteration for Q_1 elements. The matrix is approximately tridiagonal in 2d, the ILU$_0$ method with the correct ordering is exact for tridiagonal matrices.

6 Simulation of Groundwater Flow

6.1 Boundary Conditions

There are two cases where simple Dirichlet or Neumann boundary conditions are not sufficient.

6.1.1 Heterogeneous Systems

The naive use of a Neumann boundary condition for a heterogeneous porous media can yield surprising results. A Neumann boundary condition forces exactly the same flux into (or out of) each element. If the permeability of the element is very low, this is compensated by a huge pressure gradient, which can be absolutely unrealistic. This is usually not what happens in nature.

Two possibilities to avoid this are

- weight the flux with the permeability of the element

$$j_{\text{boundary}} = j_{\text{Neumann}} \cdot K_e \cdot \frac{\sum_{\text{boundary elements}} A_i}{\sum_{\text{boundary elements}} A_i K_i}$$

- Add a redistribution layer with high conductivity at the boundary.

6.1.2 Vertical Boundaries

If gravity is taken into account the steady state solution is a pressure which is increasing with depth so that the pressure gradient compensates the driving effect of gravity. If vertical cuts in two dimensions or three-dimensional regions are simulated, this has to be taken into account when Dirichlet boundary conditions should be specified on vertical boundaries.

If the z-Axis is in pointing upwards with the zero coordinate at the bottom of the domain, the boundary condition should be

$$p_{\text{boundary}} = p_{\text{Dirichlet}} - \rho_w g z$$

Table 5: Convergence Results for Test Case E for triangles (P_1), rectangles (Q_1, 2d) and cuboids (Q_1, 3d).

Test Case E, P_1, 2d, space depth-first ordering														
h	Jacobi		Gauß-Seidel		Steepest Descent		SD+SSOR		CG		CG+SSOR		CG+ILU0	
	IT	Time	IT	Time	IT	Time	IT	Time	IT	Time	IT	Time	IT	Time
1/8	233		119		228		50		8		17		9	
1/16	946	0.02	481	0.01	946	0.01	180	0.01	16		36		35	
1/32	3798	0.24	1930	0.12	3834	0.14	638	0.07	32		76	0.01	89	
1/64	15203	3.79	7724	1.94	15422	2.22	2362	0.96	66	0.01	157	0.07	183	0.05
1/128							9020	14.3	173	0.11	318	0.52	373	0.44
1/256									386	1.16	674	4.94	756	4.97
Test Case E, Q_1, 2d, lexicographic ordering														
h	Jacobi		Gauß-Seidel		Steepest Descent		SD+SSOR		CG		CG+SSOR		CG+ILU0	
	IT	Time	IT	Time	IT	Time	IT	Time	IT	Time	IT	Time	IT	Time
1/8			186		524		46		16		20		2	
1/16			756	0.02	2102	0.02	176		75		43		2	
1/32			2949	0.19	8250	0.33	666	0.07	255	0.01	89	0.01	2	
1/64			11423	2.89			2614	1.03	547	0.09	175	0.07	2	
1/128							10102	15.7	1106	0.74	344	0.55	3	
1/256									2188	7.72	664	5.19	3	0.02
Test Case E, Q_1, 3d, lexicographic ordering														
h	Jacobi		Gauß-Seidel		Steepest Descent		SD+SSOR		CG		CG+SSOR		CG+ILU0	
	IT	Time	IT	Time	IT	Time	IT	Time	IT	Time	IT	Time	IT	Time
1/8			127	0.01	264	0.01	34		26		18		8	
1/16			505	0.30	1046	0.38	122	0.11	84	0.04	37	0.04	14	0.01
1/32			1952	12.8	4100	15.2	458	4.71	209	0.80	73	0.76	24	0.22
1/64			7582	404.	16014	495.	1796	169.	422	13.8	143	13.8	44	3.72

6.2 Wells

An important aspect of groundwater flow modeling is the assessment of human influence on the groundwater by extracting water from wells (or by water injection with wells).

6.2.1 Wells in Simulations of Horizontal Flow

In 2D simulations of horizontal groundwater flow wells can be represented as point sinks or sources (for extraction or injection wells).

While the source term in the groundwater flow equation expects a source density r_w in m/s the data is usually a flow rate q_w in m^3/s. It would be natural to divide the flow rate by the volume of the well. However, as the well is usually not exactly resolved in the simulation, the flow rate in the simulation would be wrong by $V_{\mathsf{sim.well}}/V_{\mathsf{realwell}}$ Therefore it is necessary to divide q_w by the volume V of simulated well, which is just the volume of the element in which the point source is located: $r_w = \frac{q_w}{V_e}$.

Figure 21: Pressure distribution with flux vectors (left) and flux density with pressure isolines (right) around a well in a horizontal simulation of groundwater flow

Figure 21 shows a example simulation with a pressure gradient from left to right and no-flow boundary conditions at bottom and top. As the domain is homogeneous the pressure isolines are straight lines from bottom to top. The well is situated in the middle of the domain. The well creates a depression in the pressure field and a high flux density around the well, which is due to the decreasing cross-section through which the flow has to be extracted.

If the aquifer is heterogeneous with a log-normal permeability distribution (Figure 22) the pressure distribution is more complicated with a locally high pressure gradient compensating regions with low permeability. Streamlines originating on the left boundary show that the flow concentrates nevertheless on high permeability regions. Some streamlines end in the well indicating the region from which the water is drawn. Due to the Dirichlet boundary conditions the flux density is very heterogeneous over the domain.

6.2.2 Wells in Simulations of Vertical Flow

In 2D simulations of a vertical cut through an aquifer or in 3D simulations wells have to be represented in more detail. However, as they are usually thin compared to the size of the domain, they can be assumed to be line sources/sinks.

It is then necessary to distribute the extracted amount of water over the volume of the elements contributing to the line source on the grid $r_w = \frac{q_w}{\sum_{\mathsf{involved\ elements}} V_e}$
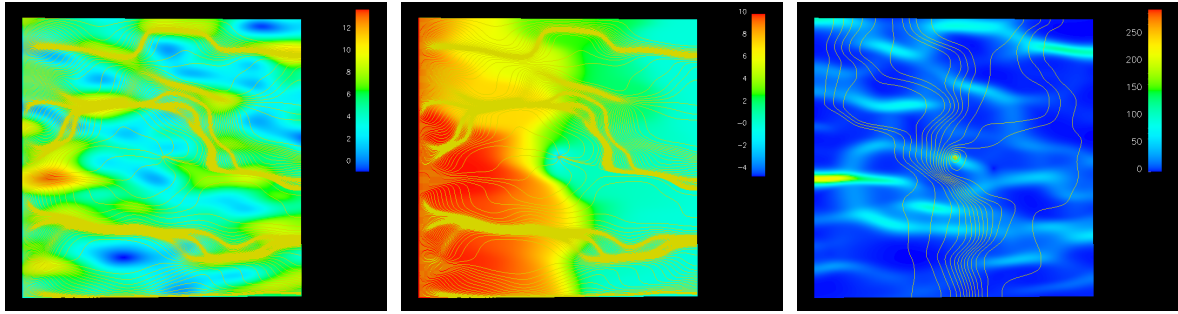
Figure 22: Permeability and pressure distribution with streamlines and flux density with pressure isolines around a well in a horizontal simulation of groundwater flow in a heterogeneous aquifer.

The simulation of horizontal flow in a vertical cut through a homogeneous aquifer with a line sink in the middle of the domain (Figure 23) is a good example that two-dimensional simulations can be misleading. The pressure isosurfaces are straight lines this would be expected if the well is a trench of infinite length. The real solution should show an increasing pressure gradient closer to the well due to the decreasing flow cross-section as obtained in Figure 21. This can be rectified by either using a radially symmetric coordinate system or by performing a three-dimensional simulation.
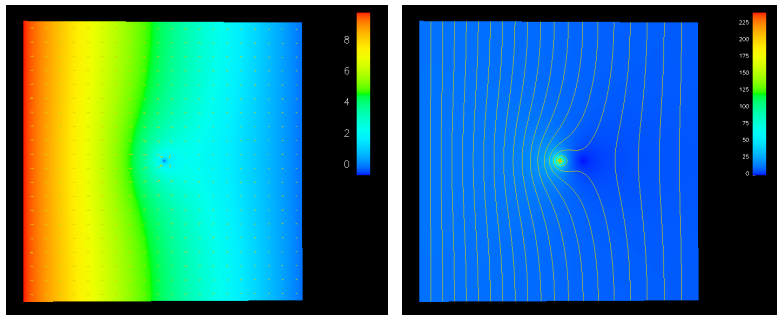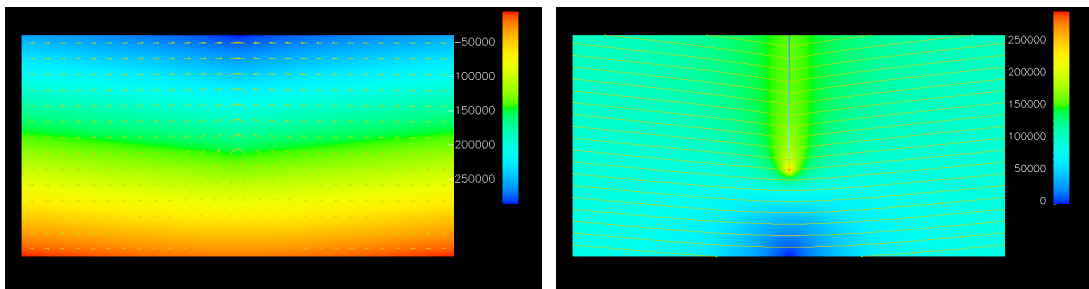

Figure 23: Pressure distribution with flux vectors (left) and flux density with pressure isolines (right) around a well in a horizontal simulation of groundwater flow along a vertical cut.

If the aquifer is heterogeneous a second problem occurs (Figure 24). If the same source density is used everywhere even in regions with a very low permeability a huge pressure gradient has to be applied. However, in a real system, the water would just be extracted easily from a region with high permeability.

Alternatives

There are different possibilities to obtain a more correct result:

- Perform a weighting with the conductivity K_e of the elements:

$$r_w = q_w \frac{K_e}{\sum_{\text{involved elements}} K_i V_i}$$

78
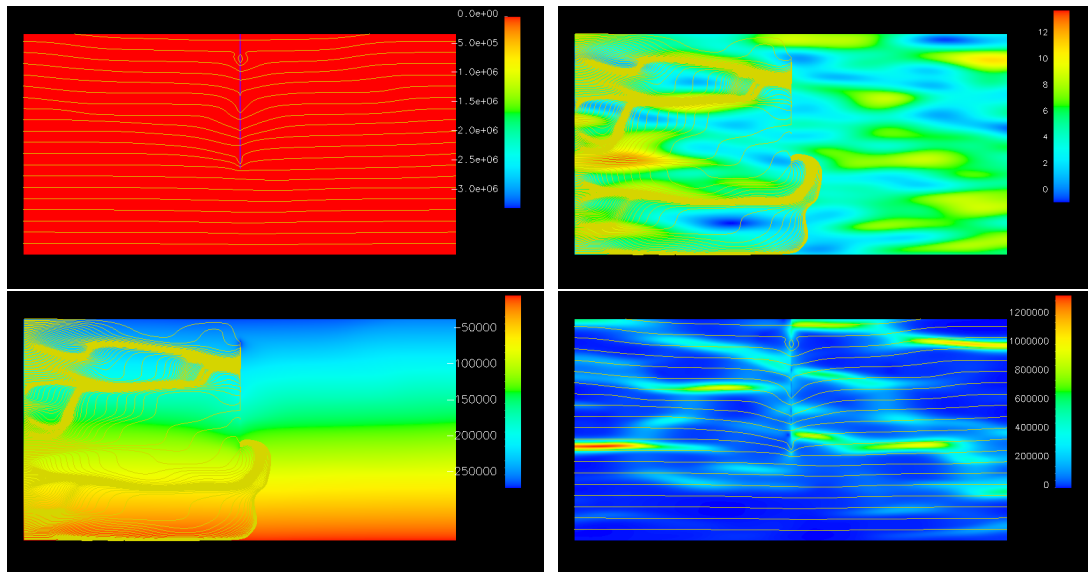
Figure 24: Source density with pressure isosurfaces (upper left), log-normal permeability distribution with streamlines (upper right), pressure distribution with streamlines (lower left) and flux density with pressure isosurfaces (lower right) for horizontal flow across a vertical cut with a line source in a heterogeneous aquifer.

- Add the well as line of elements with very high permeability (the bore hole) and either a Neumann boundary condition at the point where the well hits the boundary or a line sink/source

- Add an anisotropic higher permeability in the direction of the well along all elements containing the well.

- Add the well as a lower-dimensional element (a line) coupled to the volume simulation

6.3 Fractures

- Fractures can be very important in certain rock formations (limestone, granite . . .).

- They can provide a path for very rapid solute transport.

- In small scale simulations fractures can be resolved explicitly by adding areas of high permeability (if the fractures are always water filled)

- In large scale simulations this is hard to realize as fractures are very thin compared to the size of the domain would require very anisotropic elements

- One possibility is a dual continuum model, where the fracture domain is a separate continuous porous medium with a rate limited (solute) exchange with the matrix domain. This includes the assumption, that there are many small fractures so they can be represented on a continuum scale at the level of interest

79

- Another possibility is the representation of the fractures as one- or two-dimensional objects in a two- or three-dimensional space

6.4 Interpolation of the Flux Field

The Finite-Volume scheme only gives the normal fluxes at the interfaces. However for a visualization of the flux field we need the flux vector and for the calculation of solute transport on a grid not identical to the grid used in the water transport calculations we also need to interpolate the flux vector.

This can be done by using RT_0 Raviart-Thomas elements with the Ansatz

$$\vec{j} = \begin{pmatrix} ax + b \\ cy + d \\ ez + f \end{pmatrix}$$

The coefficients for the flux vector calculation on each grid cell can easily be calculated from the normal fluxes:

$$
\begin{aligned}
j_{k-\frac{1}{2},l_x} &= a_{k,l} x_{k-\frac{1}{2},l} + b_{k,l} \\
j_{k+\frac{1}{2},l_x} &= a_{k,l} x_{k+\frac{1}{2},l} + b_{k,l} \\
j_{k+\frac{1}{2},l_x} - j_{k-\frac{1}{2},l_x} &= a_{k,l} \left(x_{k+\frac{1}{2},l} - x_{k-\frac{1}{2},l} \right) \\
a_{k,l} &= \frac{j_{k+\frac{1}{2},l_x} - j_{k-\frac{1}{2},l_x}}{h_{k,l_x}} \\
b_{k,l} &= j_{k-\frac{1}{2},l_x} - \frac{j_{k+\frac{1}{2},l_x} - j_{k-\frac{1}{2},l_x}}{h_{k,l_x}} x_{k-\frac{1}{2},l}
\end{aligned}
$$

7 Parabolic PDEs - Heat Transport

7.1 Heat Transport in Porous Media

7.1.1 Flux Law

Heat is transported in a saturated porous medium either by convection of the liquid phase or by heat conduction. This processes can be described by:

$$\vec{J_h} = \vec{J}_{h_{\text{conv}}} + \vec{J}_{h_{\text{cond}}} \tag{11}$$

where

$$
\begin{aligned}
\vec{J}_{h_{\text{conv}}} &= T \cdot C_w \cdot \vec{J_w} \tag{12} \\
\vec{J}_{h_{\text{cond}}} &= -\lambda(\theta_w) \cdot \nabla T \tag{13}
\end{aligned}
$$

with:

T	Temperature	[K]
C_w	Volumetric heat capacity of water	[J m^{-3} K^{-1}]
J_w	volumetric water flux	[m s^{-1}]
$\lambda(\theta_w)$	Heat conductivity	[W m^{-1} K^{-1}]

7.1.2 Heat Capacity

Heat Capacity

The thermal energy content of a soil can be calculated as

$$E_h(\vec{x}) = C_{\text{tot}} \cdot T,$$

where C_{tot} is the total heat capacity of the soil The heat capacity of a soil can be computed from porosity Φ, water content and the heat capacity of the components.

$$C_{\text{tot}} \quad = \quad \theta_w C_w + (\Phi - \theta_w)C_g + (1 - \Phi) \cdot C_s \tag{14}$$

with:

C_g	Volumetric heat capacity of the gas phase	[J m^{-3} K^{-1}]
C_s	Volumetric heat capacity of the matrix	[J m^{-3} K^{-1}]

Typical Values:

C_{quartz}	2.23MJ m^{-3} K^{-1}
C_{water}	4.18 MJ m^{-3} K^{-1}
C_{air}	0.00117 MJ m^{-3} K^{-1}

Thus a saturated soil with a porosity of 33 per cent would have a heat capacity of approximately 2.88 MJ m^{-3} K^{-1}

7.1.3 Heat Conductivity

The heat conductivity of a porous medium depends not only on its composition, but also on the geometry of the pore space and the distribution of the phases. The problem is simplified by the strong dissipative nature of heat transport.

De Vries [dV52] developed a method to estimate the composition dependence of heat conductivity[3]. In analogy to the description of polarization, heat conductivity can be estimated with the formula

Heat Conductivity

$$\lambda = \frac{\sum_{i=0}^{N} k_i X_i \lambda_i}{\sum_{i=0}^{N} k_i X_i} \tag{15}$$

with
	k_i:	Ratio of the average temperature gradient in particles of type i to the average temperature gradient in the surrounding medium [–]
	X_i:	volume fraction of component i [–]
	λ_i:	heat conductivity of component i [W m^{-1}K^{-1}]

Typical values:

λ_{quartz}	6.1-9.5 W m^{-1}K^{-1}
λ_{water}	0.57 W m^{-1}K^{-1}
λ_{air}	0.025 W m^{-1}K^{-1}

---

[3]The heat conductivity is also temperature dependent. This dependence is not considered.
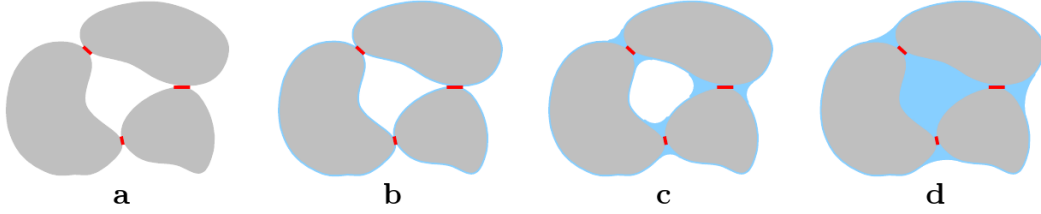
Figure 25: Sketch for dependence of thermal conductivity on water content in a coarse-textured porous medium. The contact between the grains is restricted to small regions (red) and the corresponding cross-sectional area is limiting for heat flow in a completely dry medium (a). As the water content increases, the pathways widen considerably thereby leading to a higher conductivity (b...d). (from K. Roth (2005), Soil Physics - Lecture Notes v1.0, Institut für Umweltphysik, Universität Heidelberg)

Heat Conductivity

The value of k_i depends on the ratio λ_i/λ_0,[4] and the size, form and position of the particles. If they are assumed to be ellipsoids with a distance large enough to be treated independently, k_i can be calculated:

$$k_i = \frac{1}{3} \sum_{l=a,b,c} \left[1 + \left(\frac{\lambda_i}{\lambda_0} - 1 \right) g_l \right]^{-1} \tag{16}$$

g_a, g_b, g_c are dimensionless form factors, depending on the ratio of the axes a, b and c of the ellipsoid. Their sum is equal to one. If two axes are equal, their form factors are equal as well. For spherical particles $g_a = g_b = g_c = 1/3$.

Both assumptions are clearly not valid for a natural porous medium, but according to de Vries theoretical reasons as well as measurements hint at an applicability of equation 16. My own research [ICR98] showed a good agreement of this approximation with results obtained from simulations explicitly considering the structure of a soil sample.

The heat conductivity can only be calculated with equations 15 and 16 for fully saturated porous media or completely dry soils. For water contents in between, the form factors g_a, g_b, g_c for air bubbles are necessary. De Vries [dV63] gives in example 7.6.1 a method to estimate them. Additionally, the increase in heat conductivity of the gas phase due to water vapour transport must be considered.

7.1.4 Heat Transport Equation

$$\frac{\partial E_h(\vec{x})}{\partial t} + \nabla \cdot \vec{J}_h(\vec{x}) + r_h(\vec{x}) = 0 \tag{17}$$

The time derivative of the thermal energy content is:

$$\frac{\partial E_h(\vec{x})}{\partial t} = \frac{\partial \left(C_{\text{tot}}(\vec{x})T \right)}{\partial t}. \tag{18}$$

If we neglect the temperature dependence of the heat capacity, we get

$$\frac{\partial E_h(\vec{x})}{\partial t} = C_{\text{tot}}(\vec{x}) \frac{\partial T}{\partial t} \tag{19}$$

---

[4]λ_0 is the heat capacity of the surrounding medium.
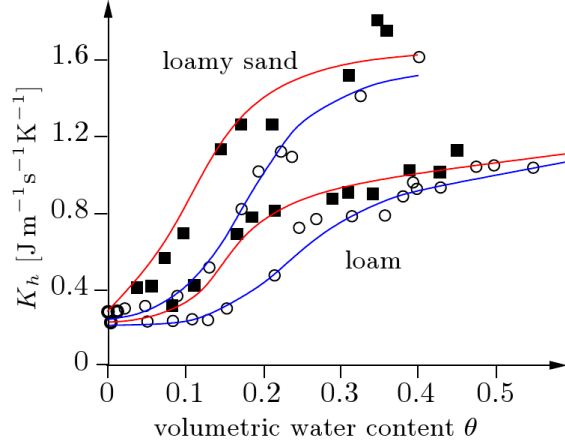
Figure 26: Measured values of thermal conductivity λ for two soils at 25℃(open circle) and at 40℃(closed squares). The solid lines are parametrized with the model of de Vries (from K. Roth (2005), Soil Physics - Lecture Notes v1.0, Institut für Umweltphysik, Universität Heidelberg)

yielding the heat transport equation

$$C_{\text{tot}}(\vec{x})\frac{\partial T(\vec{x})}{\partial t} - \nabla \cdot (\lambda(\vec{x}, \theta_w)\nabla T(\vec{x})) + C_w \nabla \cdot \left(T(\vec{x})\vec{J_w}(\vec{x}) \right) + r_h(\vec{x}) = 0 \qquad (20)$$

The heat transport equation is a parabolic equation as e.g. for constant heat conductivity and constant water flux density in one dimesion:

$$-\lambda\frac{\partial^2 T(x)}{\partial x^2} + C_w J_w(x)\frac{\partial T(x)}{\partial x} + C_{\text{tot}}(x)\frac{\partial T(x)}{\partial t} + r_h(x) = 0 \qquad (21)$$

$$\det \begin{pmatrix} -\lambda & 0 \\ 0 & 0 \end{pmatrix} = 0 \qquad (22)$$

and

$$\text{Rank} \begin{bmatrix} -\lambda & 0 & C_w J_w(x) \\ 0 & 0 & C_{\text{tot}}(x) \end{bmatrix} = 2 \qquad (23)$$

7.2 Solution with Fourier Series

We want to analyse the one-dimensional problem: Find $u(x, t)$ such that

$$\frac{\partial u}{\partial t} - \frac{\partial^2 u}{\partial x^2} = 0 \qquad \text{in } (0, 1) \times (0, \infty) \qquad (24a)$$

$$u(x, 0) = f(x) \qquad \text{for } t = 0 \quad \text{(initial condition)}, \qquad (24b)$$

$$\left.\begin{aligned} u(0, t) &= 0 \\ u(1, t) &= 0 \end{aligned}\right\} \qquad \text{(boundary condition)}. \qquad (24c)$$

One approach to obtain a solution is the separation of the variables. We use the trial function

$$u(x,t) = X(x) \cdot T(t).$$

with this we get

$$\frac{\partial u}{\partial t} = XT' \quad \text{and} \quad \frac{\partial^2 u}{\partial x^2} = X''T.$$

If we insert the trial function in the PDE we get

$$XT' - X''T = 0 \iff \frac{T'(t)}{T(t)} = \frac{X''(x)}{X(x)}, \tag{25}$$

under the condition that $u(x,t) = X(x) \cdot T(t) \neq 0$.

The left side of (25) is independent of x, the right side is independent of t. If both sides have to be equal for *all* x, t the only possible solution is

$$\frac{T'(t)}{T(t)} = \frac{X''(x)}{X(x)} = \lambda \quad \text{(const).}$$

With this we get

$$T'(t) = \lambda T(t) \quad \Rightarrow \quad T(t) = c_1 e^{\lambda t}$$
$$X''(x) = \lambda X(x) \quad \Rightarrow \quad X(x) = c_2 e^{\sqrt{\lambda} x} + c_3 e^{-\sqrt{\lambda} x} \qquad \text{it is the same } \lambda!$$

So

$$u(x,t) = e^{\lambda t} \left(A e^{\sqrt{\lambda} x} + B e^{-\sqrt{\lambda} x} \right).$$

To fulfil the boundary conditions (24c), we set

$$\left. \begin{array}{r} A = a + ib \\ B = a - ib \end{array} \right\} \quad \Rightarrow \quad A + B = 2a \stackrel{!}{=} 0 \qquad A - B = i2b$$

and $\lambda = -n^2 \pi^2$, $n \in \mathbb{N}$ ($\rightsquigarrow \sin n\pi x$).

With this we get

$$A e^{\sqrt{\lambda} x} + B e^{-\sqrt{\lambda} x} = A e^{\overbrace{in\pi}^{\sqrt{-n^2\pi^2}=}} {}^{x} + B e^{-in\pi x}$$
$$= A (\cos n\pi x + i \sin n\pi x) + B \left(\underbrace{\cos(-n\pi x)}_{=\cos n\pi x} + i \underbrace{\sin(-n\pi x)}_{=-\sin n\pi x} \right)$$
$$= (A + B) \cos n\pi x + i(A - B) \sin n\pi x$$
$$= \underbrace{2a \cos n\pi x}_{0} - \underbrace{2b \sin n\pi x}_{\text{fulfills the bc}}$$

Thus we get for each $n \in \mathbb{N}$ a solution which fulfils the boundary conditions of the form:

$$u(x,t) = -2b e^{-n^2\pi^2 t} \sin n\pi x.$$

To fulfil the initial conditions (24b) we develop f in a Fourier series

$$f(x) = \sum_{n=1}^{\infty} A_n \sin n\pi x$$

with this

$$u(x,t) = \sum_{n=1}^{\infty} A_n e^{-n^2\pi^2 t} \sin n\pi x$$

is a solution of the parabolic model problem (equation (24)).

Remark 7.1. If $u(x,t)$ is defined in this way it is no classical solution in $C^2\big(\Omega \times (0,\infty)\big)$. For each t $u(\cdot,t)$ is a function in $L^2(\Omega)$, as it is the limit of a Fourier series.

Remark 7.2. Due to the n^2 term in the e-function high „frequency" parts (large n) of the initial condition are damped much more efficiently and faster than low frequency parts. This is called the „smoothing property" of parabolic problems.

7.3 Finite Differences Approach for Parabolic Problems

We limit ourselves to the one-dimensional problem

$$
\begin{aligned}
\frac{\partial u}{\partial t} - \frac{\partial^2 u}{\partial x^2} &= f \qquad \text{in } \Omega \times T, \qquad \Omega = (0,1), T = (0, T_{end}) \\
u &= g \qquad \text{on } \partial\Omega \\
u &= u_0 \qquad \text{for } t = 0.
\end{aligned}
\tag{26}
$$

Discretisation is done with the so-called Method of Lines, i.e. first a spatial discretisation is applied then a discretisation in time.

Spatial discretisation: Finite Differences with grid $x_i = i \cdot h$, $h = \frac{1}{N}$, $i = 0, \ldots, N$.

Taylor series for $\frac{\partial^2 u}{\partial x^2}$ at point (x_i, t) yields:

$$\frac{\partial u(x_i,t)}{\partial t} = \underbrace{\frac{1}{h^2}\big[u(x_{i-1},t) - 2u(x_i,t) + u(x_{i+1},t)\big] + f(x_i,t) + O(h^2)}_{F(x_i,t)} \quad i = 1,\ldots,N-1 \tag{27}$$

This is a coupled system of ordinary differential equations for the $N-1$ unknown functions „$u_i(t) = u(x_i,t)$".

For the time discretisation we use the grid $t^k = k \cdot \tau$, $\tau = \frac{T_{end}}{K}$, $k = 0, \ldots, K$.

Onestep-θ-Method: Numerical integration yields:

$$\frac{\partial u(x_i,t)}{\partial t} = F(x_i,t) \qquad i = 1,\ldots,N-1$$

$$\Rightarrow \int_{t^k}^{t^{k+1}} \frac{\partial u(x_i,t)}{\partial t}\,\mathrm{d}t = \int_{t^k}^{t^{k+1}} F(x_i,t)\,\mathrm{d}t$$

$$\Longleftrightarrow u(x_i,t^{k+1}) - u(x_i,t^k) = \tau\Big[(1-\theta)F(x_i,t^k) + \theta \cdot F(x_i,t^{k+1})\Big] + O(\tau^p)$$

$$\text{with} \quad p = \begin{cases} 3 & \theta = \dfrac{1}{2} \qquad \text{„trapezoidal rule"} \\[2mm] 2 & 0 \le \theta \le 1,\ \theta \ne \dfrac{1}{2} \end{cases}$$

By rearranging (insert F, bring all $u(\,.\,,t^{k+1})$ to the left side) we get

$$\underbrace{\frac{\tau}{h^2}}$$

$$-\theta\gamma u(x_{i-1},t^{k+1}) + (1+2\theta\gamma)u(x_i,t^{k+1}) - \theta\gamma u(x_{i+1},t^{k+1}) =$$
$$= (1-\theta)\gamma u(x_{i-1},t^k) + \big(1-2(1-\theta)\gamma\big)u(x_i,t^k) + (1-\theta)\gamma u(x_{i+1},t^k) \quad i=1,\dots,N-1 \quad (28)$$
$$+\tau\Big[(1-\theta)f(x_i,t^k) + \theta f(x_i,t^{k+1})\Big] + O\big(\tau h^2 + \tau^p\big).$$
$$\uparrow$$
$$\text{as } \tau\cdot F!$$

with the abbreviation $\frac{\tau}{h^2}=\gamma$.

For each discrete time t^k we obtain the grid function $u_h^k\colon \bar\Omega_h \to \mathbb{R}$ by neglect of the error term in (28) and insertion of the boundary and initial conditions:

$$-\theta\gamma u_h^{k+1}(x_{i-1}) + (1+2\theta\gamma)u_h^{k+1}(x_i) - \theta\gamma u_h^{k+1}(x_{i+1})$$
$$= (1-\theta)\gamma u_h^k(x_{i-1}) + \big(1-2(1-\theta)\gamma\big)u_h^k(x_i) + (1-\theta)\gamma u_h^k(x_{i+1})$$
$$+\tau\Big[(1-\theta)f(x_i,t^k) + \theta f(x_i,t^{k+1})\Big] \qquad i=1,\dots,N-1,\ k\geq 0 \quad (29a)$$

$$u_h^{k+1}(x_i) = g(x_i,t^{k+1}) \qquad\qquad\qquad\qquad i=0,N,\ k\geq 0 \qquad\qquad (29b)$$
$$u_h^0(x_i) = u_0(x_i) \qquad\qquad\qquad\qquad\qquad i=1,\dots,N-1. \qquad\qquad (29c)$$

Remark 7.3. This system has the following properties:

1) (29a)/(29b) is a recursion for the grid function at time t^k.

2) In each time step a linear equation system

$$L_h u_h^{k+1} = M_h u_h^k + \tau f_h^k$$

has to be solved.

3) L_h is diagonal if $\theta = 0$ and tridiagonal else.

where L_h, M_h, f_h^k have the form:

$$L_h = \begin{pmatrix} 1 & 0 & 0 & & & \\ -\theta\gamma & 1+2\theta\gamma & -\theta\gamma & & & \\ 0 & -\theta\gamma & 1+2\theta\gamma & -\theta\gamma & & \\ & & \ddots & \ddots & \ddots & \\ & & & -\theta\gamma & 1+2\theta\gamma & -\theta\gamma \\ & & & 0 & 0 & 1 \end{pmatrix} \qquad (30)$$
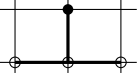
$$M_h = \begin{pmatrix} 0 & 0 & 0 & & & \\ (1-\theta)\gamma & 1-2(1-\theta)\gamma & (1-\theta)\gamma & & & \\ 0 & (1-\theta)\gamma & 1-2(1-\theta)\gamma & (1-\theta)\gamma & & \\ & & \ddots & \ddots & \ddots & \\ & & & (1-\theta)\gamma & 1-2(1-\theta)\gamma & (1-\theta)\gamma \\ & & & 0 & 0 & 0 \end{pmatrix} \qquad (31)$$

$$f_h^k = \begin{pmatrix} \frac{1}{\tau}g(x_0, t^{k+1}) \\ (1-\theta)f(x_1, t^k) + \theta f(x_1, t^{k+1}) \\ \vdots \\ (1-\theta)f(x_{N-1}, t^k) + \theta f(x_{N-1}, t^{k+1}) \\ \frac{1}{\tau}g(x_N, t^{k+1}) \end{pmatrix} \tag{32}$$

Definition 7.4 (Designation of the Standard Methods)**.**

$\theta = \mathbf{0}$ is called explicit Euler method.

As $L_h = I$ the values u_h^{k+1} can be calculated from u_h^k directly without solution of a linear equation system.

$\theta = \mathbf{1}$ is called implicit Euler method.

L_h is tridiagnoal (for one space dimension).
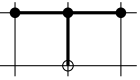
$\theta = \frac{1}{2}$ is called Crank-Nicolson method. It corresponds to the trapezoidal rule for ordinary differential equations.

It also requires the solution of a linear equation system with a tridiagonal matrix, but the precision in the time direction is higher (see below).
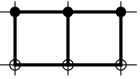
7.4 Error Analysis

For the error analysis we need the restriction operator R_h which picks the values at the finite difference nodes from the set of steady function on $\bar\Omega$:

$$R_h : \qquad C^0(\bar\Omega) \to \mathbb{R}^{N+1} \tag{33}$$
$$(R_h u)_i \;\; = \;\; u(x_i) \tag{34}$$

We can then define the error at time t^k:

$$e_h^k \;\; = \;\; \underbrace{R_h}_{\substack{\text{restriction} \\ \text{operator}}} \underbrace{u(\,.\,,t^k)}_{\substack{\text{exact solution} \\ \text{of (26)} \\ \text{at time } t^k}} - \underbrace{u_h^k}_{\substack{\text{solution} \\ \text{generated by} \\ \text{the FD scheme}}}$$

For u_h^{k+1} the equation

$$L_h u_h^{k+1} = M_h u_h^k + \tau f_h^k.$$

holds. We *define* z_h^{k+1} by the equation

$$L_h z_h^{k+1} = M_h \underbrace{R_h u(\,.\,,t^k)}_{\substack{\text{exact values} \\ \text{at last time step}}} + \tau f_h^k$$

For the errors in z_h^{k+1} (after one step with the exact values) we get:

$$L_h\left(R_h u(\,.\,,t^{k+1}) - z_h^{k+1}\right) = L_h R_h u(\,.\,,t^{k+1}) - L_h z_h^{k+1}$$

$$= \underbrace{L_h R_h u(\,.\,,t^{k+1}) - M_h R_h u(\,.\,,t^k) - \tau f_h^k}$$

<div align="center">this is the exact solution inserted</div>

<div align="center">in the differential equation (29). This is also (28)</div>

<div align="center">apart from the error term!</div>

$$=: \eta_h^k \qquad \text{„local truncation error“}$$

(35)

From (28) we get

$$\|\eta_h^k\|_\infty = O(\tau h^2 + \tau^p) \quad \text{with} \quad p = \begin{cases} 2 & 0 \leq \theta \leq 1,\ \theta \neq \dfrac{1}{2} \\[2mm] 3 & \theta = \dfrac{1}{2}. \end{cases}$$

Application of L_h to the global error e_h^{k+1} yields

$$L_h e_h^{k+1} = L_h\left(R_h u(\,.\,,t^{k+1}) - u_h^{k+1}\right)$$

$$= \underbrace{L_h R_h u(\,.\,,t^{k+1})}_{(35)} - \underbrace{L_h u_h^{k+1}}_{\text{Rem. 7.3}}$$

$$= \overbrace{M_h R_h u(\,.\,,t^k) + \tau f_h^k + \eta_h^k} - \overbrace{M_h u_h^k - \tau f_h^k}$$

$$= M_h \underbrace{\left(R_h u(\,.\,,t^k) - u_h^k\right)}_{e_h^k} + \eta_h^k$$

therefore:

$$\boxed{L_h e_h^{k+1} = M_h e_h^k + \eta_h^k} \qquad \text{recursion equation for the error}$$

This equation has the same structure as the evolution equation for u_h^{k+1}. The source term is the "local truncation error" (the error done in one step). If we solve for e_h^{k+1} we get

$$e_h^{k+1} = L_h^{-1} M_h e_h^k + L_h^{-1} \eta_h^k$$

Which can be analysed in different norms. If we apply the maximum norm $\|.\|_\infty$ we get:

$$\|e_h^{k+1}\|_\infty \leq \|L_h^{-1} M_h\|_\infty \|e_h^k\|_\infty + \|L_h^{-1}\|_\infty \|\eta_h^k\|_\infty$$

(36)

One can show:

1. $\|L_h^{-1} M_h\|_\infty \leq 1 \|L_h^{-1}\|_\infty \leq 1$ } provided $\gamma \leq \begin{cases} \infty & \text{if} \quad \theta = 1 \\ 1 & \text{if} \quad \theta = 1/2 \quad \text{Stability} \\ 1/2 & \text{if} \quad \theta = 0 \end{cases}$

2. $\|\eta_h^k\|_\infty \leq \tau O(h^2 + \tau^\beta) \quad \beta = \begin{cases} 1 & \theta \neq 1/2 \\ 2 & \theta = 1/2 \end{cases}$ Consistency

In total this gives the estimate:

$$\|e_h^k\|_\infty \le \underbrace{\|e_h^0\|_\infty}_{\text{error in initial cond. e.g. roundoff}} + \begin{cases} O(h^2 + \tau) & \theta = 0, \theta = 1 \\ O(h^2 + \tau^2) & \theta = \dfrac{1}{2} \end{cases} \tag{37}$$

It is also possible to analyse the scheme in the Euclidean norm. One obtains than

$$\theta = 1, \theta = \frac{1}{2} \qquad \text{stable in the } \|.\|_2 \text{ norm } \textit{without} \text{ time step limit}$$

$$\theta = 0 \qquad \text{again demands } \tau \le \frac{1}{2}h^2.$$

Order of convergence is the same as above. In total we get the following result:

$\theta = 1$ **(impl. Euler)**
    absolutely stable in $\|.\|_\infty$ and $\|.\|_2$
    order of convergence $O(h^2 + \tau)$
    always fulfills maximum principle (= stability in $\|.\|_\infty$)

$\theta = \frac{1}{2}$ **(Crank-Nicolson)**
    stable in $\|.\|_\infty$ for $\tau \le h^2$, absolutely stable in $\|.\|_2$-norm.
    maximum principle only fulfilled if time step condition is kept.
    order of convergence $O(h^2 + \tau^2)$

$\theta = 0$ **(expl. Euler)**
    stable in $\|.\|_\infty$ and $\|.\|_2$ only if $\tau \le \frac{h^2}{2}$
    order of convergence $O(h^2 + \tau)$

Remark 7.5. Due to the smoothing property of parabolic equations the solution initially will change quickly with time. With advancing time (with suitable boundary conditions and right side) only the long wave contributions are remaining which change slower with time.

Therefore one would like to have a small time step at the beginning of the simulation which increases with advancing time. This is prevented by the condition $\tau \le ch^2$ for the explicit Euler scheme (and for the Crank-Nicolson scheme if the maximum principle is to be observed). The explicit scheme is therefore not well suited for parabolic problems.

The spatially discretised parabolic equation (27) yields a stiff system of ordinary differential equations. The ratio of largest and smallest eigenvalue increases with $O(h^{-2})$ (it is a discretised elliptic operator). Therefore absolutely stable time discretisation schemes are necessary. $\qquad \square$

7.5 Time Step Condition for the Heat Transport Equation

The timestep condition $\tau \le ch^2$ seems to have problems with the dimensions of the contributions. However, for the problem including a coefficient a:

$$\begin{aligned} \frac{\partial u}{\partial t} - a\frac{\partial^2 u}{\partial x^2} &= f \qquad \text{in } \Omega \times T, \qquad \Omega = (0,1), T = (0, T_{end}) \\ u &= g \qquad \text{on } \partial\Omega \\ u &= u_0 \qquad \text{for } t = 0. \end{aligned} \tag{38}$$

we get the condition $a\tau \leq ch^2$ as a has the dimension L^2/T this results correctly in a dimensionless number.

To bring the heat transport equation in the same form, we have to assume that λ and C_tot are constant over the domain and there is no convective heat transport. Than we can divide by the heat capacity and get

$$\frac{\partial T(\vec{x})}{\partial t} - \frac{\lambda}{C_\text{tot}}\Delta T(\vec{x}) = -r_h(\vec{x}) \tag{39}$$

Thus we get the time step condition $\tau \leq \frac{cC_\text{tot}h^2}{\lambda}$. The quotient $\frac{\lambda}{C_\text{tot}}$ can be interpreted as thermal diffusivity.

For a realistic heat capacity of a saturated soil of 2.88 MJ m^{-3} K^{-1} and a heat conductivity of 1.5 W m^{-1} K^{-1} we get a thermal diffusivity of $5.2 \cdot 10^{-7}$ m^2 s^{-1}.

7.6 Numerical Comparison of the Time Discretisation Schemes

We solve (24) with $\Delta t = \gamma \cdot h^2$ and the initial condition

At time t = 2*h*h, gamma = 0.5 (four steps)
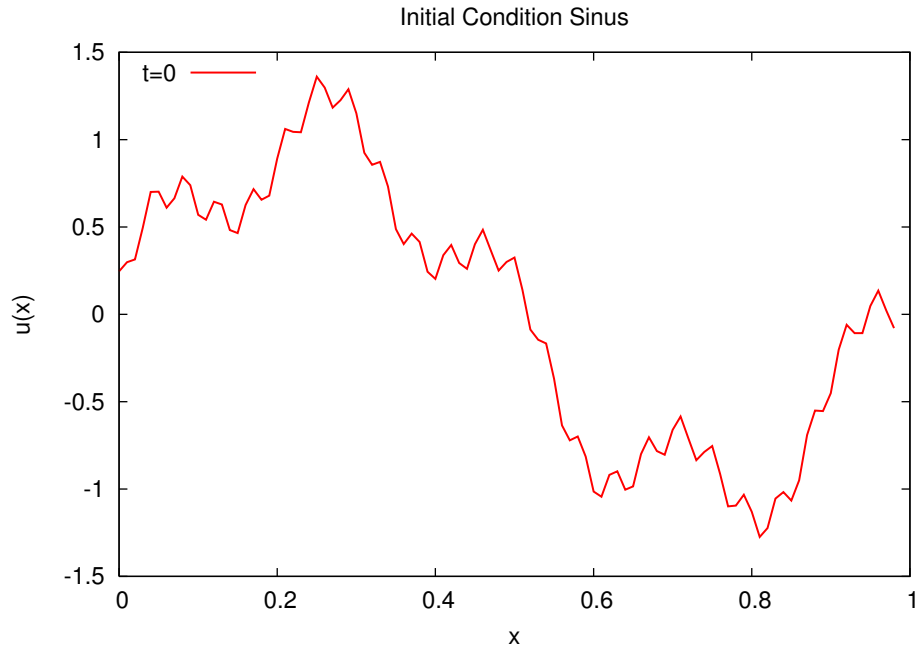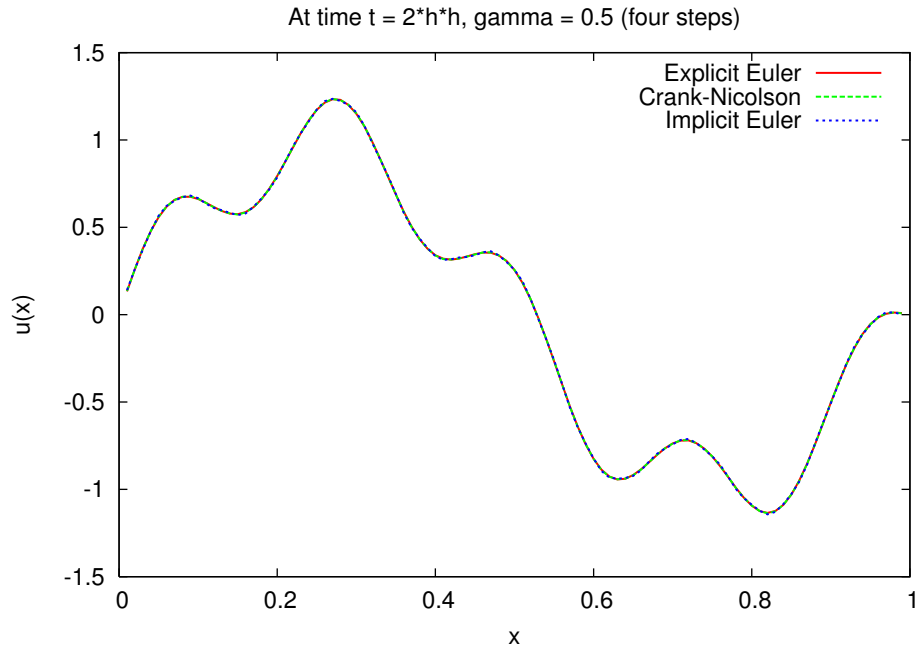
First look at the solution after four timesteps using $\gamma = 1/2$ and the three schemes for $\theta = 0, 1/2, 1$.

At time t = 2*h*h, gamma = 2 (one step)
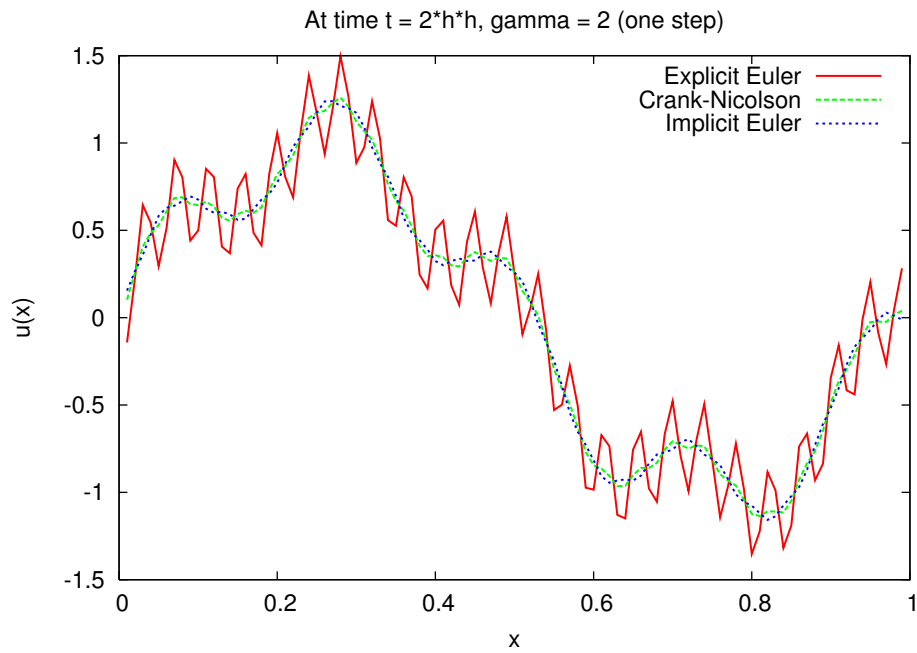
Now we use $\gamma = 2$ and perform one timestep. The stability criterion is violated for $\theta = 0, 1/2$.
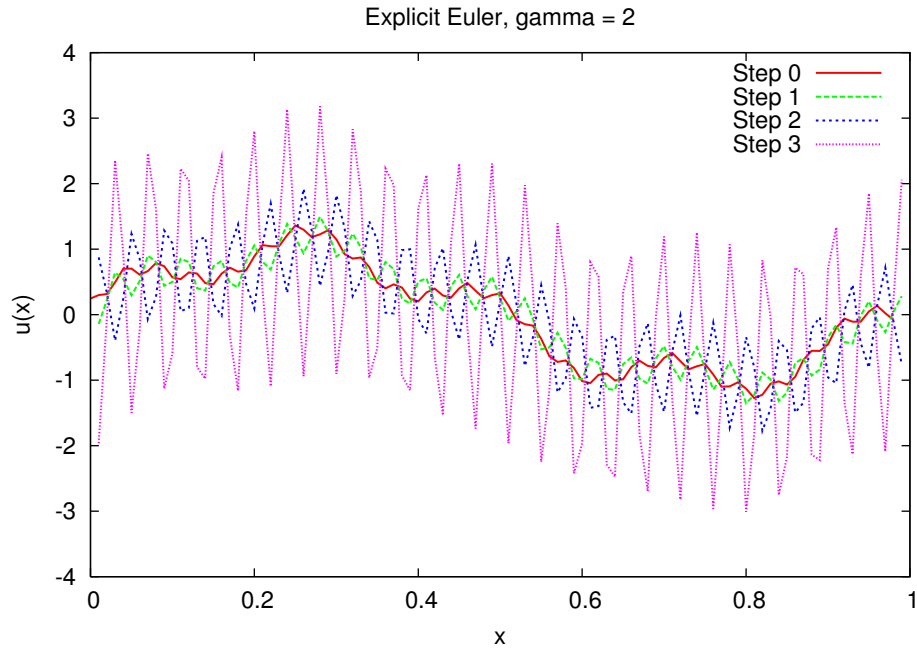
The explicit Euler scheme is unstable for $\gamma = 2$.

$\gamma = 1/2$ and 200 timesteps.
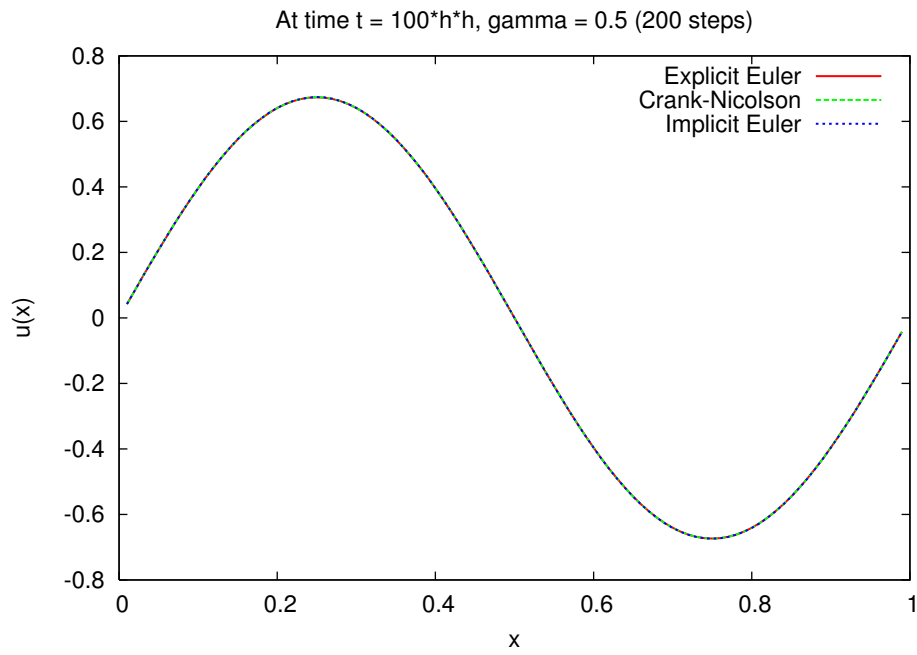
At time t = 100*h*h, gamma = 2 (50 steps)
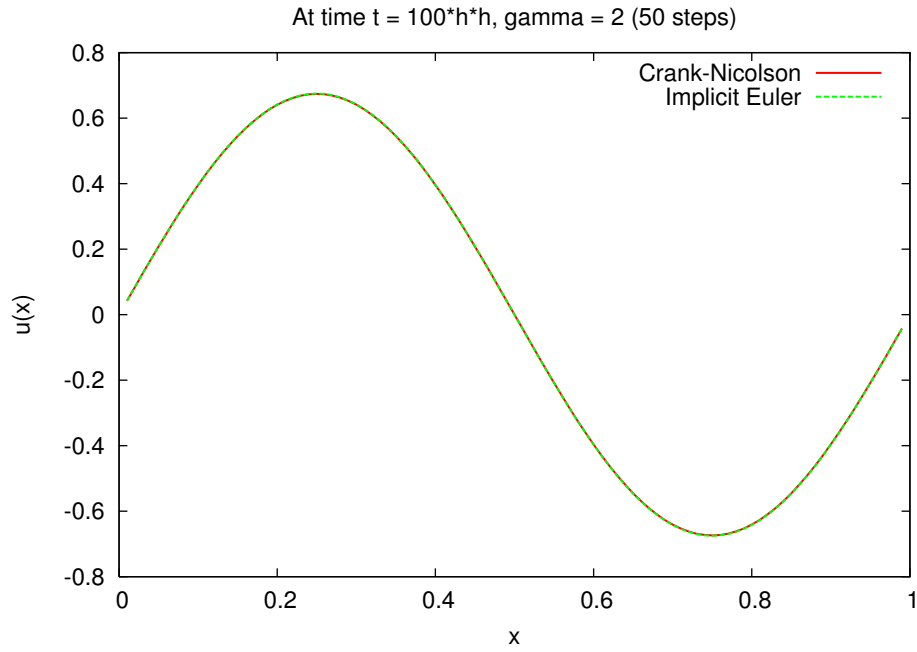
$\gamma = 2$ and 50 timesteps: The Crank-Nicolson scheme $\theta = 1/2$ seems to be stable.

Initial Condition Step
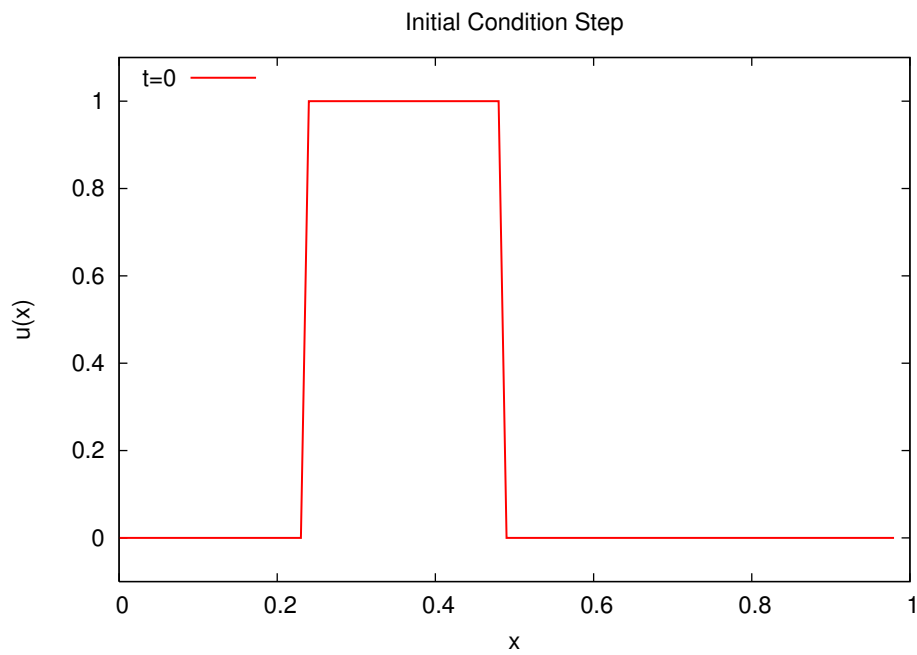
Now we test this non-smooth initial condition
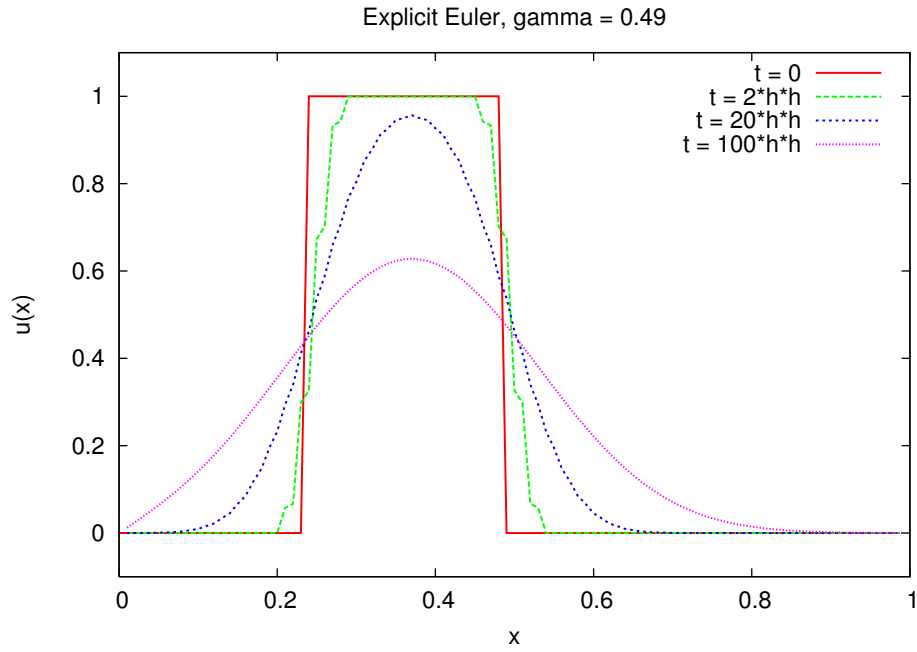
Explicit Euler scheme with $\gamma = 49/100$: stable.
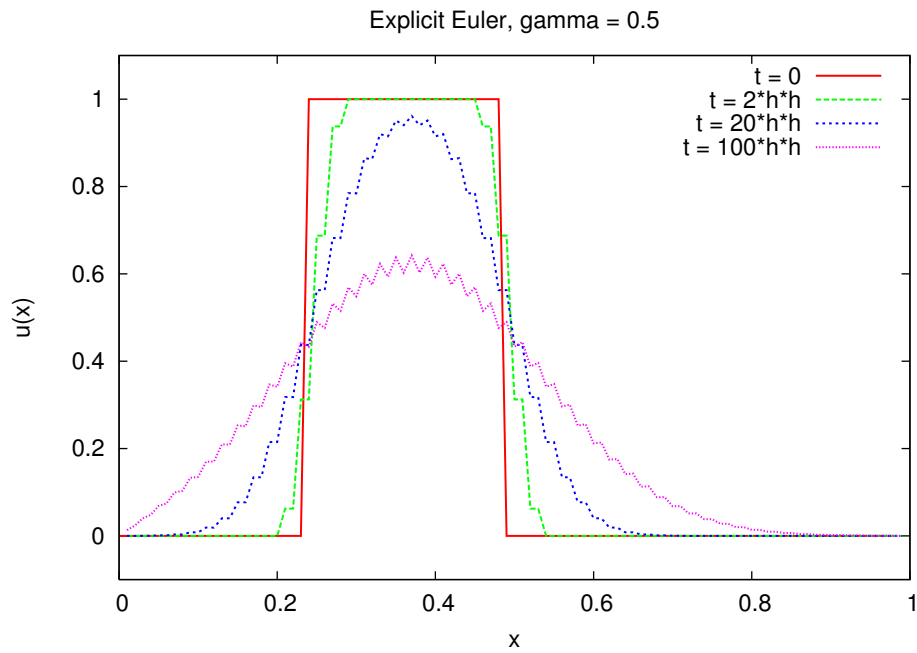
Explicit Euler scheme with $\gamma = 1/2$: this is the limit.
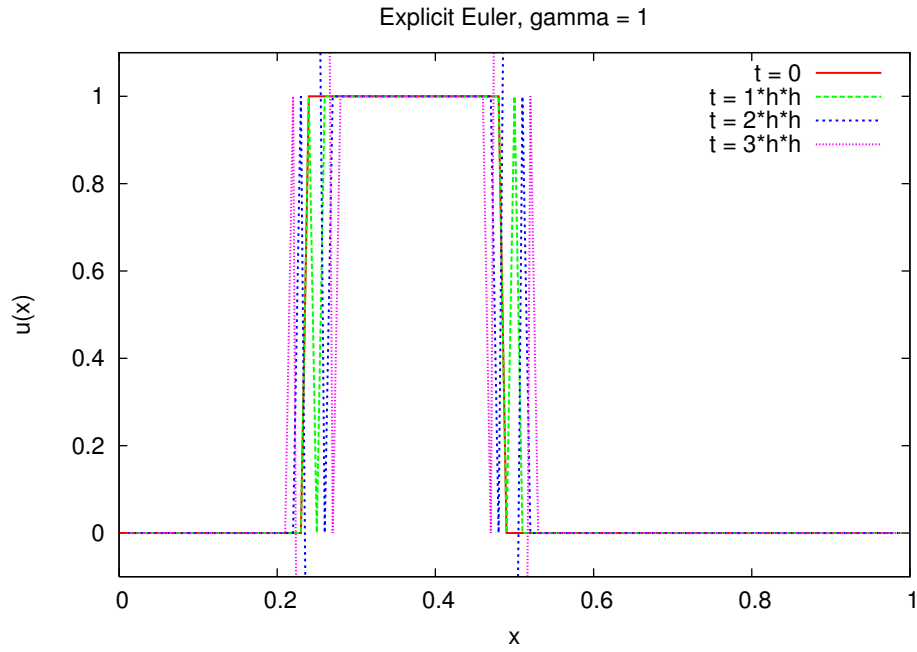
Explicit Euler scheme with $\gamma = 1$: unstable.
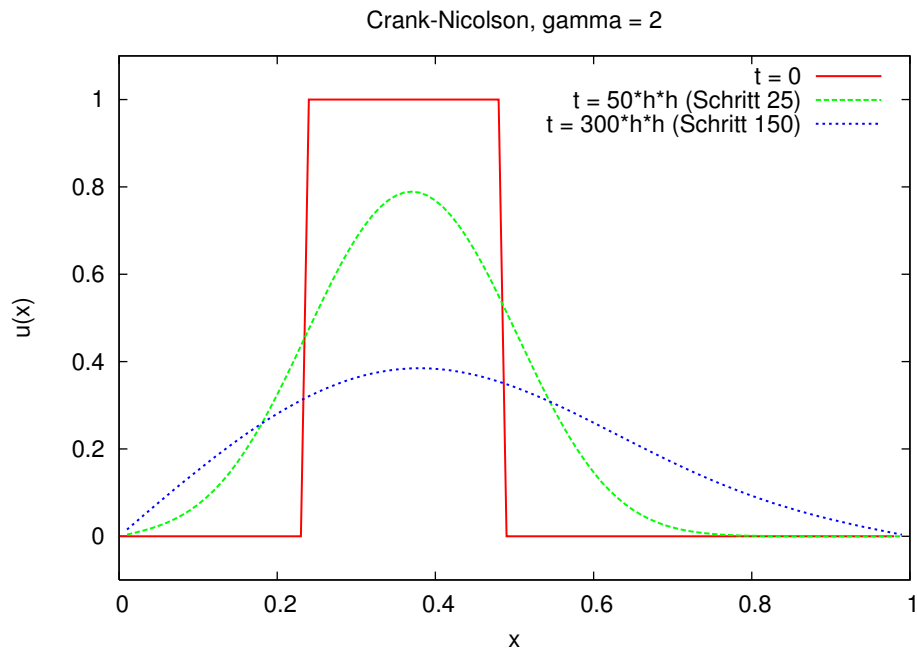
Crank-Nicolson with $\gamma = 2$: looks fine.
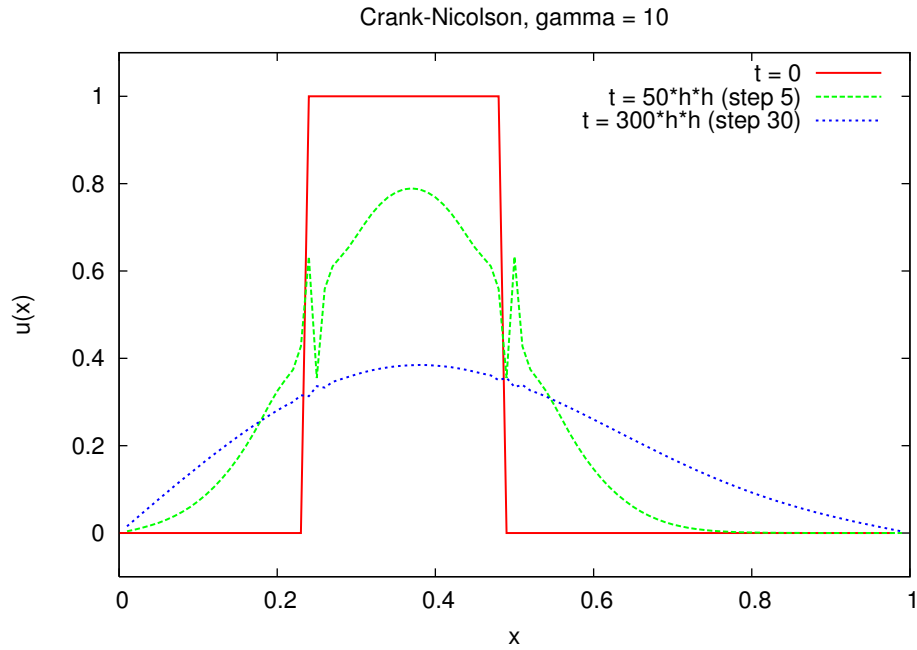
... also for $\gamma = 10$ (except for the strange jags).
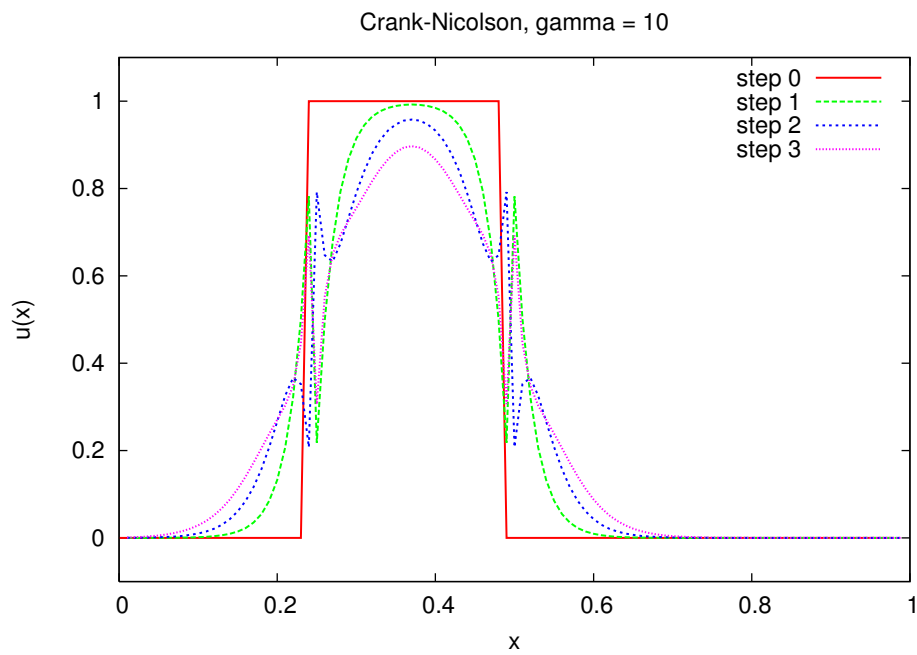
Crank-Nicolson for $\gamma = 10$: Non-physical behaviour around the jump in the initial condition for the first time steps.

Implicit Euler, gamma = 2

t = 0
t = 50*h*h (Schritt 25)
t = 300*h*h (Schritt 150)
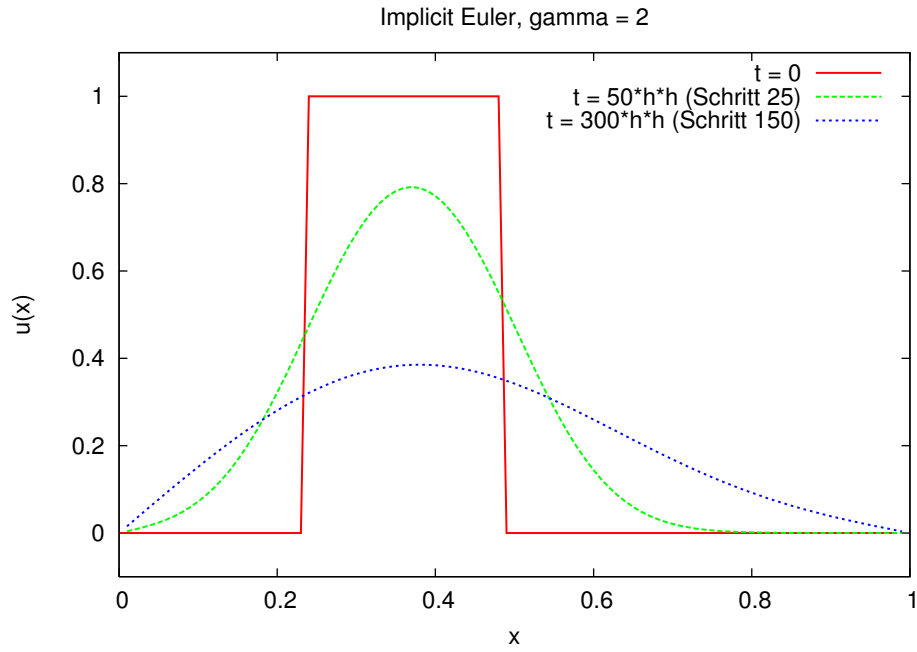
Implicit Euler scheme with $\gamma = 2$: stable.

Implicit Euler, gamma = 10

t = 0
t = 50*h*h (step 5)
t = 300*h*h (step 30)
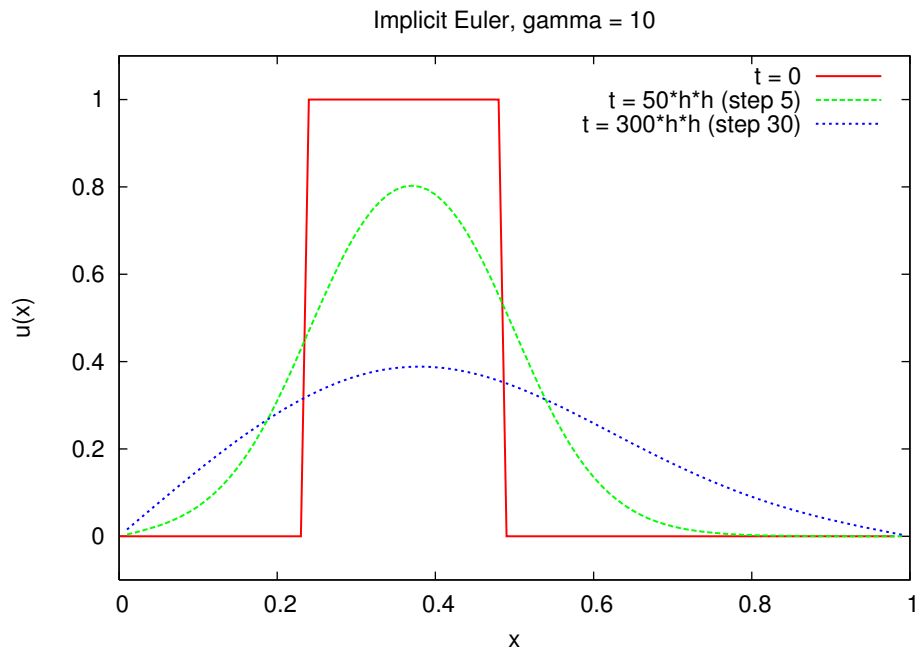
...and for $\gamma = 10$: stable as well.

Implicit Euler, gamma = 10
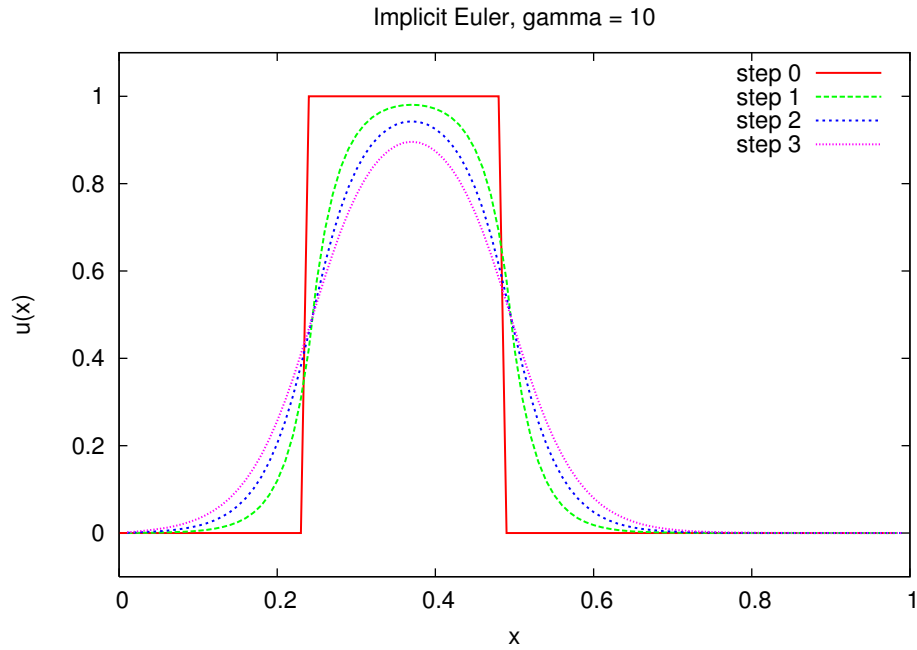
There is no non-physical behaviour in the first time steps for the implicit Euler scheme.

Vergleich IE/CN, gamma = 10
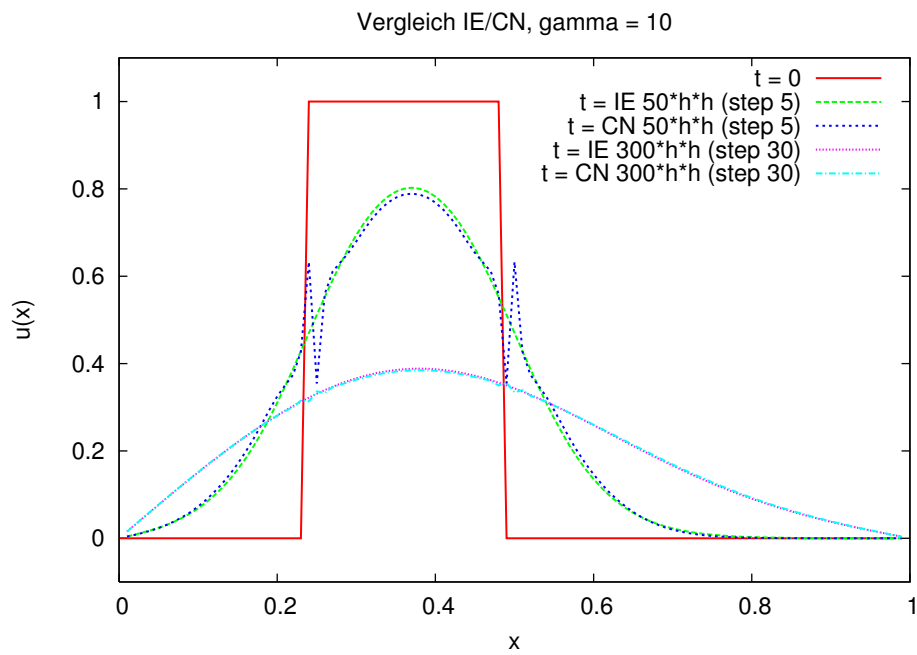
But: the Crank-Nicolson scheme has a asymptotically better convergence rate in time.

7.7 Summary

- The solutions of parabolic equations are getting smoother over time.

- In the method of lines the PDE is first discretised in space yielding a system of ordinary differential equations, which is discretised in time.

- Absolutely stable (and thus implicit) methods are used for time discretisation as they are better suited for stiff systems. A very small time step is needed with explicit schemes if the spatial resolution is high.

8 Hyperbolic PDEs - Solute Transport

8.1 Solute Transport in Porous Media

8.1.1 Flux Law

Solutes are transported in a saturated porous media either by convection of the liquid phase or by diffusion. This processes can be described by:

$$\vec{J}_s = \vec{J}_{s_{conv}} + \vec{J}_{s_{diff}} \tag{40}$$

where

$$\vec{J}_{s_{conv}} = c_s \cdot \vec{J}_w \tag{41}$$
$$\vec{J}_{s_{diff}} = -D_s(\theta_w) \cdot \nabla c_s \tag{42}$$

with:

c_s	solute concentration	[mol m^{-3}]
$\vec{J}_w$	volumetric water flux	[m s^{-1}]
$D_s(\theta_w)$	dispersion coefficient	[m^2 s^{-1}]

8.1.2 Solute Dispersion

Molecular Diffusion

Just as in free liquid molecular diffusion of solutes occurs in porous media. However, the diffusion is hindered by the solid matrix and in unsaturated porous media by the geometry of the water phase.

There are different models for this reduction of solute diffusion. Two popular parameterisations are the models of Millington and Quirk [Mil59]:

$$D_{s_{eff}} = \frac{\theta_w^{10/3}}{\Phi^2} D_{s_{molecular}} \tag{43}$$

and [MQ61]

$$D_{s_{eff}} = \frac{\theta_w^2}{\Phi^{2/3}} D_{s_{molecular}} \tag{44}$$

The diffusion coefficient of Cl$^-$ in water is $2.03 \cdot 10^{-9}$ m^2 s^{-1}

Dispersion

The combination of molecular diffusion, diffusive mixing and convective mixing leads to a larger macroscopic dispersion coefficient. This coefficient is a tensor, which is symmetric with the main directions parallel (longitudinal) to and perpendicular (transversal) to the water flow. According to [Bea61] and [Sch61] for the case of pure hydromechanic dispersion its components are

$$D_{s_{ij}} = [\lambda_l - \lambda_t] \frac{v_{w_i} v_{w_j}}{||\vec{v}_w||_2} + \lambda_t ||\vec{v}_w||_2 \delta_{ij} \tag{45}$$

where

λ_l	longitudinal dispersion coefficient	[m]
λ_t	transversal dispersion coefficient	[m]
$\vec{v}_w = \frac{\vec{J}_w}{\theta_w}$	water velocity	[m s^{-1}]

8.1.3 Convection-Dispersion Equation

$$\frac{\partial (\theta_w c_s(\vec{x}))}{\partial t} + \nabla \cdot \vec{J}_s(\vec{x}) + r_s(\vec{x}) = 0 \tag{46}$$

$$\frac{\partial (\theta_w c_s(\vec{x}))}{\partial t} - \nabla \cdot \left(\bar{D}(\vec{x}, \theta_w) \nabla c_s(\vec{x}) \right) + \nabla \cdot \left(c_s \vec{J}_w(\vec{x}) \right) + r_s(\vec{x}) = 0 \tag{47}$$

or if we divide by a homogeneous θ_w:

$$\frac{\partial c_s(\vec{x})}{\partial t} - \nabla \cdot \left(\frac{\bar{D}(\vec{x}, \theta_w)}{\theta_w} \nabla c_s(\vec{x}) \right) + \nabla \cdot (c_s \vec{v}_w(\vec{x})) + \frac{1}{\theta_w} r_s(\vec{x}) = 0 \tag{48}$$

The convection-dispersion equation is a parabolic equation as for homogeneous dispersion coefficient and water flux density in one dimension:

$$-\frac{D}{\theta_w} \frac{\partial^2 c_s(x)}{\partial x^2} + v_w(x) \frac{\partial c_s(x)}{\partial x} + \frac{\partial c_s(x)}{\partial t} + \frac{1}{\theta_w} r_s(x) = 0 \tag{49}$$

$$\det \begin{pmatrix} -\frac{D}{\theta_w} & 0 \\ 0 & 0 \end{pmatrix} = 0 \tag{50}$$

and

$$\text{Rank} \begin{bmatrix} -\frac{D}{\theta_w} & 0 & v_w(x) \\ 0 & 0 & 1 \end{bmatrix} = 2 \tag{51}$$

8.1.4 Effective Hyperbolicity of the Convection-Dispersion Equation

Mathematically the convection-dispersion equation will always be a parabolic equation. However, in its discretised form, the equation can get convection dominated. The distance covered by a diffusive process is $\sqrt{Dt}$, while the distance covered by a convective process is vt. The times to travel the distance h (the grid size) are then $t_D = h^2/2D$ and $t_C = h/v$. The process is convection dominated if $t_D > t_C$. This results in the condition

$$\frac{h^2}{2D} > \frac{h}{v} \Leftrightarrow \frac{hv}{2D} > 1$$

.

From the analysis of the matrix for a Finite-Differences discretisation one can also derive the condition $\frac{hv}{2D} > 1$.

This happens more often for solute transport than for heat transport as the diffusion coefficient for e.g. Cl^- in water is $2 \cdot 10^{-9}$ m^2 s^{-1} which leads in combination with the model of Millington Quirk for a porosity of 33 % to a diffusion coefficient of $4 \cdot 10^{-10}$ m^2 s^{-1} whereas the heat diffusion coefficient is in the order of $5 \cdot 10^{-7}$ m^2 s^{-1}.

8.2 Method of Characteristics

If we start with the multidimensional, hyperbolic, linear transport equation

$$
\begin{aligned}
\frac{\partial u}{\partial t} + \nabla \cdot (\vec{v}u) &= f \quad \text{in } \Omega \times T \\
u &= g \quad \text{on } \Gamma_{in} = \left\{ (x,t) \in \partial\Omega \times T \mid \vec{v}(\vec{x}) \cdot \vec{n}(\vec{x}) < 0 \right\} \\
&\qquad\qquad\qquad\qquad\qquad\qquad\uparrow \\
&\qquad\qquad\qquad\qquad\qquad \text{outer normal}
\end{aligned}
\tag{52}
$$

$$
u = u_0 \quad \text{for } t = 0
$$

for a given velocity field $\vec{v} \colon \Omega \times T \to \mathbb{R}^d$.

we get under the assumptions $\nabla \cdot \vec{v} = 0$ (source/sink free flux field) and $f = 0$:

$$
\frac{\partial u}{\partial t} + \vec{v} \cdot \nabla u + \underbrace{(\nabla \cdot \vec{v})}_{=0} u = 0
$$

$$
\iff \frac{\partial u}{\partial t} + \vec{v} \cdot \nabla u = 0
$$

This is called the „non-conservative" form of the hyperbolic equation.

Let $\big(\hat{x}(s), \hat{t}(s)\big)$ be a curve in $\Omega \times T$ parameterised with s.

Calculate the derivative of u in the direction of the curve

$$
\frac{\mathrm{d}}{\mathrm{d}s}\Big[u\big(\hat{x}(s), \hat{t}(s)\big)\Big] = \sum_{i=1}^{d} \left.\frac{\partial u}{\partial x_i}\right|_{(\hat{x}(s),\hat{t}(s))} \cdot \left.\frac{\partial \hat{x}_i}{\partial s}\right|_s + \left.\frac{\partial u}{\partial t}\right|_{(\hat{x}(s),\hat{t}(s))} \cdot \left.\frac{\partial \hat{t}}{\partial s}\right|_s
\tag{53}
$$
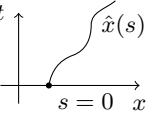
Up to now the curve was arbitrary. Now we choose:

$$
\begin{aligned}
\left.\frac{\mathrm{d}\hat{t}}{\mathrm{d}s}\right|_s &= 1, & \hat{t}(0) &= t_0 \\
\left.\frac{\mathrm{d}\hat{x}_i}{\mathrm{d}s}\right|_s &= v_i\big(\hat{x}(s), \hat{t}(s)\big), & \hat{x}_i(0) &= x_{0,i}
\end{aligned}
\tag{54}
$$

This is a system of ordinary differential equations for the curve which is only determined by the data of the differential equation.

Evaluation of the derivative along this special curve yields:

$$
\frac{\mathrm{d}}{\mathrm{d}s}\Big[u\big(\hat{x}(s), \hat{t}(s)\big)\Big] = \underbrace{\nabla u\big(\hat{x}(s), \hat{t}(s)\big) \cdot \vec{v}\big(\hat{x}(s), \hat{t}(s)\big) + \frac{\partial u\big(\hat{x}(s), \hat{t}(s)\big)}{\partial t} \cdot 1}_{\text{this is the PDE}} = 0
$$

Conclusion: Along the „Charakteristic" (54) the solution of u is constant.

Γ_{in}
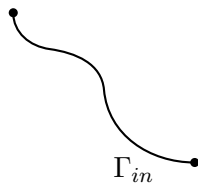
Example 8.1. We use $\Omega = \mathbb{R}$, i.e. no boundary condition, only initial condition, and $\vec{v} = a = const > 0$.

Charakteristic:

$$\frac{d\hat{t}(s)}{ds} = 1; \qquad \hat{t}(0) = 0 \qquad \Rightarrow \boxed{\hat{t}(s) = s} \quad \begin{array}{l}\text{choose } \hat{t} \text{ as inde-}\\ \text{pendent variable}\end{array}$$

$1D!$

$$\frac{d\hat{x}(s)}{ds} = a; \qquad \hat{x}(0) = x_0 \qquad \Rightarrow \boxed{\hat{x} = x_0 + a \cdot \hat{t}} \quad (a > 0!)$$

slope a in (t, a)

x_0 \qquad $\hat{x}$
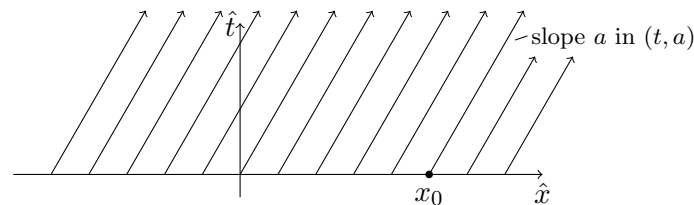
How can $u(x, t)$ be determined?

„Backtracking" of the Charakteristic: Determine (x, t) to $x_0(x, t)$ such that

$$x = \underbrace{x_0(x, t)}_{\text{unknown}} + a \cdot t$$

$$\Longleftrightarrow x_0(x, t) = x - a \cdot t$$

and

$$\boxed{u(x, t) = u_0(x - a \cdot t)} \qquad \text{„Displacement of the function } u_0 \text{ to the right" } (a > 0).$$

This also works for discontinuous initial conditions!

$$u_0(x) = \begin{cases} 1 & x \geq 0 \\ 0 & \text{sonst} \end{cases}$$

The jump moves with the velocity a to the right.

$$u(x, t) = \begin{cases} 1 & x \geq a \cdot t \\ 0 & \text{sonst} \end{cases}$$

102

Generally (with boundary, multidimensional): Define the „tracking operator" $\Phi(x,t,t') \in \bar{\Omega}$ with:

i. e. $\Phi(x,t,t')$ traces the point (x,t) up to the time t' and then yields the new position.

$$\text{Solve (54) for } t_0 = t, \quad x_0 = x.$$
$$\text{Set } \Phi(x,t,t') = \hat{x}(s^*), \text{ such that } \hat{t}(s^*) = t'.$$
$$u(x,t) = \begin{cases} u_0\big(\Phi(x,t,0)\big) & \text{if } \Phi(x,t,0) \in \bar{\Omega} \\ g(t^*) & \text{for } \Phi(x,t,t^*) \in \partial\Omega \end{cases}$$

8.3 Finite Differences for linear hyperbolic PDEs

We continue to analyse the multidimensional, hyperbolic, linear transport equation 52 for a given velocity field $\vec{v}$.

As before we limit ourselves to the spatially one-dimensional case with constant velocity $a > 0$:

$$\frac{\partial u}{\partial t} + \frac{\partial(au)}{\partial x} = 0 \qquad \text{in } (0,1) \times (0,\infty)$$
$$u(0,t) = g(t) \tag{55}$$
$$u(x,0) = u_0(x)$$

Same Ansatz as for parabolic equations: Method of lines

The fully dicretized version (second order in space (central difference quotient), one-step θ method in time) is:

$$\frac{u_h^{k+1}(x_i) - u_h^k(x_i)}{\tau} + \frac{(1-\theta)a}{2h}\Big[u_h^k(x_{i+1}) - u_h^k(x_{i-1})\Big]$$
$$+ \frac{\theta a}{2h}\Big[u_h^{k+1}(x_{i+1}) - u_h^{k+1}(x_{i-1})\Big] = 0 \qquad k \geq 0,\ i = 1,\ldots,N-1$$

$$\Longleftrightarrow \quad -\frac{\tau\theta a}{2h}u_h^{k+1}(x_{i-1}) + u_h^{k+1}(x_i) + \frac{\tau\theta a}{2h}u_h^{k+1}(x_{i+1}) =$$
$$= \frac{\tau(1-\theta)a}{2h}u_h^k(x_{i-1}) + u_h^k(x_i) - \frac{\tau(1-\theta)a}{2h}u_h^k(x_{i+1})$$

The equation system has the same structure as in the parabolic case $L_h u_h^{k+1} = M_h u_h^k$.

However,

- L_h is *no* M-Matrix (pos. sign) if $\theta > 0$.

- L_h is not symmetric

- L_h diagonally dominant if $2 \cdot \frac{\tau\theta a}{2h} < 1$

$$\text{therefore} \quad \theta = 0 \qquad \text{and } \tau, h, a \text{ arbitrary, obvious: } L_h = I$$
$$\theta \neq 0 \qquad \text{and } \tau < \frac{h}{\theta a}.$$

- Remark: How do we handle the right boundary if $a > 0$? We can not give boundary conditions at outflow boundaries...

$\theta = 0$, explicit case

$$u_h^{k+1} = M_h u_h^k \qquad \text{with } M_h = \text{tridiag}\left(-\frac{\tau a}{2h}, \ 1, \ \frac{\tau a}{2h}\right)$$

$$\Rightarrow \|M_h\|_\infty = 1 + \frac{\tau|a|}{h} > 1 \text{ for all } \tau, h$$

$$\Rightarrow \text{Method is unconditionally instable in the maximum norm}$$

$\theta = 1$, fully implicit case

$$\text{no M-Matrix!}$$

$$L_h u_h^{k+1} = u_h^k \text{ with } L_h = \text{tridiag}\left(\frac{\tau a}{2h}, \quad \overset{\swarrow}{1}, \ -\frac{\tau a}{2h}\right)$$

Numerical results show, that the method is stable for $\frac{\tau}{h} \geq C(a)$, i.e. if τ is *large enough (!)*, but not in the maximum norm.

Alternative: take one-sided difference quotient in space (which one?).

$$\frac{u_h^{k+1}(x_i) - u_h^k(x_i)}{\tau} + \frac{(1-\theta)a}{h}\left[u_h^k(x_i) - u_h^k(x_{i-1})\right] + \frac{\theta a}{h}\left[u_h^{k+1}(x_i) - u_h^{k+1}(x_{i-1})\right] = 0$$

$$\iff -\frac{\tau\theta a}{h}u_h^{k+1}(x_{i-1}) + \left(1 + \frac{\tau\theta a}{h}\right)u_h^{k+1}(x_i) = \frac{\tau(1-\theta)a}{h}u_h(x_{i-1}) + \left(1 - \frac{\tau(1-\theta)a}{h}\right)u_h(x_i)$$

again $L_h u_h^{k+1} = M_h u_h^k$

L_h is a M-Matrix, if $a \geq 0$. If $a < 0$ one chooses the other one-sided difference quotient.

$$\frac{\partial u}{\partial x}(x_i, t) = \frac{u(x_{i+1}, t) - u(x_i, t)}{h} + O(h)$$

and again gets a M-Matrix!

Thus the choice of the difference quotient depends on the sign of a.

- L_h is unsymmetric, but bi-diagonal.

- There is no boundary condition at the right boundary as in the continuous case!

$\theta = 0$, explicit case

$$\text{Diagonal}$$
$$\downarrow$$

$$u_h^{k+1} = M_h u_h^k \text{ with } M_h = \text{bidiag}\left(\frac{\tau a}{h}, 1 - \frac{\tau a}{h}\right)$$

$$\|M_h\|_\infty = \left|\frac{\tau a}{h}\right| + \left|1 - \frac{\tau a}{h}\right| = 1, \text{ if } 0 \leq \frac{\tau a}{h} \leq 1$$

$$\frac{\tau a}{h} \geq 0 \quad \text{obvious}$$

$$\frac{\tau a}{h} \leq 1 \quad \text{is called CFL-condition after Courant, Friedrich, Levy (1928)}$$

graphically:

slope $a \leq \frac{h}{\tau}$ (in the x-t graph)
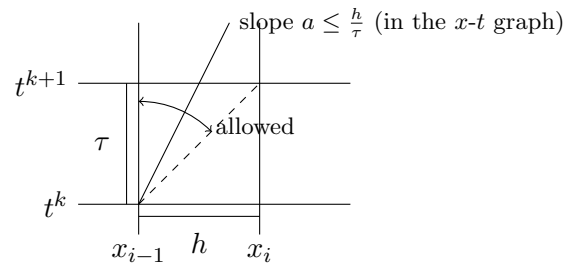
$\theta = 1$, implicit case

$$L_h u_h^{k+1} = u_h^k \qquad L_h = \text{bidiag}\left(-\frac{\tau a}{h}, 1 + \frac{\tau a}{h}\right)$$

$$\|L_h^{-1}\|_\infty \leq 1 \text{ for all } \frac{\tau}{h} \text{ as } L_h 1 \geq 1$$

method is unconditionally stable!

8.3.1 Numerical Diffusion

A closer inspection of the discretisation error yields an explanation why the one-sided differences are working well:

We analyse the one-sided difference quotient with implicit Euler scheme:

Taylor series expansion yields:

expand around $(x, t+\tau)$

$$\frac{\partial u}{\partial t}: \qquad \frac{u(x,t+\tau) - u(x,t)}{\tau} = \left.\frac{\partial u}{\partial t}\right|_{(x,t+\tau)} - \frac{\tau}{2}\left.\frac{\partial^2 u}{\partial t^2}\right|_{(x,t+\tau)} + O(\tau^2)$$

!

$$\frac{\partial u}{\partial x}: \qquad \frac{u(x,t+\tau) - u(x-h,t+\tau)}{h} = \left.\frac{\partial u}{\partial x}\right|_{(x,t+\tau)} - \frac{h}{2}\left.\frac{\partial^2 u}{\partial x^2}\right|_{(x,t+\tau)} + O(h^2)$$

For sufficiently smooth u:

$$\frac{\partial u}{\partial t} + a \cdot \frac{\partial u}{\partial x} = 0 \qquad \left\{ \begin{array}{l} \Rightarrow \quad \dfrac{\partial^2 u}{\partial t^2} + a \cdot \dfrac{\partial^2 u}{\partial x \partial t} = 0 \\[2em] \Rightarrow \quad \dfrac{\partial^2 u}{\partial t \partial x} + a \cdot \dfrac{\partial^2 u}{\partial x^2} = 0 \end{array} \right\} \qquad \frac{\partial^2 u}{\partial t^2} - a^2 \frac{\partial^2 u}{\partial x^2} = 0$$

thus

$$\boxed{\frac{\partial^2 u}{\partial t^2} = a^2 \frac{\partial^2 u}{\partial x^2}}$$

If we insert the exact solution in the difference equation we get:

$$\frac{u(x, t+\tau) - u(x, t)}{\tau} + a\frac{u(x, t+\tau) - u(x-h, t+\tau)}{h} =$$

$$= \left(\frac{\partial u}{\partial t} + a\frac{\partial u}{\partial x}\right)\bigg|_{(x,t+\tau)} - \left(\frac{\tau}{2}\frac{\partial^2 u}{\partial t^2} + \frac{ah}{2}\frac{\partial^2 u}{\partial x^2}\right)\bigg|_{(x,t+\tau)} + O(h^2 + \tau^2)$$

$$= \left(\frac{\partial u}{\partial t} + a\frac{\partial u}{\partial x}\right)\bigg|_{(x,t+\tau)} - \frac{a^2\tau + ah}{2}\frac{\partial^2 u}{\partial x^2}\bigg|_{(x,t+\tau)} + O(h^2 + \tau^2)$$

This implies:

- The leading term of the discretisation error acts as a diffusion term. Note that the sign is correct.

- The discrete method can also be interpreted as a second order exact (!) discretisation of the Convection-Dispersion equation

$$\frac{\partial u}{\partial t} + a\frac{\partial u}{\partial x} - \frac{a^2\tau + ah}{2}\frac{\partial^2 u}{\partial x^2} = 0$$

  The diffusion coefficient depends on the position.

- The central difference quotient can be stabilised by addition of an „artificial" diffusion term.

- Because of $\frac{\partial^2 u}{\partial t^2} = a^2\frac{\partial^2 u}{\partial x^2}$ the *time* discretisation error of the implicit Euler scheme can be interpreted as a diffusion term in space. This explains the stabilisation of the central difference quotient for a large enough τ (!).

- The upwind-method smears steep fronts in the solution.
  $\Rightarrow$ This is called „numerical" Diffusion.

8.3.2 Numerical Comparison

We solve (55) with $\Delta t = \gamma \cdot \frac{h}{a}$ for a smooth pulse and a rectangular blob as initial condition with $a = 1$ and $h = 1/200$.
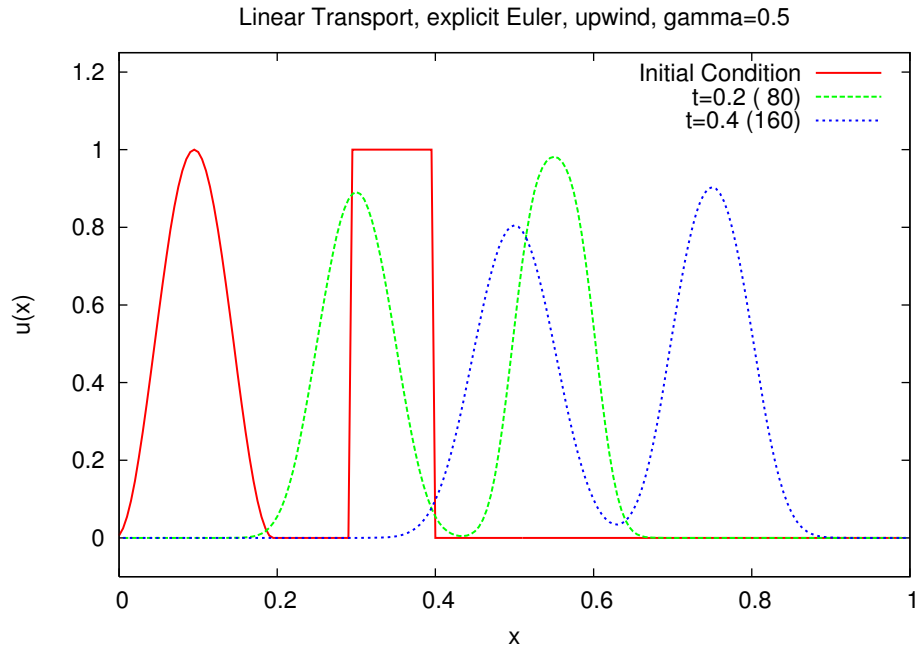
Explicit Euler, upwind with $\gamma = 1/2$.
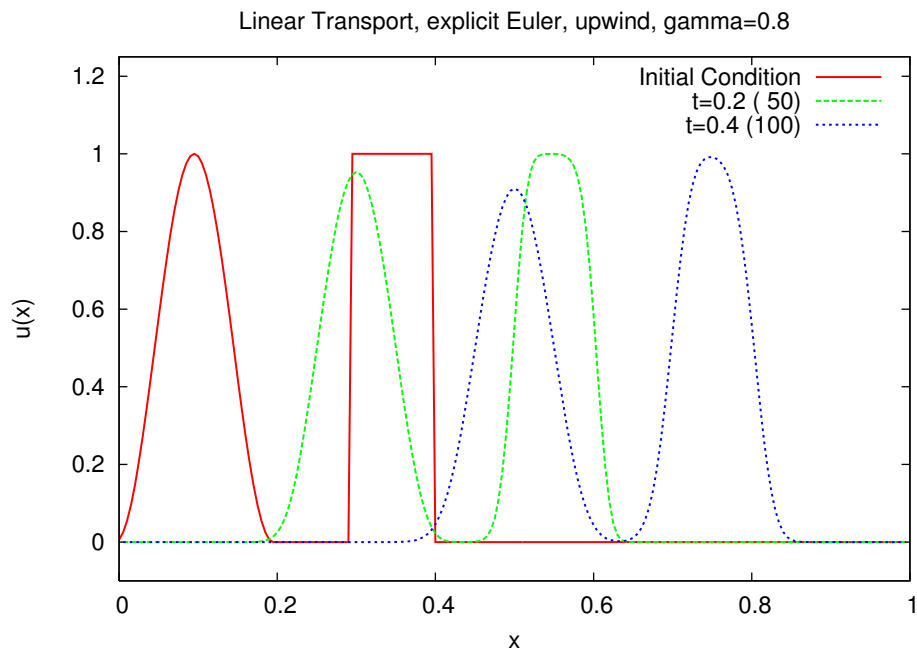
Explicit Euler, upwinding with $\gamma = 4/5$.
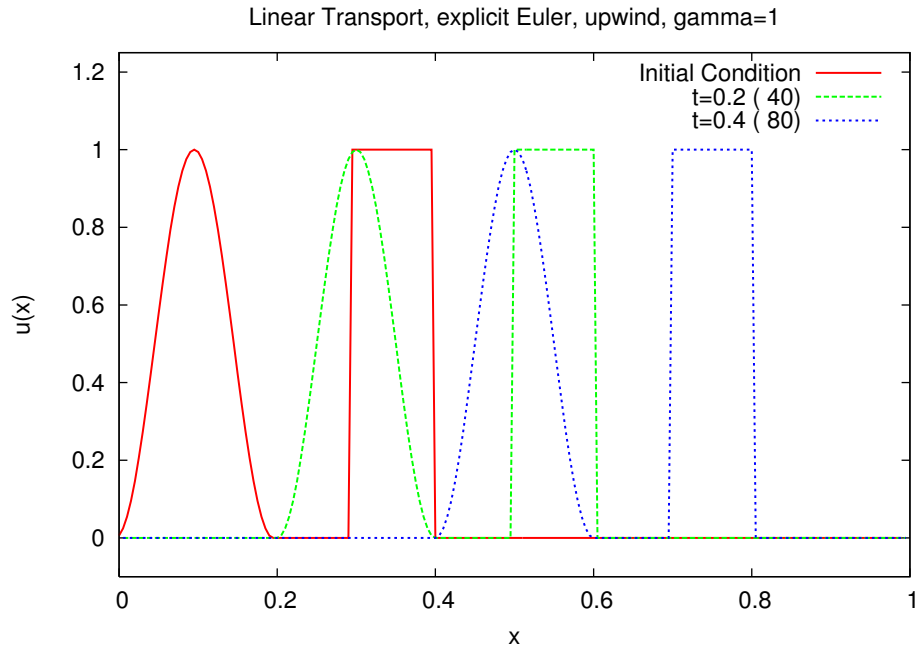
Explicit Euler, upwinding with $\gamma = 1$.
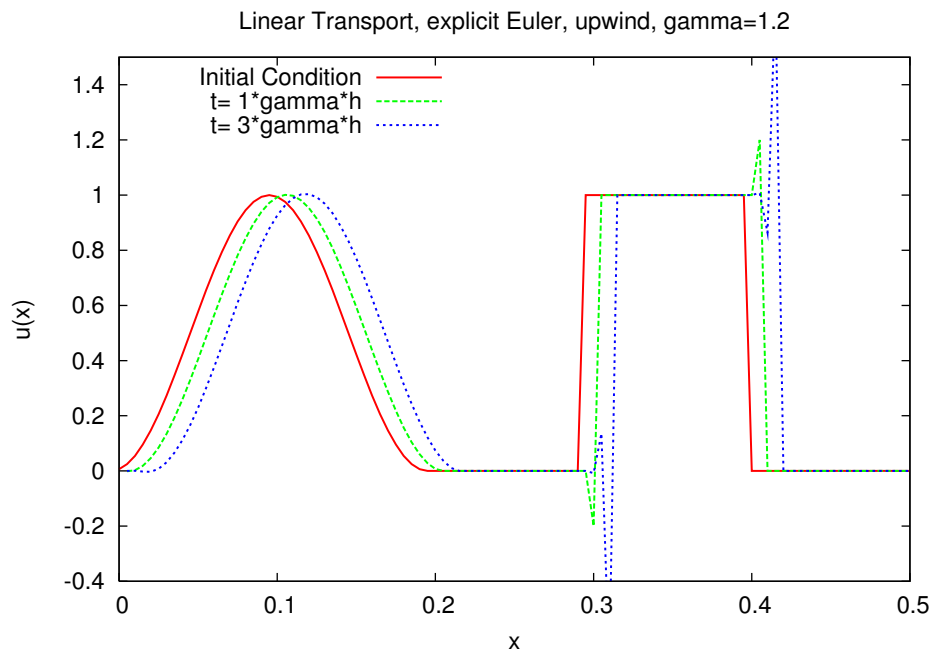
Explicit Euler, upwinding with $\gamma = 1.2$: Courant condition is strict.

Linear Transport, explicit Euler, upwind, t=0.4
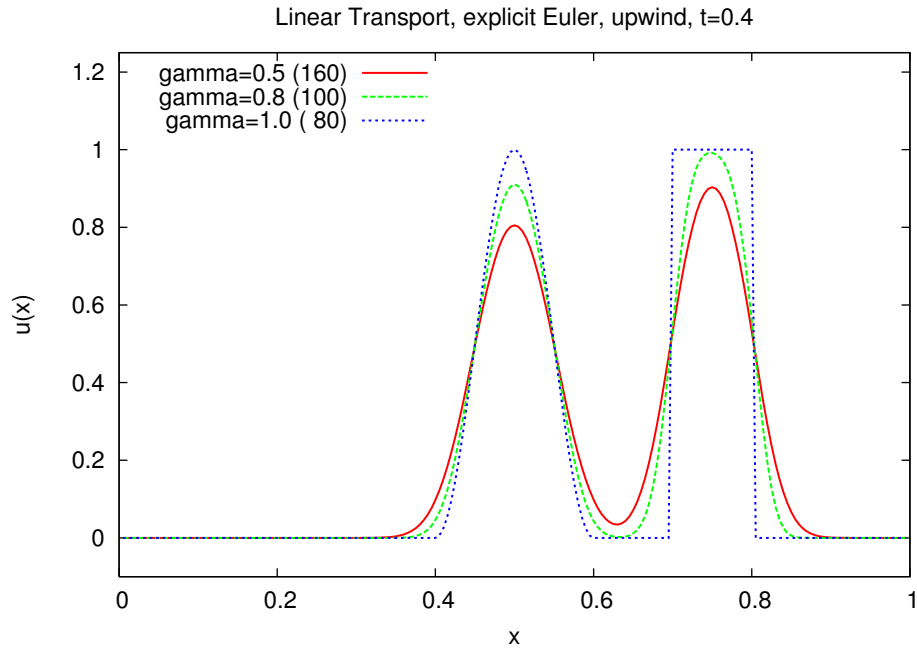
Explicit Euler, upwinding: stable for $\gamma \leq 1$, is getting better with increasing γ.

Linear Transport, implicit Euler, upwind, gamma=0.5
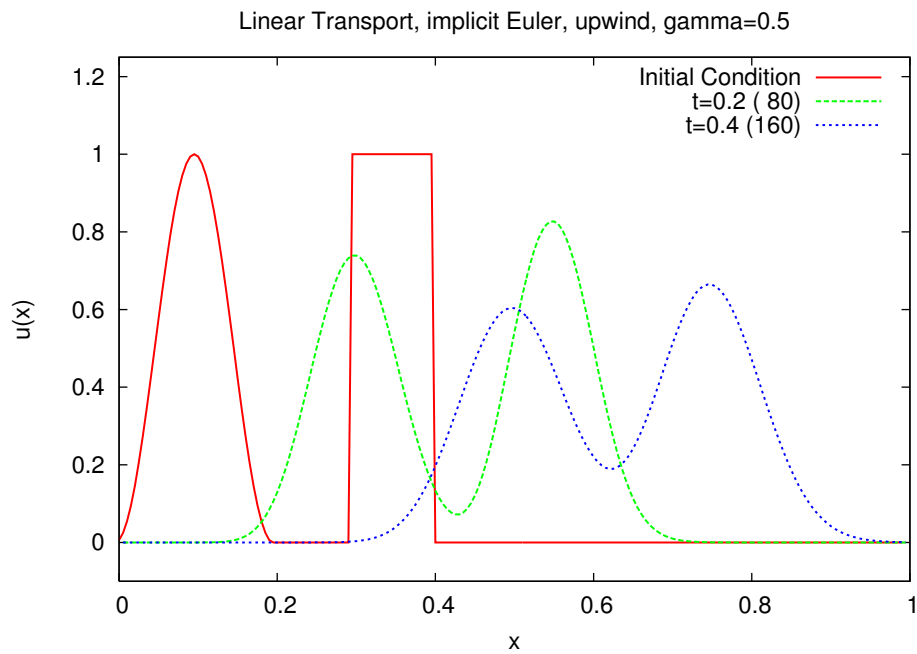
Implicit Euler, upwinding with $\gamma = 0.5$.
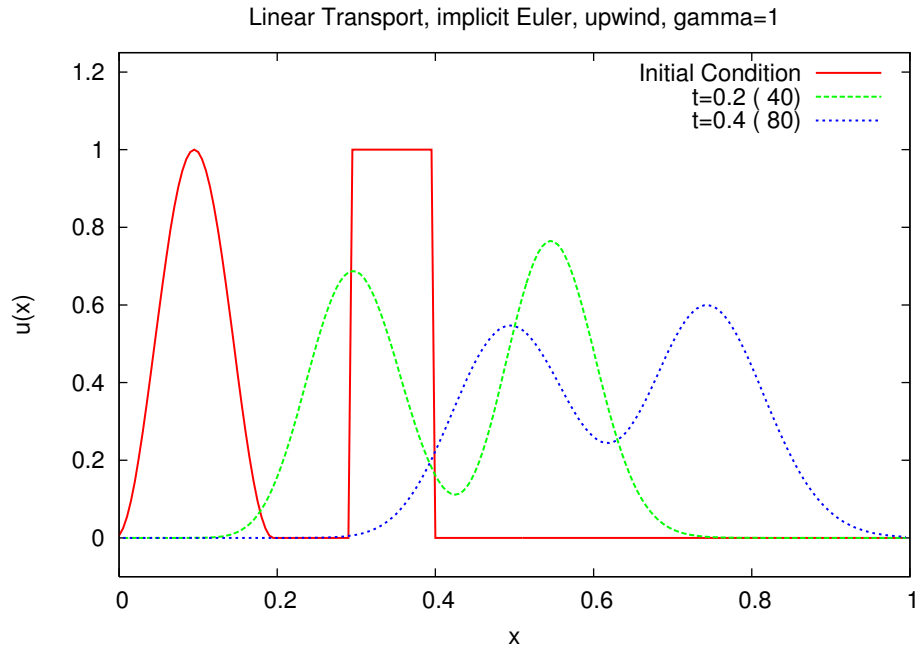
Implicit Euler, upwinding with $\gamma = 1$.

Implicit Euler, upwinding with $\gamma = 2$.

Linear Transport, implicit Euler, upwind, t=0.4
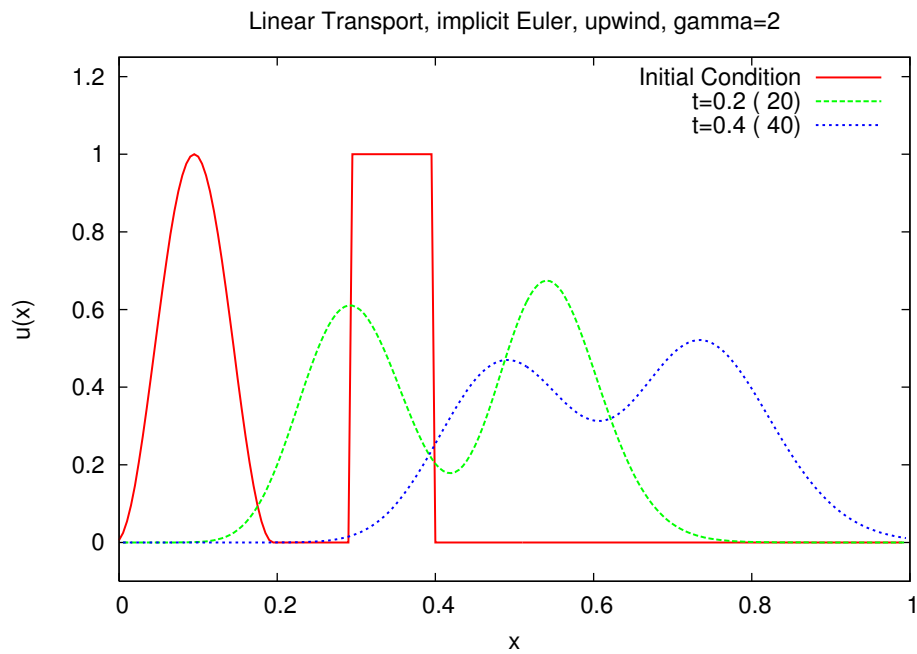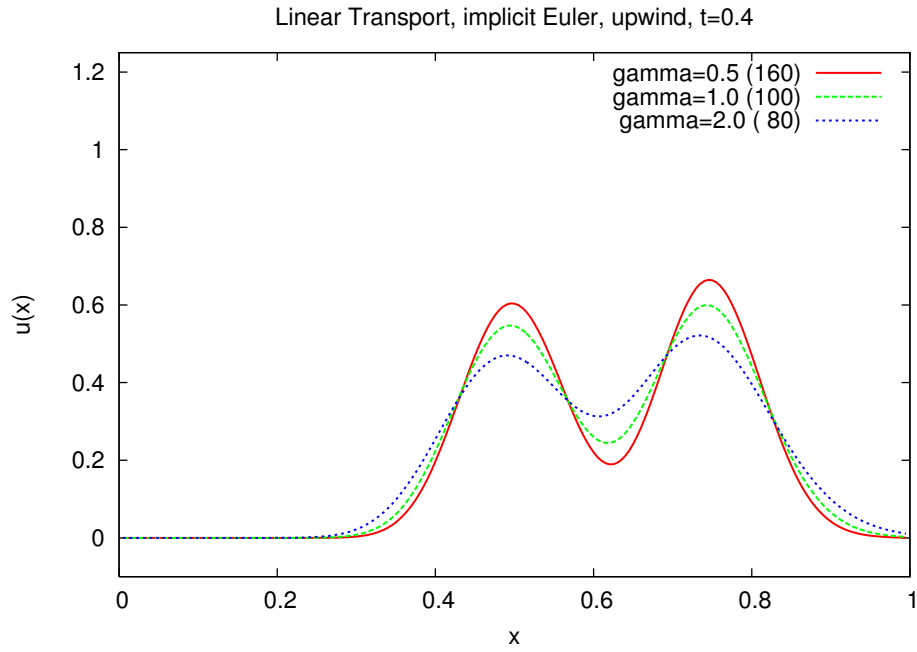
Implicit Euler, upwinding: stable for all γ but diffusive. Is getting worse with increasing γ

Linear Transport, implicit Euler, central, gamma=0.1
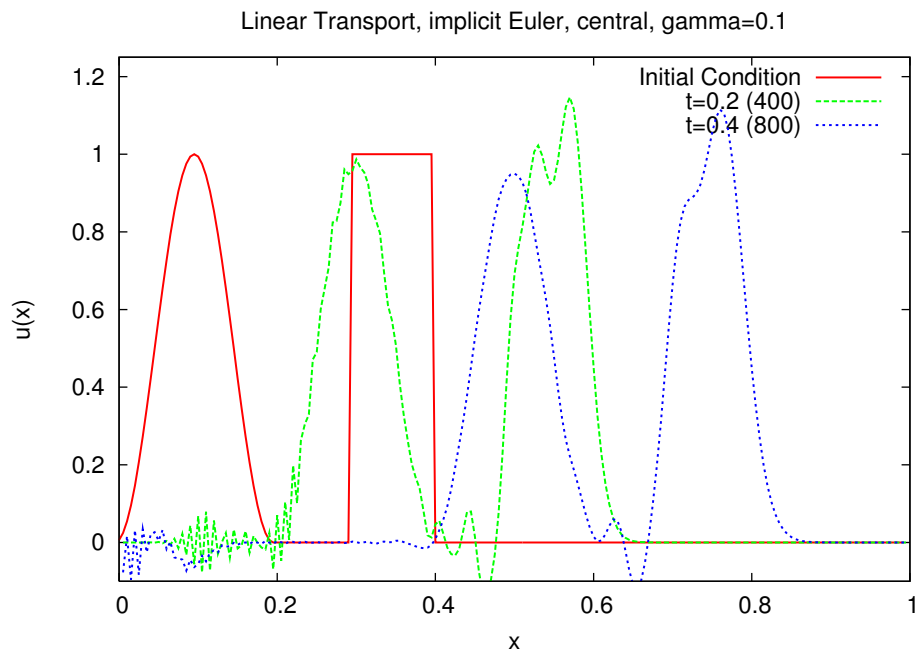
Implicit Euler, central differences with $\gamma = 0.1$.
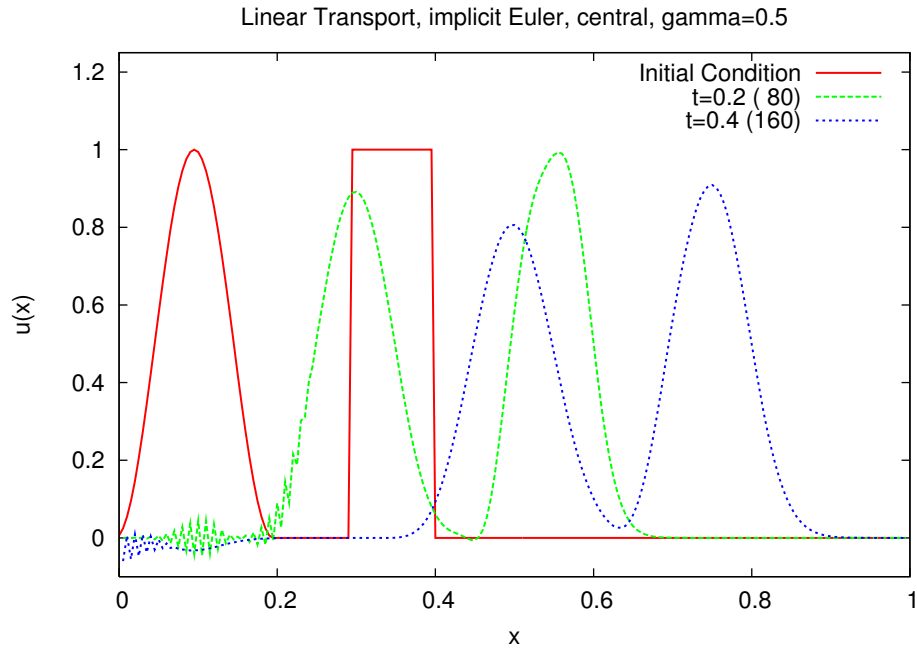
Implicit Euler, central differences with $\gamma = 0.5$.
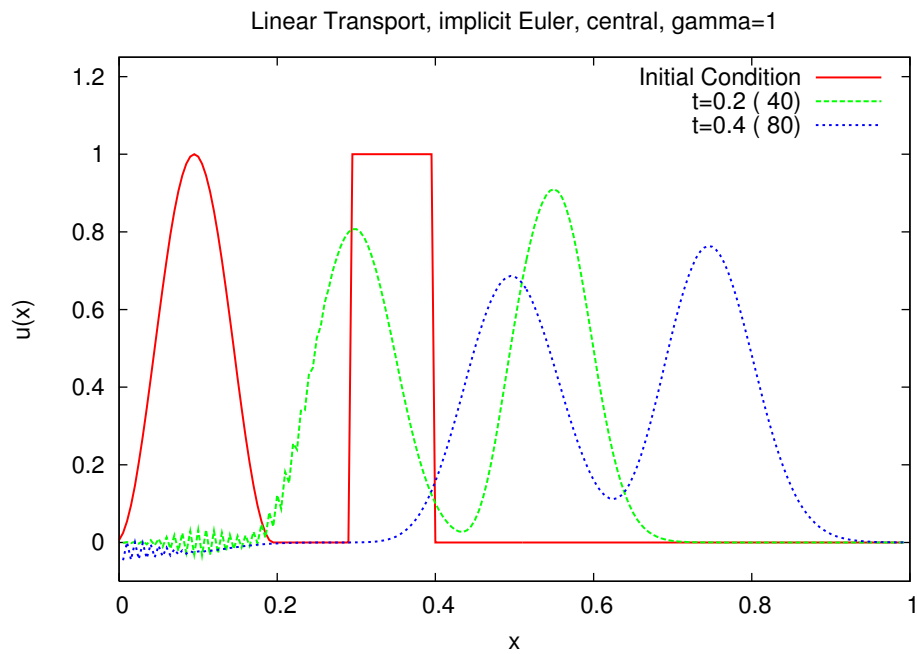
Implicit Euler, central differences with $\gamma = 1$.
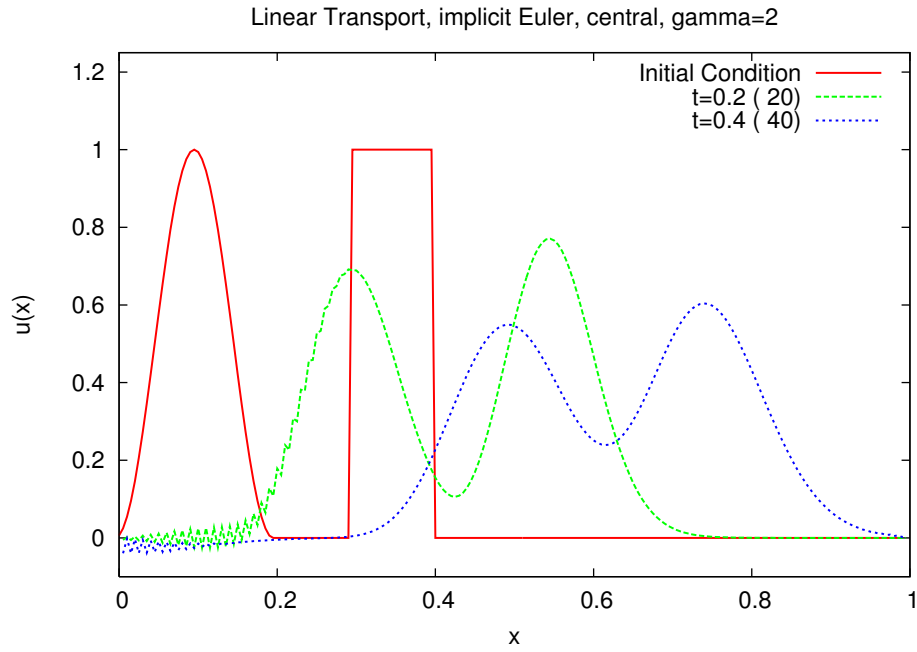
Implicit Euler, central differences with $\gamma = 2$.

Implicit Euler, central differences with $\gamma = 10$.

Linear Transport, implicit Euler, central, t=0.4
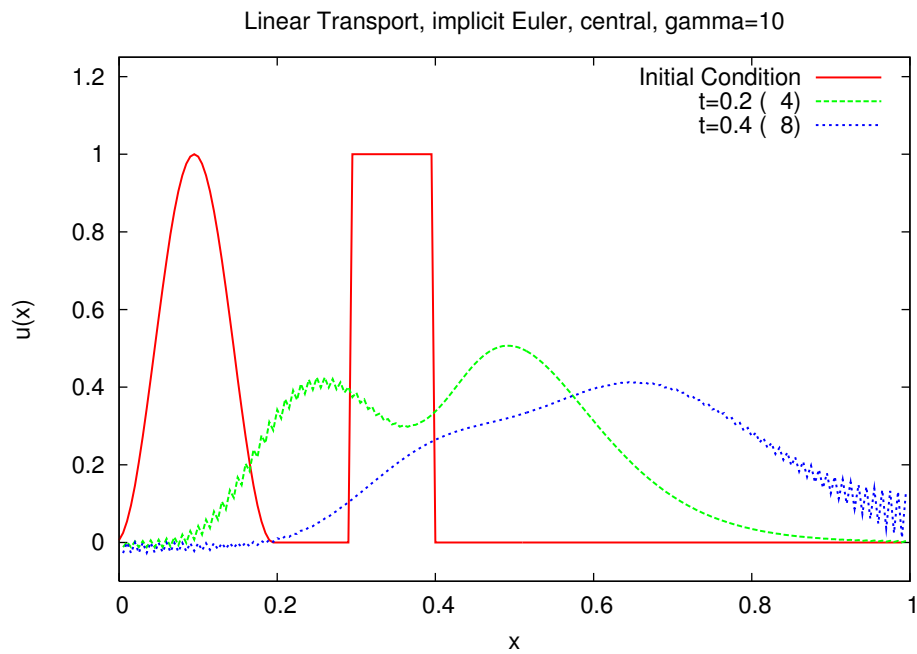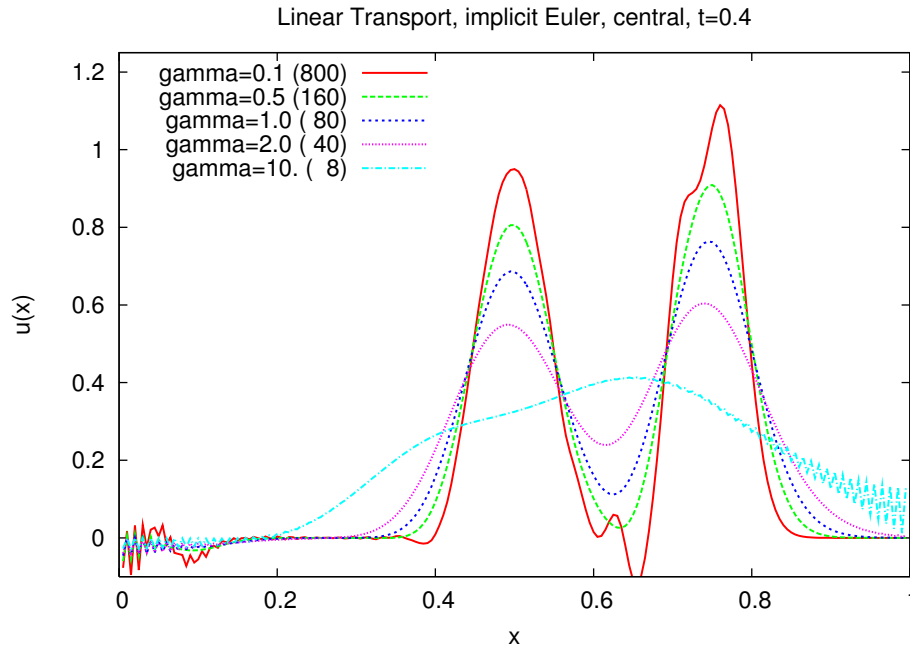
Implicit Euler, central differences: diffusive, oscillations are decreasing with increasing γ.

Linear Transport, gamma=0.5
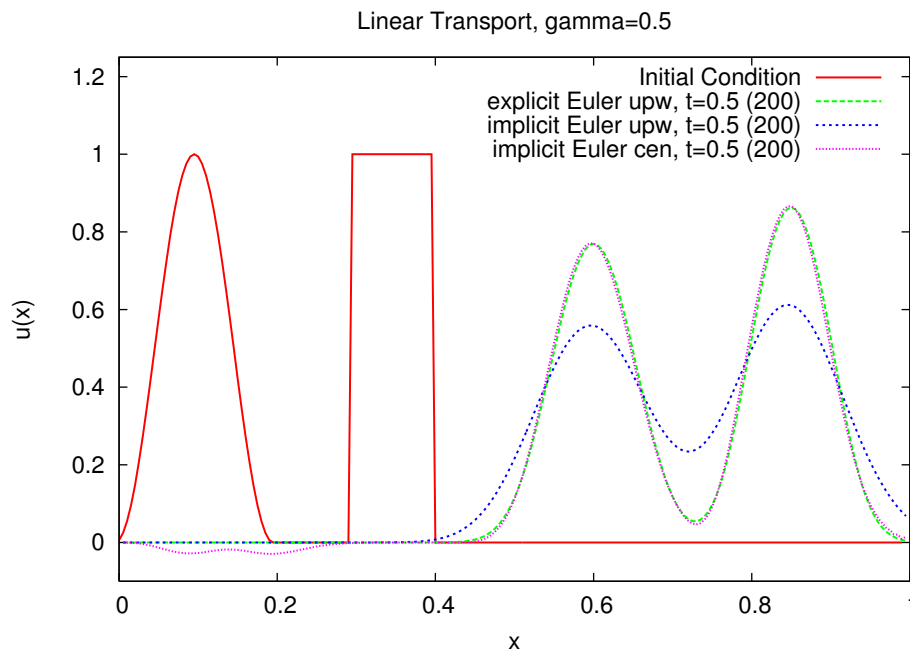
Comparison of all methods with $\gamma = 0.5$: Explicit Euler with upwinding is the method of choice.

8.4 Finite-Volume method for hyperbolic equations

The following part is oriented on [Lev02, Chap. 4].

We discretise the equation

$$\frac{\partial u}{\partial t} + \frac{\partial f(u)}{\partial x} = 0 \qquad \text{in } (\overset{x}{0}, 1) \times (\overset{t}{0}, \infty) \tag{56}$$

with suitable initial and boundary conditions with a Cell-Centred Finite-Volume scheme.

For a purely convective equation we have the flux function $f(u) = a \cdot u$ with $\mathbb{R} \ni a > 0$.

If we integrate the equation again over the grid cell ω_i

$$\int_{\omega_i} \frac{\partial u}{\partial t} \, \mathrm{d}x + \int_{g_{ij}} \frac{\partial f(u)}{\partial x} \, \mathrm{d}x = 0$$

$$\overset{\text{permutation}}{\Longleftrightarrow} \quad \frac{\mathrm{d}}{\mathrm{d}t} \overset{\frown}{\int_{\omega_i}} u(x,t) \, \mathrm{d}x + f\big(u(x_{i+\frac{1}{2}},t)\big) - f\big(u(x_{i-\frac{1}{2}},t)\big) = 0 \tag{57}$$

The (classical) solution of (56) fulfills (57) for arbitrary intervals ω (partial integration).

For a fully discretised equation we also integrate over an interval in time (t^k, t^{k+1}):

$$\underbrace{\frac{1}{h} \int_{\omega_i} u(x, t^{k+1}) \, \mathrm{d}x}_{\substack{\text{cell average at time} \\ t^{k+1}}} =$$

$$= \underbrace{\frac{1}{h} \int_{\omega_i} u(x, t^k) \, \mathrm{d}x}_{\substack{\text{cell average at} \\ \text{time } t^k}} - \frac{\tau}{h} \left[\underbrace{\frac{1}{\tau} \int_{t^k}^{t^{k+1}} f\big(u(x_{i+\frac{1}{2}},t)\big) \, \mathrm{d}t}_{\substack{\text{average flow over the} \\ \text{boundary } x_{i+\frac{1}{2}} \text{ in time} \\ \text{interval } (t^k, t^{k+1})}} - \underbrace{\frac{1}{\tau} \int_{t^k}^{t^{k+1}} f\big(u(x_{i-\frac{1}{2}},t)\big) \, \mathrm{d}t}_{\substack{\text{average flow over the} \\ \text{boundary } x_{i-\frac{1}{2}} \text{ in time} \\ \text{interval } (t^k, t^{k+1})}} \right] \tag{58}$$

This equation describes the *exact* evolution of cell averages.

Finite-Volume methods use the cell averages

$$U_i^k = \frac{1}{h} \int_{\omega_i} u(x, t^k) \, \mathrm{d}x + \quad \text{error}$$

as unknowns. The fluxes over the faces are the approximated quantities.

For an explicit scheme it is obvious to choose

$$\frac{1}{\tau} \int_{t^k}^{t^{k+1}} f\big(u(x_{\underline{\underline{i+\frac{1}{2}}}},t)\big) \, \mathrm{d}t = \underbrace{\mathcal{F}\big(U_i^k, U_{i+1}^k\big)}_{=F_{i+\frac{1}{2}}^k} + \quad \text{error} \tag{59}$$
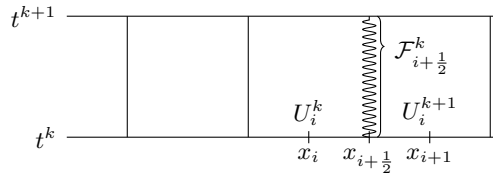
$\mathcal{F}$ is called numerical flow function.

The fully discretised method is obtained by neglect of the error terms. (58) and (59) yield:

$$U_i^{k+1} = U_i^k - \frac{\tau}{h}\big(\mathcal{F}(U_i^k, U_{i+1}^k) - \mathcal{F}(U_i^k, U_{i-1}^k)\big) \qquad (60)$$

As in the explicit Finite-Difference method U_i^{k+1} does only depend on U_{i-1}^k, U_i^k, U_{i+1}^k.

Finite-Volume methods are globally conservative:

total $\dfrac{\text{mass}}{\text{energy}}$ at time $t^{k+1} =$

$$= \sum_{i=0}^{N-1} h \cdot U_i^{k+1} = \sum_{i=0}^{N-1} h\Big(U_i^k - \frac{\tau}{h}\big(\mathcal{F}(U_i^k, U_{i+1}^k) - \mathcal{F}(U_{i-1}^k, U_i^k)\big)\Big)$$

over all cells → cell average!

mass, energy in cell i

$$= \underbrace{\sum_{i=0}^{N-1} hU_i^k}_{\substack{\text{mass at time} \\ t^k}} -\Big(\tau\mathcal{F}(U_N^k, U_{N-1}^k) + \mathcal{F}(U_{„-1“}^k, U_0^k)\Big)$$

special fluxes defined by the boundary conditions!

all internal fluxes cancel each other.

Finite-Volume methods *exactly* represent the conserved quantity. This is *not* true for Finite-Difference methods in general (i.e. with non-equidistant grids, variable coefficients, non-linearities).

8.4.1 Requirements for the flux function

The analysis of FD methods delivered the two important criteria consistency (local truncation error, local approximation) and stability (error propagation). This is the same for FV methods.

To guarantee consistency two requirements for the flux function are necessary:

I
$$\mathcal{F}(Q, Q) = f(Q)$$
for each Q!   if u constant in x and t, the flux evaluation should be constant.

II steadiness of the flux function:
$$|\mathcal{F}(Q_i, Q_{i+1}) - f(\bar{Q})| \le L\max\big(|Q_i - \bar{Q}|, |Q_{i+1} - \bar{Q}|\big).$$

The numerical flux should converge to the correct value if $Q_i, Q_{i+1} \to \bar{Q}$ converge.

For the stability of explicit schemes the CFL-condition is a necessary prerequisite (but not sufficient as the unconditionally instable method shows).

Request: Characteristic has to be contained in the numerical sphere of influence, i.e.

$$|a| \le \frac{h}{\tau} \iff \boxed{\left|\frac{a\tau}{h}\right| \le 1}$$

$\nu = \left|\dfrac{a \cdot \tau}{h}\right|$ is called *Courant number*.

116

8.4.2 Unstable Flux Function

As

$$\mathcal{F}(U_i^k, U_{i+1}^k) \approx \frac{1}{\tau} \int\limits_{t^k}^{t^{k+1}} f\big(u(x_{i+\frac{1}{2}}, t)\big) \, \mathrm{d}t$$

application of the trapezoidal rule yields

$$\mathcal{F}(U_i^k, U_{i+1}^k) = \frac{1}{2} \left[f(U_i^k) + f(U_{i+1}^k) \right]. \tag{61}$$

$\mathcal{F}$ is consistent (fulfills I and II from above).
For $f(u) = au$ we obtain the scheme

$$U_i^{k+1} = U_i^k - \frac{\tau}{h} \left(\frac{1}{2} \left[aU_i^k + aU_{i+1}^k \right] - \frac{1}{2} \left[aU_{i-1}^k + aU_i^k \right] \right)$$

$$= U_i^k - a\tau \underbrace{\frac{1}{2h} \left(U_{i+1}^k - U_{i-1}^k \right)}_{\text{central Difference}}$$

This method was found to be unconditionally unstable in (55).

8.4.3 Upwinding Method

Idea: Use knowledge about characteristics and information spreading in the numerical flux function.

Let $a > 0$. The form of the characteristic
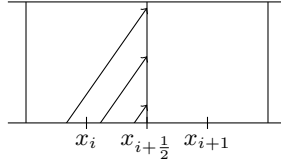
suggests that $\mathcal{F}(U_i^k, U_{i+1}^k)$ should only depend on U_i. We set

$$\mathcal{F}(U_i^k, U_{i+1}^k) = f(U_i^k) \overset{\text{in our model problem}}{=} a \cdot U_i^k.$$

and obtain the flux function for the scheme

$$U_i^{k+1} = U_i^k - \frac{\tau}{h} a \left(U_i^k - U_{i-1}^k \right). \tag{62}$$

In the context of a Finite-Volume method there is a graphic interpretation. As the values U_i^k are *cell averages* we can also interpret them as piecewise constant functions (Figure a):

| time t^k | time $t^{k+1} = t^k + \tau$ |
| (a) (Reconstruct) | (b) (Evolve) |
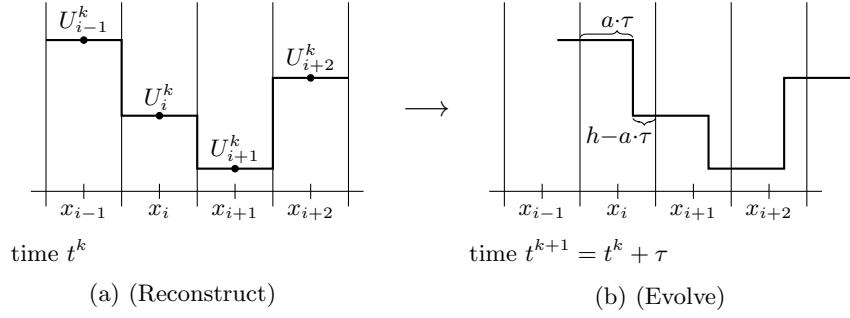
According to the method of characteristics this function is propagated in the time interval τ by $a \cdot \tau$ to the right. Courant $\frac{a \cdot \tau}{h} \leq 1 \iff a \cdot \tau < h$ means that the travel distance has to be smaller than one grid cell (Figure b).

The cell averages at time t^{k+1} are obtained as averages over this unsteady function in each cell:

$$U_i^{k+1} = \frac{a \cdot \tau}{h} U_{i-1}^k + \frac{h - a \cdot \tau}{h} U_i^k = \frac{a \cdot \tau}{h} U_{i-1}^k + \left(1 - \frac{a \cdot \tau}{h} \right) U_i^k$$

convex combination, as $\frac{a \cdot \tau}{h} \leq 1$!

$\Rightarrow$ maximum principle

$$= U_i^k - \frac{\tau}{h} a \left(U_i^k - U_{i-1}^k \right)$$

This is identical to (62)!

For $a < 0$ a similar method is obtained.

For an arbitrary a

$$\mathcal{F}(U_i^k, U_{i+1}^k) = \max(a, 0) \cdot U_i^k + \min(a, 0) \cdot U_{i+1}^k$$
$$= \begin{cases} a U_i^k & a \geq 0 \\ a U_{i-1}^k & a < 0 \end{cases}$$

8.4.4 Godunov Methods

The method described above can be generalised as REA method (*R*econstruct, *E*volve, *A*verage):

1) Reconstruct a *piecewise polynomial* function from cell averages. In the simplest case a piecewise constant function.

2) Solve the hyperbolic equation with this initial condition exactly to obtain a solution at time $t + \tau$.

3) Compute new cell averages from this solution.

This method can be generalised to more complicated equations and was first proposed by Godunov in 1957 for the (non-linear) Euler equation of gas dynamics. It is also the starting point for higher order schemes which reduce the phenomenon of numerical dispersion.

8.5 Higher order schemes with REA

In the framework of REA one can obtain second order precision if the reconstruction step is improved: linear instead of constant reconstruction:
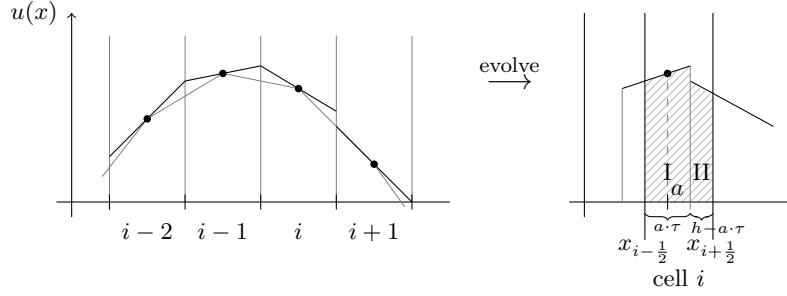
In cell i: Set

$$\tilde{u}_i^k(x) = U_i^k + \sigma_i^k(x - x_i)$$

Attention:

$$\frac{1}{h} \cdot \int\limits_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} U_i^k + \sigma_i^k(x - x_i) \; \mathrm{d}x = U_i^k$$

the slope σ_i^k does not influence the average thus the method is conservative.

The choice of σ_i is discussed below. Let us assume that we know σ_i already.

For Courant $\frac{|a|\tau}{h} \leq 1$ and $a > 0$ evolve and averaging yields:

$$\underbrace{h \cdot U_i^{k+1}}_{\text{area}} = a \cdot \tau \cdot \tilde{u}_{i-1}^k \bigg(\underbrace{\Big(x_{i-\frac{1}{2}} + \frac{a \cdot \tau}{2} \Big)}_{\substack{\text{evaluation point}}} \overset{\uparrow}{\underset{\substack{\text{evolution} \\ \text{of the profile}}}{- a \cdot \tau}} \bigg) + (h - a \cdot \tau)\tilde{u}_i^k \bigg(\underbrace{\Big(x_{i+\frac{1}{2}} - \frac{h - a \cdot \tau}{2} \Big) - a \cdot \tau}_{\substack{x_i + \frac{h}{2} - \frac{h}{2} - \frac{a \cdot \tau}{2} \\ = x_i - \frac{a \cdot \tau}{2}}} \bigg)$$

$$\underset{= x_{i-1} + \frac{h}{2} - \frac{a \cdot \tau}{2} = x_{i-1} + \frac{1}{2}(h - a \cdot \tau)}{}$$

$$= a \cdot \tau \cdot \left(U_{i-1}^k + \sigma_{i-1}^k \Big(\cancel{x_{i-1}} + \frac{1}{2}(h - a \cdot \tau) - \cancel{x_{i-1}} \Big) \right)$$

$$+ (h - a \cdot \tau)\left(U_i^k - \sigma_i^k \Big(\cancel{x_i} - \frac{a \cdot \tau}{2} - \cancel{x_i} \Big) \right)$$

$$= a \cdot \tau U_{i-1}^k + (h - a \cdot \tau)U_i^k + \frac{a \cdot \tau}{2}(h - a \cdot \tau)\sigma_{i-1}^k - (h - a \cdot \tau)\frac{a \cdot \tau}{2}\sigma_i^k$$

divide by h

$$\overset{\downarrow}{\Longleftrightarrow} \boxed{U_i^{k+1} = U_i^k - a\frac{\tau}{h}\left(U_i^k - U_{i-1}^k \right) - \frac{a \cdot \tau}{2}\left(1 - \frac{a \cdot \tau}{h} \right)\left(\sigma_i^k - \sigma_{i-1}^k \right)} \qquad (63)$$

upwind + correction depends on slopes
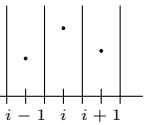
How do we choose σ_i?   Three obvious possibilities are

$$\text{central:} \quad \sigma_i^k = \frac{U_{i+1}^k - U_{i-1}^k}{2h} \quad \text{(Fromm)}$$

$$\text{upwind:} \quad \sigma_i^k = \frac{U_i^k - U_{i-1}^k}{h} \quad \text{(Beam-Warming)} \tag{64}$$

$$\text{downwind:} \quad \sigma_i^k = \frac{U_{i+1}^k - U_i^k}{h} \quad \text{(Lax-Wendroff)}$$

Remarks:

- For Fromm and Beam-Warming the computation of σ_{i-1} needs U_{i-2}!.

- The three schemes are second-order accurate and are stable in $\| \, . \, \|_2$ as long as the CFL-condition is fulfilled.

- The schemes produce oscillations at discontinuities.

It has been proofed that

Satz 8.2 (Godunov, 1959)**.** All monotony preserving, *linear* methods are at most first order accurate.
   See [Lev02]   □

Monotony preserving = does not introduce new minima or maxima.
   Godunov states: There is no linear method (i.e. $L_{h,\tau} U^{k+1} = M_{h,\tau} U^k$), which is second order accurate and monotony preserving.
   The Fromm, Beam-Warming and Lax-Wendroff schemes for example are second order accurate but can lead to oscillations (i.e. non monotonic solutions). Full upwinding is a monotony preserving linear method but only first order accurate.

8.5.1 Slope Limiter Methods

How can Godunov be circumvented? With *non-linear* methods! (though the problem itself is linear)

Idea: Keep the REA approach but choose σ_i^k *depending on the solution*.
   As σ_i^k has to be constrained, this methods are called *slope limiter*.
   One possibility to measure oscillation is the *total variation* TV:

$$TV\left(U^k\right) := \sum_{i=-\infty}^{\infty} |U_i^k - U_{i-1}^k|$$

here: infinite domain.

Definition 8.3. A method is called total variation non increasing (TVNI), if for each step

$$TV(U^{k+1}) \leq TV(U^k).$$

□

Satz 8.4 (Harten 1983)**.** A TVNI-Schema creates no new extrema in the solution. If U^k is monotone, U^{k+1} is monotone as well.

Proof: Given U^k, assume that $U_i^k \leq U_{i+1}^k$ (works also in the other direction). Obviously:

$$TV(U^k) = \sum_{i=-\infty}^{\infty} |\underbrace{U_i - U_{i-1}}_{\geq 0}| = \sum_{i=-\infty}^{\infty} U_i - U_{i-1} = \underset{\underset{\text{Telescope}}{\uparrow}}{U_\infty^k - U_{-\infty}^k}$$

If U^{k+1} has a local minimum at U_{j+1}^{k+1}:

$$\begin{array}{c} j \\ j-1\ \vert\ \ \cdot\ \cdot\ \cdot \\ \cdot\ \cdot\ \cdot\ \ \vert\ j+2 \\ j+1 \end{array}$$
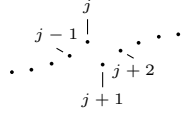
$$TV\left(U^{k+1}\right) = \sum_{i=-\infty}^{j} |\underbrace{U_i^{k+1} - U_{i-1}^{k+1}}_{>0}| + |\underbrace{U_{j+1}^{k+1} - U_j^{k+1}}_{<0}| + \sum_{i=j+2}^{\infty} |\underbrace{U_i^{k+1} - U_{i-1}^{k+1}}_{>0}|$$

$$= U_j^{k+1} - U_{-\infty}^{k+1} + U_j^{k+1} - U_{j+1}^{k+1} + U_\infty^{k+1} - U_{j+1}^{k+1}$$

$$= \underbrace{U_\infty^{k+1} - U_{-\infty}^{k+1}}_{= TV(U^k)} + 2\underbrace{\left(U_j^{k+1} - U_{j+1}^{k+1}\right)}_{>0 \text{ assumption}} \leq TV(U^k) \qquad \text{-} \quad \square$$

they can't change in a single step! Courant!

Therefore it makes sense to search for methods which do not increase the total variation.

In REA methods the total variation is completely determined by the reconstruction. Evolve and average do not increase the total variation (without proof).

A potential choice for the slope is:

$$\sigma_i^k = \text{minmod}\left(\underbrace{\frac{U_{i+1}^k - U_i^k}{h}}_{\substack{\text{downwind} \\ \text{slope}}}, \underbrace{\frac{U_i^k - U_{i-1}^k}{h}}_{\substack{\text{upwind} \\ \text{slope}}},\right)$$

with

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

Idea: Take the smaller slope (keep variation small) or 0, if there is a local extremum.

What happens at a discontinuity?

Assumption: $Cr = \frac{1}{2}$
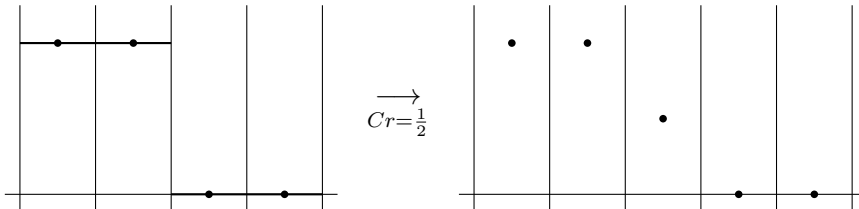
slope 0 everywhere

with slope $-\frac{\frac{1}{2}}{h}$

$\longrightarrow \;$ ...
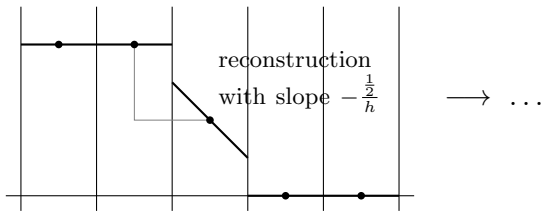
Observation: the reconstructed slope could be a factor 2 larger without violating the monotony. Actually there is the „Superbee" limiter which still has the TVNI property:

$$\sigma_i^k = \text{maxmod}\left(\sigma_i^{(1)}, \sigma_i^{(2)}\right), \qquad \text{maxmod}(a,b) = \begin{cases} a & \text{if } |a| \geq |b| \\ b & \text{if } |b| > |a| \end{cases}$$

with

$$\sigma_i^{(1)} = \text{minmod}\left(\frac{U_{i+1}^k - U_i^k}{h}, 2\frac{U_i^k - U_{i-1}^k}{h}\right)$$

$$\sigma_i^{(2)} = \text{minmod}\left(2\frac{U_{i+1}^k - U_i^k}{h}, \frac{U_i^k - U_{i-1}^k}{h}\right)$$

Remark: if the sign is different $\sigma_i^{(1)} = \sigma_i^{(2)} = 0 \Rightarrow \sigma_i^k = 0$

Example:

$\sigma_i^{(1)} = \text{minmod}(0.8, 2 \cdot 0.2) = 0.4$

$\sigma_i^{(2)} = \text{minmod}(2 \cdot 0.8, 0.2) = 0.2$

$\sigma_i = \text{maxmod}(0.2, 0.4) = 0.4$

if the slopes are very different the result is determined by the smaller one. If both slopes are $\frac{1}{2}$ the result remains $\frac{1}{2}$

- There are many different limiters

- The criteria for a TVNI limiter function are well known but are not covered in this script (see [Lev02] and the cited literature)

8.5.2 Numerical Comparison

Again the model problem, $a = 1$, $h = 1/200$.
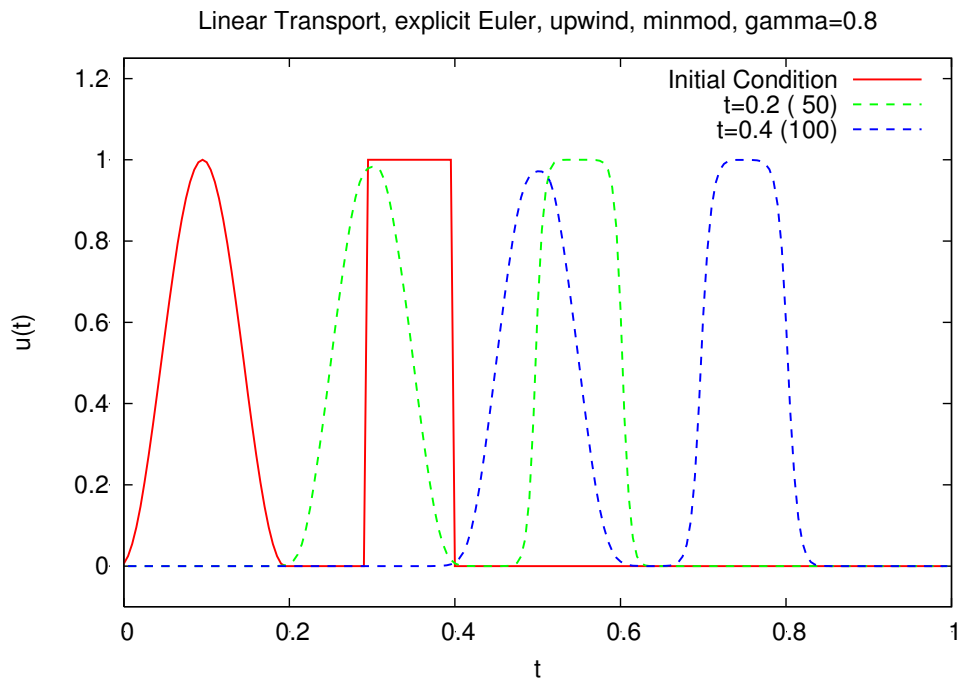
Minmod with $\gamma = 0.8$.
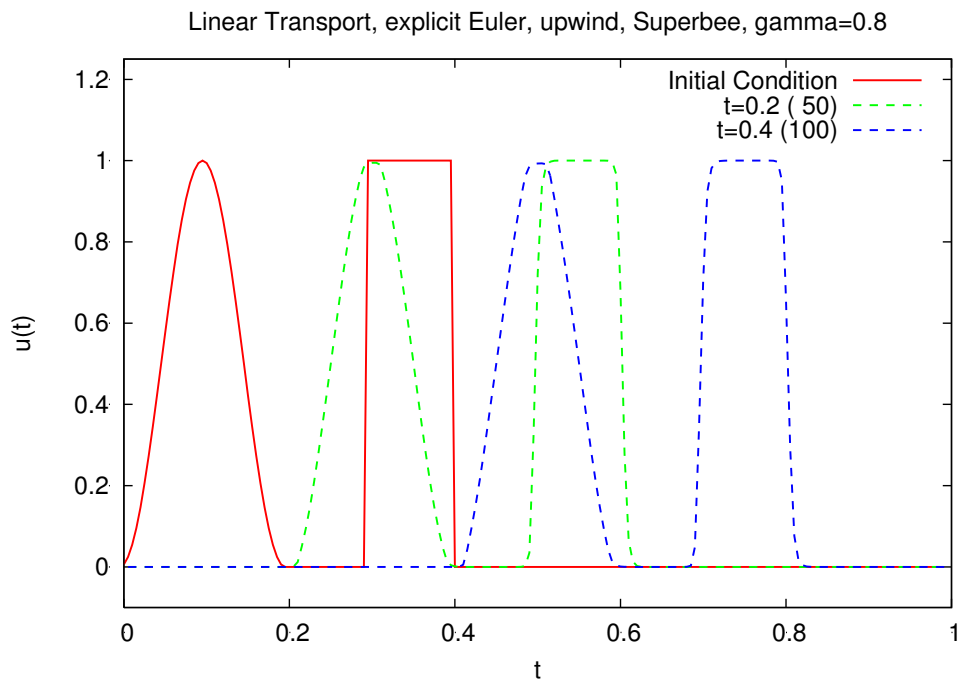
Superbee with $\gamma = 0.8$.

Linear Transport, Comparison of different Schemes, gamma=0.8, t=0.4
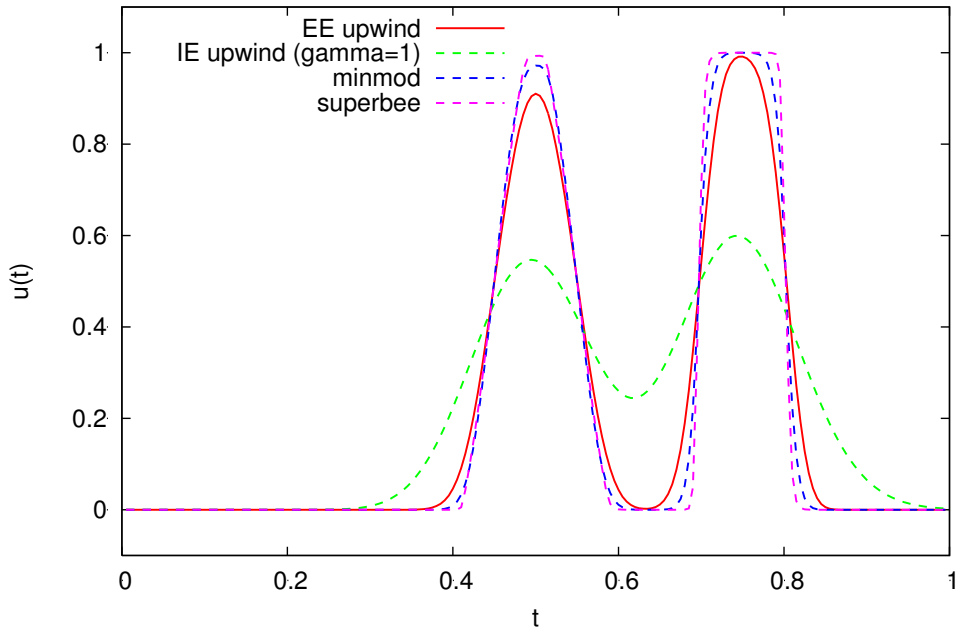
Comparison of different methods with $\gamma = 0.8$.

8.5.3 Summary

- The Satz of Godunov shows that all monotone linear methods are only first order accurate.

- This results in slope-limiter methods, which switch between first and second order according to the solution (and circumvent the Satz of Godunov due to the non-linearity).

8.6 Particle Tracking

Continuous versions of the method of characteristics are possible (e.g. Ellam, Modified method of characteristics...) but difficult. A rather straight forward version is particle tracking.

In particle tracking space and time are treated as continuous but the concentrations are discretised. The solute is represented as a set of particles each of which has the same mass. The particles are propagated with the velocity field and afterwards concentration in a grid cell is calculated by counting the particle number per grid cell and dividing it by the volume of the grid cell.

If $P(\vec{x}, t)$ is the probability for a particle to be at location $\vec{x}$ at time t one can show ([DAD05] that the time dependency of $P(\vec{x}, t)$ is given by the Fokker-Planck-Kolmogorov Equation (FPKE)

$$\frac{\partial P(\vec{x}, t)}{\partial t} = -\frac{\partial}{\partial x}\left[\vec{A}(\vec{x})P(\vec{x}, t)\right] + \frac{1}{2}\frac{\partial^2}{\partial x^2}\left[\bar{B}(\vec{x})P(\vec{x}, t)\right] \tag{65}$$

under the conditions:

- $\vec{A}(\vec{x})$ is the mean of the jump velocity

- $\bar{B}(\vec{x})$ is the statistical dispersion tensor of the jump velocity around its mean

- higher moments cancel out

This equation is analogue to the Convection-Dispersion equation if

$$
\begin{align}
\theta_w C(\vec{x}, t) &\equiv P(\vec{x}, t) \tag{66}\\
\vec{v}_w(\vec{x}, t) &\equiv \vec{A}(\vec{x}) \tag{67}\\
2\bar{D}(\vec{x}, t) &\equiv \bar{B}(\vec{x}) \tag{68}
\end{align}
$$

To get a formal equivallency with the FPKE (Equation8.6) one has to rewrite the CDE:

$$
\frac{\partial\left(\theta_w c_s(\vec{x}, t)\right)}{\partial t} = -\frac{\partial}{\partial x}\left[\left(\vec{v}_w(\vec{x}, t) + \frac{\partial \bar{D}(\vec{x}, t)}{\partial x}\right)\theta_w c_s(\vec{x}, t)\right] + \frac{1}{2}\frac{\partial^2}{\partial x^2}\left[2\bar{D}(\vec{x}, t)\theta_w c_s(\vec{x}, t)\right] \tag{69}
$$

and use $\vec{A}(\vec{x}) = \vec{v}_w(\vec{x}, t) + \frac{\partial \bar{D}(\vec{x}, t)}{\partial x}$.

While $\vec{A}(\vec{x})$ and $\bar{B}(\vec{x})$ are stationary $\vec{v}_w(\vec{x}, t)$ and $\bar{D}(\vec{x}, t)$ are time dependent. However it has been shown in practise that the FPKE can still be used.

8.6.1 Numerical Implementation

We want to determine the position x at time t of a particle that is initially at position $\vec{x}_0$ at time t_0.

The algorithm is given by

$$
\vec{x}(t + \tau) = \vec{x}(t) + \int\limits_{t}^{t+\tau} \vec{v}(\vec{x}(t'))dt' + \sqrt{2\bar{D}(x(t))\tau} \cdot \vec{Z} \tag{70}
$$

where $\vec{Z}$ is a vector of d independent random numbers drawn from a normal deviate (with zero mean and unit variance). This approximation is valid if the time step is not too large. Else especially the random jump for the dispersion can lead to strange results. It makes therefore sense to obey a (local) step size restriction: $\vec{x}(t + \tau) - \vec{x}(t) < c$ where c usually is a fraction of the grid size of the velocity field (similar to the CFL-condition). However, this restriction applies only for individual particles in a certain cell and can change with time and position.

Using the central limit theorem, it can be shown that also a random vector $\sqrt{3}\vec{Z}$ can be used which is composed of random numbers which are uniformly distributed between -1 and 1, which has been shown to be computationally more efficient [Uff85].

Convective Term For the displacement by convection by a stationary flow field $\vec{v}(\vec{x})$ we get $\frac{\partial x}{\partial t} = \vec{v}(\vec{x})$. We can separate the variables and obtain by integration:

$$
\int\limits_{x_0}^{x} \frac{1}{\vec{v}(\vec{x})}dx = t - t_0 \tag{71}
$$

For a constant flux this of course just yields $\Delta\vec{x} = \vec{v}\tau$. For our interpolated flux field $\vec{v} = \begin{pmatrix} ax + b \\ cy + d \\ ez + f \end{pmatrix}$ we get e.g. for the x component:

$$\int_{x_0}^{x} \frac{1}{ax + b} dx = \frac{1}{a} \log\left(\frac{ax + b}{ax_0 + b} \right) = \tau \tag{72}$$

or $x = (x_0 + \frac{b}{a})\exp(a\tau) - \frac{b}{a}$

$$\vec{x} = \begin{pmatrix} (x_0 + \frac{b}{a})\exp(a\tau) - \frac{b}{a} \\ (y_0 + \frac{d}{c})\exp(c\tau) - \frac{d}{c} \\ (z_0 + \frac{f}{e})\exp(e\tau) - \frac{f}{e} \end{pmatrix} \tag{73}$$

If the particle crosses the boundary between two grid cells, the step has to be interrupted at the boundary and continued with the new velocity coefficients.

Alternatively the velocity field can be integrated with a stable numerical integration formula of matching order like the midpoint method:

$$\vec{x}_{n+1/2} = \vec{x}_n + \frac{1}{2}\tau\vec{v}(\vec{x}_n) \tag{74}$$
$$\vec{x}_{n+1} = \vec{x}_n + \tau\vec{v}(\vec{x}_{n+1/2}) \tag{75}$$

Diffusive Term The additional term $\frac{\partial \bar{D}(\vec{x},t)}{\partial x}$ in the effective convection velocity is necessary to avoid unphysical results. However, it can only be evaluated, if the variation in $\bar{D}(\vec{x},t)$ is smooth enough to calculate a first order derivative. This is for example given if the change in the dispersivity is just due to small velocity changes.

If there are jumps in the dispersion coefficient due to changes in porosity or water content other means are necessary. One possibility is to introduce a reflection principle. If a particle reaches the interface between two materials, an additional random number is drawn. If the random number is larger than a reflection coefficient, the particle crosses the interface else it is reflected. [SAM93] derive the condition $P_\lambda = \frac{\sqrt{D_\lambda}}{\sqrt{D_\lambda} + \sqrt{D_\gamma}}$ for a particle to enter (or remain in) material λ and $P_\gamma = 1 - P_\lambda$ to enter (or remain in) material γ. The same criterium is used for particles comming from either side of the interface.

A different water content can be taken into account by modifying the probabilities according to [Lim06] to $P_\lambda = \frac{\theta_\lambda\sqrt{D_\lambda}}{\theta_\lambda\sqrt{D_\lambda} + \theta_\gamma\sqrt{D_\gamma}}$ and $P_\gamma = 1 - P_\lambda = \frac{\theta_\gamma\sqrt{D_\lambda}}{\theta_\lambda\sqrt{D_\lambda} + \theta_\gamma\sqrt{D_\gamma}}$.

An alternative is to smooth the jump at the interface over a small interval. This is called the interpolation method [LFT96, SFGGH06].

[BVIV11] published an improved reflection scheme, which uses a different way to split the distance which is covered in the two adjacent elements over one time step by splitting the time according to the square roots of time:

$$\sqrt{\tau_2} = \sqrt{\tau} - \sqrt{\tau_1}$$

. They also proposed an one-sided splitting scheme, where the particles from the side with the lower product $\theta\sqrt{D}$ are let through unhindered, whereas the particles from the other side are transmitted with the probability $P_\lambda = \frac{\theta_\lambda\sqrt{D_\lambda}}{\theta_\gamma\sqrt{D_\gamma}}$.

Timestep Control In the simplest form the same timestep is used for all particles. The size of the timestep is then limited by the largest velocity to fulfill the CFL condition. This has the advantage, that the positions of all particles are known after each timestep to calculate e.g. the movement of the center of mass of the particle distribution or its spreading.

Alternatively the timestep can be choosen for each particle individually. The timestep is then only very small if the velocity at the position where the particle is at the moment is high. If for example we use the integration formula and have calculated the velocity $\vec{v}(\vec{x}_{n+1/2})$ we can determine the timestep size as $\tau < \frac{ch}{||\vec{v}(\vec{x}_{n+1/2})||_2}$. where c is a constant smaller than one. However, if the convection velocity is very small, diffusion can be the dominating processes. The timestep limit is then given by $\tau < \frac{c^2h^2}{2||\bar{D}(x(t))||_2}$. The total timestep condition would then be

$$\tau < \max\left(\left|\left|\frac{ch}{\vec{v}(\vec{x}_{n+1/2})}\right|\right|_2, \left|\left|\frac{c^2h^2}{2\bar{D}(x(t))}\right|\right|_2\right)$$

For diffusion dominated problems this limit can get rather severe.

Calculation of Concentrations While the calculation of spatial moments of the solute is possible without any spatial discretisation the particles have to be projected on a grid to calculate concentrations:

$$c_s(\vec{X}_c, t) \propto \sum_{i=1}^{N} m_i W_c(\vec{x}_i(t) - \vec{X}_c) \tag{76}$$

where $\vec{X}_c$ is the centroid of grid cell c, m_i is the mass of the particles and W_c is a projection function selecting particles inside the grid cell c. This counting can be done for each particle individually, which makes the method trivially parallel.

If a particle reaches an outflow boundary, a counter in a field of time intervals can be increased yielding a breakthrough curve.

8.6.2 Initial and Boundary Conditions

For the initial condition particles are distributed randomly according to the solute concentration in regions where solutes are present.

No-flux boundary conditions are easy to realize by implementing a reflection at the boundary. Outflow boundaries are also easy. The tracking of a particle is terminated if it reaches an outflow boundary. Dirichlet boundary conditions are more difficult to realize as the number of particles can increase very rapidly.

8.6.3 Assets and Drawbacks

Pro

- Particle tracking is nearly completly free of numerical dispersion for convection dominated cases

- No time and space discretisation necessary

- It can be easily parallelized

- Implementation is straightforward

- Linear sorption and decay can easily be integrated

- Easy to implement

Con

- Discrete concentrations $\Rightarrow$ chemical reactions are hard to realize

- Time-dependent boundary conditions difficult

- Diffusion dominated flow is very slow

- Random fluctuations in concentrations occur which are proportional to the square root of the partical number $\Rightarrow$ high accuracy is very expensive

- High quality random number generator necessary

9 Solution of non-linear Equations - Sorption

9.1 Sorption

Solutes can be bound (adsorbed) to the surface of the solid phase. The nature of this binding varies for different solutes and solid phases.

Macroscopically sorption is described by sorption isotherms. They are relations between the 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 sorbed material only depends on the concentration in the fluid phase and a material parameter characterising the intensity of the binding. These assumptions are true for low concentrations.

$$c_{s_{\text{sorb}}} = K_s c_s \tag{77}$$

where K_s is the sorption parameter $[\text{m}^3 \text{ kg}^{-1}]$ The sorbed amount of substance is given by

$$n_{s_{\text{sorb}}} = \rho_b c_{s_{\text{sorb}}} = \rho_b K_s c_s \tag{78}$$

where ρ_b is the bulk density of the soil $[\text{kg m}^{-3}]$, i.e. the mass of the solute phase per volume of soil.

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

assumes that the energy of the sorption sites decreases logarithmically

Langmuir Isotherm
$$c_{s_{\text{sorb}}} = \frac{K_L c_{\text{max}} c_s}{1 + K_L c_s}$$

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

Convection-Dispersion Equation with Sorption

The Convection-Dispersion equation including sorption can be written as

$$\frac{\partial \left[\theta_w c_s(\vec{x}) + \rho_b c_{s_{\text{sorb}}}(c_s(\vec{x}))\right]}{\partial t} - \nabla \cdot \left(\bar{D}(\vec{x}, \theta_w)\nabla c_s(\vec{x})\right) + \nabla \cdot \left(c_s \vec{J}_w(\vec{x})\right) + r_s(\vec{x}) = 0 \qquad (79)$$

For linear sorption

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

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.

For non-linear sorption isotherms we get a non-linear partial differential equation.

Langmuir Isotherm

If the Langmuir isotherm is used it is still possible to implement a rather simple explicit model. With $n_s = \theta_w c_s + \rho_b \frac{K_L c_{\text{max}} c_s}{1 + K_L c_s}$ we get the equation

$$\frac{\partial n_s(\vec{x}_i)}{\partial t} \approx \frac{n_{s_i}^{k+1} - n_{s_i}^k}{\tau} = F(\vec{x}_i, t^k)$$

if we know c_s^k we can calculate

$$n_{s_i}^{k+1} = n_{s_i}^k + \tau F(\vec{x}_i, t^k)$$

To get c_s^{k+1} we have to solve the equation

$$n_s^{k+1} = \theta_w c_s^{k+1} + \rho_b \frac{K_L c_{\text{max}} c_s^{k+1}}{1 + K_L c_s^{k+1}}$$

This is a quadratic equation for c_s^{k+1} and as the concentration has to be positive we get the result:

$$c_s^{k+1} = \frac{1}{2K_L\theta_w}\left[\left(K_L n_s^{k+1} - \rho_b K_L c_{\text{max}} - \theta_w\right)\right. \qquad (80)$$

$$\left. + \sqrt{\left(\rho_b K_L c_{\text{max}} + \theta_w - K_L n_s^{k+1}\right)^2 + 4K_L\theta_w n_s^{k+1}}\right] \qquad (81)$$

For the Freundlich Isotherm the equation

$$n_s^{k+1} = \theta_w c_s^{k+1} + \rho_b K_F (c_s^{k+1})^n$$

is not directly invertible and it is necessary to solve the equation numerically.

9.2 Solving non-linear Equations

If we use the Freundlich sorption isotherm we have to solve the equation

$$f(c_s^{k+1}) = n_s^{k+1} - \theta_w c_s^{k+1} + \rho_b K_F (c_s^{k+1})^n = 0$$

for c_s^{k+1} each grid point. There are different methods to find the root of a non-linear equation.

9.2.1 Interval Bisection

The first method for the solution of non-linear equations we want to discuss is interval bisection.

Idea: Let us assume that an interval $I_0 = [a_0, b_0]$ exists, where $f(a_0), f(b_0)$ have different sign, i.e. $f(a_0) \cdot f(b_0) < 0$. According to the intermediate value theorem (for steady functions) f has at least one root in $[a_0, b_0]$.

This leads to the following algorithm:

Given: $I_0 = [a_0, b_0]$ with $f(a_0) \cdot f(b_0) < 0$ and tolerance ε;
for $(t = 0, 1, \ldots)$ **do**
   $x_t = \frac{1}{2}(a_t + b_t)$; {center of the interval}
   if $(f(x_t) = 0)$ **then**
     break; {ready!}
   end if
   if $(f(a_t)f(x_t) < 0)$ **then**
     $a_{t+1} = a_t$; $b_{t+1} = x_t$; {root in $[a_t, x_t]$}
   else
     $a_{t+1} = x_t$; $b_{t+1} = b_t$; {$f(x_t)f(b_t) < 0$ as $VZ(x_t) = VZ(a_t)$!}
   end if
   if $(b_t - a_t < \varepsilon)$ **then**
     break; {error is acceptable}
   end if
end for

In each step we have

$$a_t \leq a_{t+1} < b_{t+1} \leq b_t$$

and

$$|b_{t+1} - a_{t+1}| = \frac{1}{2}|b_t - a_t| = \left(\frac{1}{2}\right)^{t+1}|b_0 - a_0|.$$

Therefore the scheme has the following properties:

- The convergence rate is $\frac{1}{2}$ per step.

- Bisection is numerically stable (insusceptible to cancellation errors) and therefore the method of choice for scalar functions with only one root in a given interval.

- Unfortunately the method can only be applied for real functions (not for e.g. complex functions).

9.2.2 Fixpoint Iteration

Root finding can be reformulated to the search for a fixpoint.

For a given $f : I \to \mathbf{R}$ we formulate the auxiliary function

$$g(x) = x + \sigma f(x) \qquad \text{with } 0 \neq \sigma \in \mathbf{R}.$$

Obviously

$$g(x) = x \qquad \Leftrightarrow \qquad x + \sigma f(x) = x$$
$$\Leftrightarrow \qquad \sigma f(x) = 0$$
$$\Leftrightarrow \qquad f(x) = 0 \ .$$

The search for roots of f therefore is equivalent to the search for fixpoints

$$g(x) = x$$

of g.

The search for fixpoints is analysed by the following Satz.

Satz 9.1 (Banachscher[5] Fixpunktsatz). Let $I \subset \mathbf{R}$ be a non-empty, closed interval and $g : I \to I$ a „Lipschitz[6]-steady" transformation

$$|g(x) - g(y)| \leq q|x - y| \qquad x, y \in I$$

with $q < 1$ (contraction). Then the sequence generated by

$$x^{(t+1)} = g(x^{(t)})$$

converges for arbitrary initial values to the unique fixpoint $z \in I$.

An approximation for the error is given by:

$$|x^{(t)} - z| \leq \frac{q}{1-q}|x^{(t)} - x^{(t-1)}| \leq \frac{q^t}{1-q}|x^{(1)} - x^{(0)}|.$$

Proof: As $g : I \to I$ the sequence $x^{(t)} = g(x^{(t-1)}) = g(g(x^{(t-2)})) = \ldots g^t(x^{(0)})$ is well defined. Additionally we have:

$$|x^{(t+1)} - x^{(t)}| = |g(x^{(t)}) - g(x^{(t-1)})| \leq q|x^{(t)} - x^{(t-1)}| \leq \ldots \leq q^t|x^{(1)} - x^{(0)}|$$

Now we show that $x^{(t)}$ is a Cauchy sequence. Let $\varepsilon > 0$ and $m \geq 1$ be given

$$\begin{aligned}
|x^{(t+m)} - x^{(t)}| &\leq |x^{(t+m)} - x^{(t+m-1)} + x^{(t+m-1)} - x^{(t+m-2)} + \ldots + x^{(t+1)} - x^{(t)}| \\
&\leq |x^{(t+m)} - x^{(t+m-1)}| + |x^{(t+m-1)} - x^{(t+m-2)}| + \ldots + |x^{(t+1)} - x^{(t)}| \\
&\leq q^{t+m-1}|x^{(1)} - x^{(0)}| + q^{t+m-2}|x^{(1)} - x^{(0)}| + \ldots + q^t|x^{(1)} - x^{(0)}| \\
&\leq (q^{t+m-1} + q^{t+m-2} + \ldots + q^t)|x^{(1)} - x^{(0)}| \\
&\leq q^t \frac{1-q^m}{1-q}|x^{(1)} - x^{(0)}| \leq \varepsilon \qquad \text{for } t \geq t(\varepsilon) \text{ large enough.}
\end{aligned}$$

Due to the completeness axiom every Cauchy sequence converges to a limit $z \in \mathbf{R}$. Because of $g : I \to I$ and I closed we have $z \in I$.

---

[5]Stefan Banach, 1892-1945, polish mathematician.
[6]Rudolf O. S. Lipschitz, 1832-1903, German mathematician.

Error estimate:

$$\begin{aligned}
|x^{(t+m)} - x^{(t)}| &\leq |x^{(t+m)} - x^{(t+m-1)}| + \ldots + |x^{(t+1)} + x^{(t)}| \quad \text{(as above)} \\
&\leq q^m |x^{(t)} - x^{(t-1)}| + \ldots + q|x^{(t)} - x^{(t-1)}| \\
&\leq (q^m + \ldots + q)|x^{(t)} - x^{(t-1)}| \\
&\leq \frac{q}{1-q}|x^{(t)} - x^{(t-1)}|
\end{aligned}$$

$x^{(t+m)}$ converges to z for $m \to \infty$, the right side is independent of m, therefore

$$|z - x^{(t)}| \leq \frac{q}{1-q}|x^{(t)} - x^{(t-1)}| \leq \frac{q^t}{1-q}|x^{(1)} - x^{(0)}|.$$

Uniqueness: Let $z' \neq z$ be an additional fixpoint then

$$|z - z'| = |g(z) - g(z')| \leq q|z - z'| \quad \Leftrightarrow \quad 1 \leq q \quad (z - z' \neq 0).$$

This is a contradiction to $q < 1$ (Lipschitz). Therefore $z = z'$. $\qquad\square$

Remark 9.2. A sufficient condition for the Lipschitz steadiness of g is $|g'(x)| \leq q$ for all $x \in I$.
From to the mean value theorem of differential calculus we conclude:

$$\frac{g(x) - g(y)}{x - y} = g'(\xi) \Leftrightarrow g(x) - g(y) = g'(\xi) \cdot (x - y)$$
$$\Rightarrow |g(x) - g(y)| = |g'(\xi)||x - y|$$

and thus Lipschitz steadiness if $g'(x) \leq q$ for all $x \in I$.
If additionally $q < 1$ we get the contraction property. $\qquad\square$

Remark 9.3. $|g'(x)| \leq q$ is a sufficient condition for Lipschitz steadiness.
For the function $|x|$:
$$|\,|x| - |y|\,| \leq |x - y|$$
we get Lipschitz steadiness with the constant 1.
The advantage of Banachs Fixpunktsatz is that the iteration function g does *not* have to be differentiable. $\qquad\square$

Geometrical Interpretation of the fixpoint iteration
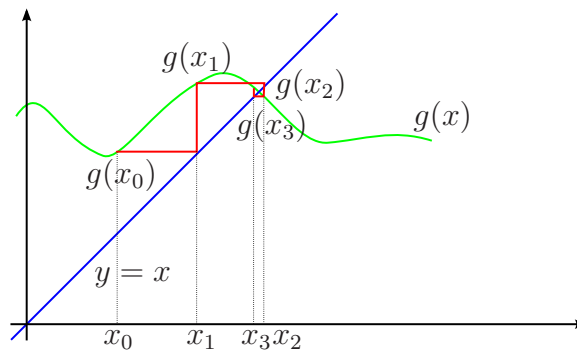
Remark 9.4. Banachs' Fixpunktsatz can be generalised to functions $g : G \to \mathbf{R}^n$, $G \subseteq \mathbf{R}^n$. Accordingly we demand again

$$\|g(x) - g(y)\| \leq q\|x - y\|, \qquad x, y \in G$$

with $q < 1$.

The iterative solution of $Ax = b$ corresponds to the search for the root $f(x) = b - Ax = 0$. Relaxation methods could be written as

$$x^{(k+1)} = x^{(k)} + M^{-1}(b - Ax^{(k)}) = \underbrace{(I - M^{-1}A)}_{S}\, x^{(k)} + \underbrace{M^{-1}b}_{c} = g(x^{(k)}).$$

Analysis of the Lipschitz Steadiness of g:

$$\|g(x) - g(y)\| = \|Sx - Sy\| = \|S(x - y)\| \leq \|S\|\|x - y\|.$$

For $\|S\| < 1$ we get convergence independent of the initial value. $\qquad\qquad\square$

9.2.3 Newton's Method

The search for the root $f(x) = 0$ can be derived from a geometrical idea.

Replace the function f at point $x^{(t)}$ by the tangent of f and calculate its root. This is the new approximation $x^{(t+1)}$.
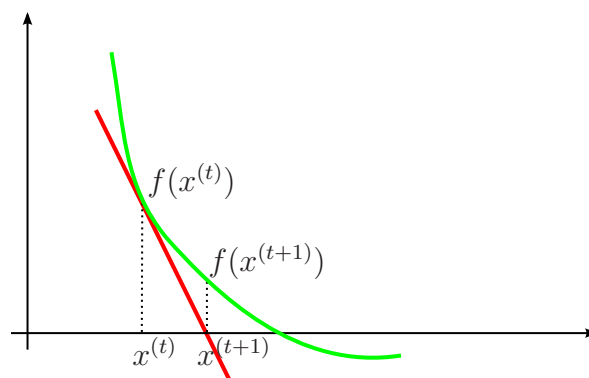
Formally the equation for the tangent at point $x^{(t)}$ is

$$T(x) = f'(x^{(t)})(x - x^{(t)}) + f(x^{(t)}).$$

The root of the tangent is given by

$$T(x^{(t+1)}) = 0 \qquad \Leftrightarrow \qquad f'(x^{(t)})(x^{(t+1)} - x^{(t)}) + f(x^{(t)}) = 0$$

$$\Leftrightarrow \qquad\qquad x^{(t+1)} = x^{(t)} - \frac{f(x^{(t)})}{f'(x^{(t)})}.$$

Obviously the precondition is $f'(x^{(t)}) \neq 0$, i. e. there is only a *single* root at point x.

The convergence properties of Newton's method are given by the following Satz.

Satz 9.5. Let the function $f \in C^2[a,b]$ have a root z in (a,b) and let

$$m := \min_{a \le x \le b} |f'(x)| > 0, \qquad M := \max_{a \le x \le b} |f''(x)|.$$

If $\varrho > 0$ is chosen to get

$$q = \frac{M}{2m}\varrho < 1, \quad K_\varrho(z) = \{x \in \mathbf{R} \,|\, |x - z| \le \varrho\} \subset [a,b].$$

Then the newton iterates $x^{(t)} \in K_\varrho(z)$ are defined for each initial value $x^{(0)} \in K_\varrho(z)$ and we get the estimates

$$|x^{(t)} - z| \le \frac{2m}{M}q^{(2^t)} \quad \text{and} \quad |x^{(t)} - z| \le \frac{M}{2m}|x^{(t)} - x^{(t-1)}|^2 \quad \text{respectively.}$$

Proof: See [Ran06, Satz 5.1]. $\qquad\square$

Remark 9.6. • Newton's method converges „quadratically":

$$|x^{(t)} - z| \le C|x^{(t)} - x^{(t-1)}|^2, \qquad |x^{(t)} - z| \le q^{(2^t)}.$$

Bisection and fixpoint iteration only have „linear" convergence:

$$|x^{(t)} - z| \le C|x^{(t)} - x^{(t-1)}|, \qquad |x^{(t)} - z| \le Cq^t.$$

i.e. $C = 1, q = 0.1$:

t	linear	quadratic
1	10^{-1}	10^{-1}
2	10^{-2}	10^{-2}
3	10^{-3}	10^{-4}
4	10^{-4}	10^{-8}

With linear convergence the number of valid digits is proportional to t, with quadratic convergence it doubles in each step!

• The disadvantage of Newton's method is the only *local* convergence, i.e. the initial value has to be sufficiently close to the solution. $\qquad\square$

Example 9.7. We want to calculate the nth root of x, i.e. we solve

$$f(x) = x^n - a = 0 \qquad \text{for } a > 0.$$

Newton's method is given by

$$x^{(t+1)} = x^{(t)} - \frac{f(x^{(t)})}{f'(x^{(t)})} = x^{(t)} - \frac{(x^{(t)})^n - a}{n(x^{(t)})^{n-1}}.$$
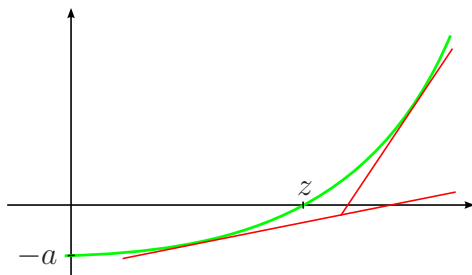
Converges for each $x^{(0)} > 0$ to the positive root as $x^{(1)} > z$ for $x^{(0)} < z$ and the sequence is *monotone* decreasing for $x^{(t)} > z$.

For $n = 2$ (i.e. $x^2 - a = 0$) we get quadratic convergence if $|x^{(t)} - \sqrt{a}| < 2\sqrt{a}$. $\qquad\square$

The so called *damped Newton's method* is given by

$$x^{(t+1)} = x^{(t)} - \lambda^{(t)} \frac{f(x^{(t)})}{f'(x^{(t)})}$$

with $\lambda^{(t)} \in (0, 1]$.

Instead of adding the full correction it is first multiplied by a „damping" factor $\lambda^{(t)}$.

With a suitable choice of $\lambda^{(t)}$ the „area of convergence" can be increased.

The derivative $f'(x^{(t)})$ in Newton's method can also be calculated by numerical differentiation.

- still yields quadratic convergence

- susceptible to cancellation.

If $f'(x)$ is not calculated exactly this is often called a *Quasi*-Newton's method.

9.2.4 Newton's Method in $\mathbf{R}^n$

Now we want to solve the n-dimensional problem

$$f_i(x_1, \ldots, x_n) = 0 \qquad i = 1, \ldots, n$$
$$\Leftrightarrow \qquad \vec{f}(\vec{x}) = 0 \text{ with } \vec{x} = (x_1, \ldots, x_n)^T \text{ and } \vec{f} : \mathbf{R}^n \to \mathbf{R}^n.$$

A Taylor series in $\mathbf{R}^n$ yields a generalisation of the tangent:

$$\vec{f}(\vec{x} + \Delta\vec{x}) = \vec{f}(\vec{x}) + \bar{J}(\vec{x})\Delta\vec{x} + \text{ remainder}.$$

$\bar{J}(\vec{x})$ is the *Jacobian* at position $\vec{x}$:

$$\left(\bar{J}(\vec{x})\right)_{i,j} = \frac{\partial \vec{f_i}}{\partial x_j}(\vec{x}) \quad \in \mathbf{R}^{n \times n}.$$

Determination of the root of the „tangent" yields:

$$\vec{f}(\vec{x}^{(t)}) + \bar{J}(\vec{x}^{(t)})(\vec{x}^{(t+1)} - \vec{x}^{(t)}) \overset{!}{=} 0$$
$$\Leftrightarrow \quad x^{(t+1)} = \vec{x}^{(t)} - \left(\bar{J}(\vec{x})\right)^{-1} \vec{f}(\vec{x}^{(t)}).$$

Each step of Newton's method requires the solution of a linear equation system

$$\bar{J}(\vec{x}^{(t)})v = \vec{f}(\vec{x}^{(t)}).$$

This is again done with direct or iterative methods.

For inexact or Quasi-Newton methods the linear equation system is either

- solved only approximately, or

- the Jacobian is not assembled in each Newton step.

9.2.5 Summary

- Non-linear algebraic equations can only be solved iteratively. Thus the convergence for the presented methods is only guaranteed under certain conditions.

- For scalar functions with only one root in the interval bisection is ideal if robustness is important.

- Fixpoint iteration requires that the iteration function is a contraction. However, it converges independent of the initial value.

- Newton's method requires the differentiability of the non-linear function and converges only if the initial value is sufficiently close to the solution. On the other side it is very fast due to the quadratic convergence.

- Fixpoint iteration as well as Newton's method can be generalised to systems.

10 Richards Equation

Water transport in soils at the interface between atmosphere and geosphere is of utmost importance for the mankind. Soils supply crops with water, nutrients and footing. They also act as a filter for drinking water as contaminated water has to pass the unsaturated zone to reach the aquifers. Evaporation from bare soil and transpiration from plants is an important climate control. Soils also store CO_2 and can produce Greenhouse gases like methane and nitrous oxide. Soils are the habitat of many microorganisms and animals.

10.1 Flux Law

Buckingham [Buc07] proposed in 1907 an extension of Darcy's law, which describes water flow in unsaturated soils:

$$J_w = -K_w(\theta_w)\frac{d\psi_w}{dz}$$

today this is called the Buckingham-Darcy law. In three dimensions we would of course get

$$\vec{J}_w = -\bar{K}_w(\theta_w, \vec{x})\nabla\psi_w$$

$\psi_w = \psi_m - \rho_w gz$ is the total water potential, consisting of the matrix potential ψ_m and the gravity potential $\rho_w gz$ (this is the form for constant density else the gravity potential is $\int_0^z \rho_w(z)gdz$).

- The matrix potential ψ_m is defined as the energy to extract an infinitesimal small quantity of water from the capillary bound state to free water. In civil engineering the capillary pressure $p_c = -\psi_m$ is used instead of the matrix potential.

- In soil physics it is very common to use the pressure head $h_m = \frac{\psi_m}{\rho_w g}$ which is the length of an equivalent water column hanging below the sample creating an underpressure.

- The hydraulic conductivity $K_w(\theta_w)$ is a material property. It describes the decrease of the water conductivity with water content. It is usually strongly non-linear with a steeper gradient in the wet range.

10.2 Richards Equation

Richards Equation

Together with mass conservation one obtains an equation proposed for the first time by Richards[Ric31].

$$\frac{\partial \theta_w}{\partial t} \quad + \quad \nabla \cdot \vec{J}_w + r_w = 0 \tag{82}$$

$$\frac{\partial \theta_w}{\partial t} \quad - \quad \nabla \cdot \left[\bar{K}(\theta_w, \vec{x}) \nabla \psi_w \right] + r_w = 0 \tag{83}$$

In this form the equation contains two independent variables ψ_w and θ_w. An algebraic relation between these variables is therefore necessary. In the form $\theta_w = f(\psi_w)$ it is called soil water retention curve, in the form $\psi_m = g(\theta_w)$ soil-moisture characteristic curve and in the form $p_c = g(\theta_w)$ it is called capillary pressure saturation curve. The relation is often not unique but may be hysteretic.

10.3 Formulations

There are different formulations of Richards equation.

Potential Form of Richards' equation

If we use the chain rule $\frac{\partial \theta}{\partial t} = \frac{\partial \theta}{\partial \psi_m} \frac{\partial \psi_m}{\partial t}$ with the specific soil water capacity $C_w(\psi_m) = \frac{\partial \theta}{\partial \psi_m}$ (which is the derivative of the soil water characteristic) we get the potential form of Richards equation:

$$C_w(\psi_m) \frac{\partial \psi_m}{\partial t} - \nabla \cdot \left[\mathbf{K}(\psi_m) \left(\nabla \psi_m - \rho g \mathbf{e_z} \right) \right] + \gamma = 0$$

If we write the equation in terms of the pressure head we get

$$C_w(h_m) \frac{\partial h_m}{\partial t} - \nabla \cdot \left[\mathbf{K}(h_m) \left(\nabla h_m - \mathbf{e_z} \right) \right] + \gamma = 0$$

While the potential form allows to write Richards equation in terms of just one variable it leads to non-mass conservative formulations if discretised with first order time discretisations as the specific water capacity has to be evaluated either at the old or the new time step.

Water Content Form of Richards' equation

If instead we use the chain rule $\frac{\partial \psi_m}{\partial x} = \frac{\partial \psi_m}{\partial \theta} \frac{\partial \theta}{\partial x}$ and call $\mathbf{K}(\theta) \cdot \frac{\partial \psi_m}{\partial \theta} = \mathbf{D_w}(\theta)$ the soil water diffusivity, we obtain Richards equation in the water content form:

$$\frac{\partial \theta}{\partial t} - \nabla \cdot \left[\mathbf{D_w}(\theta) \nabla \theta - \mathbf{K}(\theta) \rho g \mathbf{e_z} \right] + \gamma = 0$$

The water content form has advantages for the analysis of Richards equation. However, it can only be used in unsaturated conditions as both the time and the space derivatives vanish if the soil becomes saturated.

Mixed Form of Richards' equation

$$\frac{\partial \theta(\psi_m)}{\partial t} - \nabla \cdot \left[\mathbf{K}(\theta(\psi_m)) \left(\nabla \psi_m - \rho g \mathbf{e_z} \right) \right] + \gamma = 0$$

The mixed form allows a mass conservative solution and can be used for saturated as well as for unsaturated conditions as only the time derivative vanishes and not the space derivative if we allow positive total water potentials (which are equivalent to the pressures in saturated flow). However, it requires the solution of a non-linear equation system.

10.4 PDE Classification

At a first glance Richards equation looks like a typical parabolic equation. However this is not true. If we look at the (one-dimensional) water content form:

$$\frac{\partial \theta_w}{\partial t} - \frac{\partial}{\partial z} \left[\bar{D}_w(\theta_w) \frac{\partial \theta_w}{\partial z} - K_w(\theta_w) \rho_w g \right] + r_w = 0$$

and split the divergence

$$\frac{\partial \theta_w}{\partial t} - \frac{\partial}{\partial z} \left[\bar{D}_w(\theta_w) \frac{\partial \theta_w}{\partial z} \right] + \rho_w g \frac{\partial K_w(\theta_w)}{\partial \theta_w} \frac{\partial \theta_w}{\partial z} + r_w = 0$$

we see that we have a convection term caused by gravity with the velocity $v_w(\theta_w) = \rho_w g \frac{\partial K_w(\theta_w)}{\partial \theta_w}$.

This is not the whole truth. By applying the chain rule to the first transport term we get

$$\frac{\partial}{\partial z} \left[\bar{D}_w(\theta_w) \frac{\partial \theta_w}{\partial z} \right] = \bar{D}_w(\theta_w) \frac{\partial^2 \theta_w}{\partial z^2} + \frac{\partial \bar{D}_w(\theta_w)}{\partial z} \cdot \frac{\partial \theta_w}{\partial z}$$

the second part is again a convection term.

Therefore Richards equation can be written as a non-linear convection-dispersion equation:

$$\frac{\partial \theta_w}{\partial t} + \vec{V}_w(\theta_w) \nabla \theta_w - \bar{D}_w(\theta_w) \Delta \theta_w + r_w = 0$$

with the velocity $\vec{V}_w(\theta_w) = \rho_w g \frac{\partial K_w(\theta_w)}{\partial \theta_w} \vec{e}_z - \nabla \bar{D}_w(\theta_w)$.

Thus Richards equation is a degenerate parabolic equation and can get effectively hyperbolic at steep fronts, when the gradient of the diffusivity is high (even in horizontal flow) and in vertical flow near saturation, when gravity is the main driving force. The behaviour depends strongly on the material functions.

10.5 Hydraulic Functions

The two important functions characterising the hydraulic properties of a porous medium are the soil water retention curve and the hydraulic conductivity function. Especially the unsaturated hydraulic conductivity of a soil sample is very difficult to measure. Therefore it is common to use parametrised functions which reduces the measurement problem to the determination of suitable parameters for the sample.

10.5.1 Soil Water Retention Curve

The soil water retention curve depends mainly on the pore size distribution (which itself depends on the shape and the size of the constituents) and the topography of the pores.

Both of the parametrisations shown here give a formula to calculate the effective saturation

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

10.6 Hydraulic Functions

10.6.1 Soil Water Retention Curve

Brooks-Corey

Brooks and Corey proposed in 1966 the model

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

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.

Dependency of Brooks-Corey model on λ
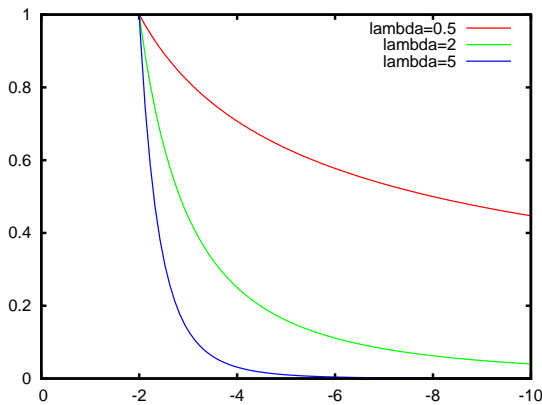
Dependency of Brooks-Corey model on ψ_{m_0}

Van Genuchten

To circumvent the problem with the unsteadiness of the derivative van Genuchten proposed in 1980 an alternative parametrisation.

$$S_{\text{eff}}(\psi_m) = \frac{\theta - \theta_{w_r}}{\theta_{w_s} - \theta_{w_r}} = [1 + (\alpha|\psi_m|)^n]^{-m}$$
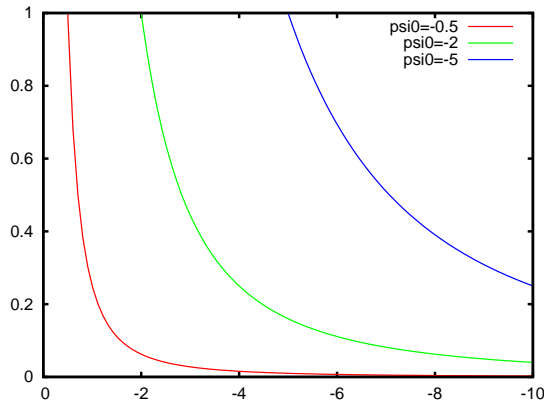
Parameters:

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

α its inverse is the point of inflection of the soil water retention curve. Thus for high n's (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 usually not a free parameter but is set to $m = 1 - \frac{1}{n}$.

Dependency of van Genuchten model on n

Dependency of van Genuchten model on α

140
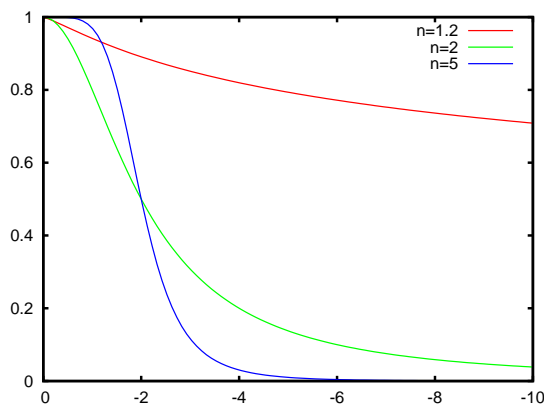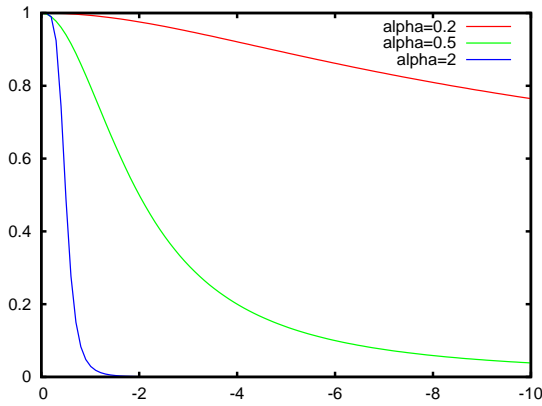
10.6.2 Hydraulic Conductivity Function

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 we know the pore size distribution than we can calculate the hydraulic conductivity from

$$K(S_{\text{eff}}) = K_{\text{sat}} S_{\text{eff}}^{\tau} \frac{\int_0^{S_{\text{eff}}} \frac{1}{h^2} dS}{\int_0^1 \frac{1}{h^2} dS}$$

The term S_{eff}^{τ} should take the tortuosity of the pores into account. τ is a dimensionless fitting parameter.

The Burdine model is usually used together with the Brooks-Corey model. The resulting function is:

$$K(S_{\text{eff}}) = K_{\text{sat}} S_{\text{eff}}^{\tau+1+2/\lambda}$$

where usually $\tau = 2$ is used.

Dependency of Brooks-Corey Burdine model on λ
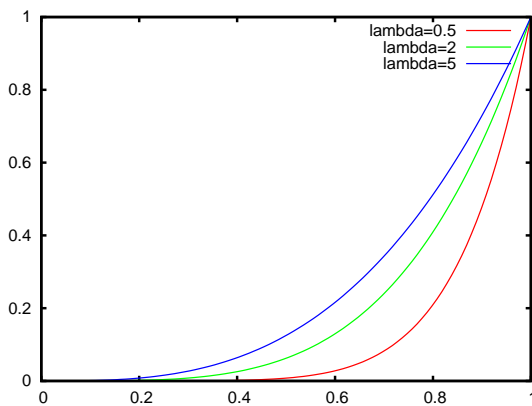
Mualem Model

Mualem proposed in 1976 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 that then

the square could be taken out of the integral to obtain

$$K(S_{\mathsf{eff}}) = K_{\mathsf{sat}} S_{\mathsf{eff}}^{\tau} \left[\frac{\int_0^{S_{\mathsf{eff}}} \frac{1}{h} dS}{\int_0^1 \frac{1}{h} dS} \right]^2$$

For the Brooks-Corey model Mualem obtained

$$K(S_{\mathsf{eff}}) = K_{\mathsf{sat}} S_{\mathsf{eff}}^{\tau+2+2/\lambda}$$

Mualem argued that one obtained for many soils good agreement with $\tau = 0.5$. Thus the Brooks-Corey Mualem model yields an exponent of $2.5 + 2/\lambda$ in contrast to $3 + 2/\lambda$ obtained by the Brooks-Corey Burdine model.

Comparison of Brooks-Corey Burdine and Mualem model
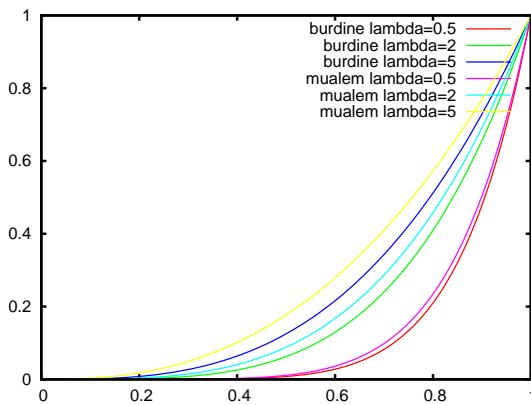
van Genuchten-Mualem Model

$$S_e = [1 + (\alpha h)^n]^{-m}$$

van Genuchten solved:

$$\int\limits_0^{S_e} \frac{1}{h(S)} dS.$$

for $m = i - 1/n$, where usually $i = 1$ is used with the result:

$$\int\limits_0^{S_e} \frac{1}{h(S)} dS = 1 - (1 - S_e^{1/m})^m \quad (m = 1 - 1/n).$$

$$K_r(S_e) = S_e^{\tau} \cdot \left[1 - \left(1 - S_e^{1/m} \right)^m \right]^2.$$

Validity Limits for the van Genuchten-Mualem Model

For the Mualem model one has to evaluate the integral

$$\int_0^{S_e} \frac{1}{h(S)} dS = -\int_{h(S_e)}^{\infty} \frac{1}{h} \frac{dS}{dh} dh.$$

According to the Young-Laplace-Equation:

$$h = \frac{p_c}{\rho_w g} = \frac{1}{\rho_w g} \frac{2\sigma_w \cos(\theta)}{r},$$

where σ_w is the surface tension of water, θ the contact angle and r is the pore radius.

⇒ arbitrary large pores can dominate the integral.

Pores of arbitrary large radius are only excluded if the derivative $\frac{dS}{dh}$ goes faster to zero than $1/h$. As

$$\frac{dS}{dh} = -\alpha m n \left(\alpha h\right)^{n-1} \left[1 + \left(\alpha h\right)^n\right]^{-(m+1)},$$

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

The maximum of $\frac{dS}{dh}$ 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. $\Rightarrow \alpha^{-1} \gg h_e \Leftrightarrow \alpha h_e \ll 1$

Dependency of van Genuchten Mualem model on n

143
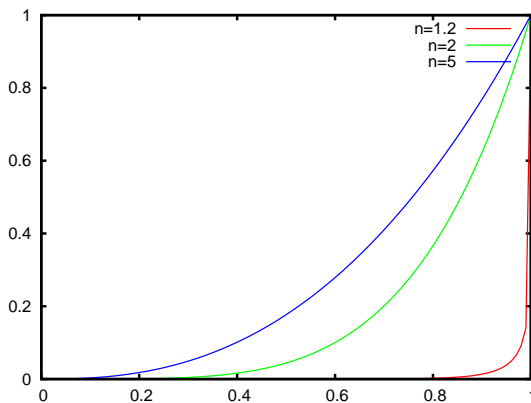
Dependency of van Genuchten Mualem model on τ
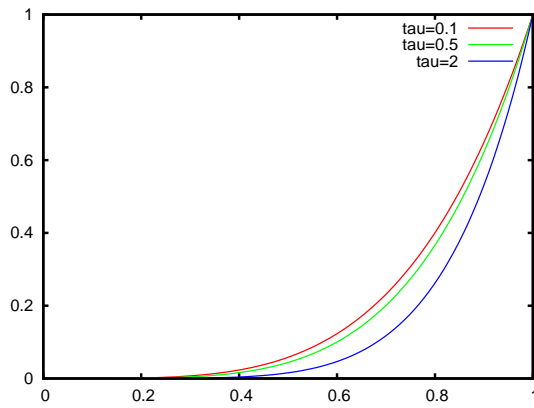
van Genuchten-Mualem Model with Entry Pressure

$$S_e = \begin{cases} [1 + (\alpha(h - h_e))^n]^{-m} & h > h_e \\ 1 & h \leq h_e \end{cases}.$$

Makes solution of integral

$$\int_0^{S_e} \frac{1}{\left(S^{-1/m} - 1\right)^{1/n} + \alpha h_e} dS.$$

necessary

Modified van Genuchten-Mualem Model (T. Vogel+2001)

$$S_e = \begin{cases} \frac{1}{S_e^*} \cdot [1 + (\alpha h)^n]^{-m} & h > h_e \\ 1 & h \leq h_e \end{cases}, \quad S_e^* = [1 + (\alpha h_e)^n]^{-m}$$

$$K_r = \begin{cases} S_e^\tau \cdot \left[\frac{1 - \left(1 - (S_e S_e^*)^{1/m}\right)^m}{1 - \left(1 - S_e^{*1/m}\right)^m} \right]^2 & S_e < S_e^* \\ 1 & S_e \geq S_e^* \end{cases}$$

144

Modified van Genuchten-Mualem and Brooks-Corey Model

If $\alpha h_e \gg 1$ then

$$[1 + (\alpha h)^n]^{-m} \approx (\alpha h)^{-mn} \quad \text{for} \quad h > h_e$$

and

$$S_e \approx \frac{(\alpha h)^{-mn}}{(\alpha h_e)^{-mn}} = \left(\frac{h}{h_e}\right)^{-mn}.$$

$\Rightarrow$ The modified van Genuchten-Mualem model converges to the Brooks-Corey model for $\alpha h_e \gg 1$

10.7 Numerical Solution

Numerical Solution of Richards' Equation

The mixed form of Richards equation together with a suitable parametrisation forms a non-linear partial differential equation. Steps to the solution are

1. discretisation in space (e.g. with cell-centred Finite-Volume scheme)

2. discretisation in time (e.g. implicit Euler scheme)

3. linearisation of the non-linear equation system (with Newton or Fixpoint Iteration)

4. solution of the resulting linear equation system

Discretisation in space and time can be done by our usual cell-centred Finite-Volume scheme and a one-step time discretisation (usually implicit Euler). If steep infiltration fronts can occur an upwinding of the relative permeability might be necessary to avoid problems due to an effective hyperbolicity of the equation.

Discretisations of the potential form have a mass balance problem, as the specific water capacity has to be determined either at the old or the new time step. Thus we will only discuss discretisations of the mixed form.

Discretized Equation

A cell-centred Finite-Volume discretisation of Richards equation with an implicit Euler scheme in time for a equidistant grid yields (one-dimensional):

$$
h \left(\theta(\psi_i^{j+1}) - \theta(\psi_i^j) \right)
$$
$$
- \frac{\tau}{h} \Big\{ K_{i-0.5}(\psi_{i-1}^{j+1}, \psi_i^{j+1}) \cdot \psi_{i-1}^{j+1} + K_{i+0.5}(\psi_i^{j+1}, \psi_{i+1}^{j+1}) \cdot \psi_{i+1}^{j+1}
$$
$$
- \left[K_{i-0.5}(\psi_{i-1}^{j+1}, \psi_i^{j+1}) + K_{i+0.5}(\psi_i^{j+1}, \psi_{i+1}^{j+1}) \right] \cdot \psi_i^{j+1}
$$
$$
- h\rho g \left[K_{i+0.5}(\psi_i^{j+1}, \psi_{i+1}^{j+1}) - K_{i-0.5}(\psi_{i-1}^{j+1}, \psi_i^{j+1}) \right] - h^2 \gamma_i^{j+1} \Big\} = 0
$$

$$
h \left(\theta(\psi_i^{j+1}) - \theta(\psi_i^j) \right) - \frac{\tau}{h} \Big\{ K_{i-0.5}(\psi_{i-1}^{j+1}, \psi_i^{j+1}) \cdot \psi_{i-1}^{j+1} + K_{i+0.5}(\psi_i^{j+1}, \psi_{i+1}^{j+1}) \cdot \psi_{i+1}^{j+1}
$$
$$
- \left[K_{i-0.5}(\psi_{i-1}^{j+1}, \psi_i^{j+1}) + K_{i+0.5}(\psi_i^{j+1}, \psi_{i+1}^{j+1}) \right] \cdot \psi_i^{j+1}
$$
$$
- h\rho_w g \left[K_{i+0.5}(\psi_i^{j+1}, \psi_{i+1}^{j+1}) - K_{i-0.5}(\psi_{i-1}^{j+1}, \psi_i^{j+1}) \right] - h^2 r_i^{j+1} \Big\} = 0
$$

10.7.1 Solution of non-linear equations

The independent variable for the solution of Richards equation in the mixed form is the potential. There are two non-linearities in the equation: the non-linear relation between water content and potential and the non-linear relation between hydraulic conductivity and potential. The two common linearisation methods are Piccard iteration and Newton iteration.

Picard Iteration

Picard iteration is based on the idea of fixpoint iteration. In Picard iteration the values of the last iteration are used to calculate $\theta(\psi_m)$ and $K(\theta(\psi_m))$. We denote the values from the last iteration with the superscript k and the values of the new iteration with the superscript $k+1$ and write $K_{i-0.5}^{j+1,k}$ instead of $K_{i-0.5}(\psi_{i-1}^{j+1,k}, \psi_i^{j+1,k})$.

$$
h \left(\theta(\psi_i^{j+1,k}) - \theta(\psi_i^j) \right)
$$
$$
- \frac{\tau}{h} \Big\{ K_{i-0.5}^{j+1,k} \cdot \psi_{i-1}^{j+1,k+1} + K_{i+0.5}^{j+1,k} \cdot \psi_{i+1}^{j+1,k+1}
$$
$$
- \left[K_{i-0.5}^{j+1,k} + K_{i+0.5}^{j+1,k} \right] \cdot \psi_i^{j+1,k+1}
$$
$$
- h\rho g \left(K_{i+0.5}^{j+1,k} - K_{i-0.5}^{j+1,k} \right) - h^2 \gamma_i^{j+1,k} \Big\} = 0
$$

Improved Picard Iteration

To retain a dependency of the water content on the current iterate Celia et al. (1990) proposed to use a first-order Taylor series development $\theta(\psi_i^{j+1}) \approx \theta(\psi_i^{j+1,k}) + \Delta\psi_i^{j+1,k+1} C_i^{j+1,k}$.

146

It is then convenient to write the equation in terms of the correction:

$$h\left(\theta(\psi_i^{j+1,k}) + \Delta\psi_i^{j+1,k+1} C_i^{j+1,k} - \theta(\psi_i^j)\right)$$

$$-\frac{\tau}{h} \cdot \left\{ K_{i-0.5}^{j+1,k} \cdot (\psi_{i-1}^{j+1,k} + \Delta\psi_{i-1}^{j+1,k+1}) + K_{i+0.5}^{j+1,k} \cdot (\psi_{i+1}^{j+1,k} + \Delta\psi_{i+1}^{j+1,k+1}) \right.$$

$$-\left[K_{i-0.5}^{j+1,k} + K_{i+0.5}^{j+1,k}\right] \cdot (\psi_i^{j+1,k} + \Delta\psi_i^{j+1,k+1})$$

$$\left. -h\rho g\left(K_{i+0.5}^{j+1,k} - K_{i-0.5}^{j+1,k}\right) - h^2\gamma_i^{j+1}\right\} = 0$$

Improved Picard Iteration

After rearranging we get

$$\left(hC_i^{j+1,k} + \frac{\tau}{h}\left[K_{i-0.5}^{j+1,k} + K_{i+0.5}^{j+1,k}\right]\right)\Delta\psi_i^{j+1,k+1}$$

$$-\frac{\tau}{h}K_{i-0.5}^{j+1,k} \cdot \Delta\psi_{i-1}^{j+1,k+1} - \frac{\tau}{h}K_{i+0.5}^{j+1,k} \cdot \Delta\psi_{i+1}^{j+1,k+1} = h\left(\theta(\psi_i^j) - \theta(\psi_i^{j+1,k})\right)$$

$$+\frac{\tau}{h} \cdot \left\{ K_{i-0.5}^{j+1,k} \cdot \psi_{i-1}^{j+1,k} + K_{i+0.5}^{j+1,k} \cdot \psi_{i+1}^{j+1,k} - \left[K_{i-0.5}^{j+1,k} + K_{i+0.5}^{j+1,k}\right] \cdot \psi_i^{j+1,k} \right.$$

$$\left. -h\rho g\left(K_{i+0.5}^{j+1,k} - K_{i-0.5}^{j+1,k}\right) - h^2\gamma_i^{j+1}\right\}$$

The resulting linear equation system is symmetric and diagonally dominant.

For the convergence of the Picard iteration it is necessary that it is a contraction, which is not guaranteed.

Newton Iteration

We can also solve the non-linear equation systems with Newton' method. We define the non-linear equations

$$f_i(\vec{\psi}) = h\left(\theta(\psi_i^{j+1}) - \theta(\psi_i^j)\right)$$

$$-\frac{\tau}{h}\left\{ K_{i-0.5}(\psi_{i-1}^{j+1}, \psi_i^{j+1}) \cdot \psi_{i-1}^{j+1} + K_{i+0.5}(\psi_i^{j+1}, \psi_{i+1}^{j+1}) \cdot \psi_{i+1}^{j+1} \right.$$

$$-\left[K_{i-0.5}(\psi_{i-1}^{j+1}, \psi_i^{j+1}) + K_{i+0.5}(\psi_i^{j+1}, \psi_{i+1}^{j+1})\right] \cdot \psi_i^{j+1}$$

$$\left. - h\rho g\left[K_{i+0.5}(\psi_i^{j+1}, \psi_{i+1}^{j+1}) - K_{i-0.5}(\psi_{i-1}^{j+1}, \psi_i^{j+1})\right] - h^2\gamma_i^{j+1}\right\}$$

and search for the root $\mathbf{f}(\psi) = 0$

$$\mathbf{J}(\psi^k)\Delta\psi^{k+1} = \mathbf{f}(\psi^k).$$

- The Jacobian is usually not symmetric.

- In contrast to Picard iteration the convergence of Newton's method is quadratic close enough to the solution.

The right side is easy to calculate as it is only the non-linear defect. The Jacobian is more difficult. As it is hard to assemble analytically, it is much easier to assemble it by numerical differentiation.

Numerical Differentiation

Given: $\vec{\psi}^k$;
for (all elements i) **do**
   calculate f0= $f_i(\vec{\psi}^k)$;
   for (all involved nodes j) **do**
     set $t = \psi_j^k$;
     set $\delta = \epsilon(\psi_j^k + 1)$;
     set $\psi_j^k = \psi_j^k + \delta$;
     calculate fp= $f_i(\vec{\psi}^k)$;
     set $J_{ij} = $ (fp - f0)$/\delta$;
     set $\psi_j^k = t$;
   end for
  end for

A typical value for ϵ is the square root of floating point epsilon, e.g. $\epsilon = 10^{-7}$ for 14 digits precision.

Cost Effective Assembly of Jacobian

We can also write the assembling of the Jacobian a bit different. The derivative of the non-linear equation essentially consists of two parts. The derivative of the storage term $\frac{\partial \theta(\psi_i^{j+1})}{\partial \psi_i^{j+1}}$ which only exists for the center node and ends up on the diagonal, and the derivatives of the flux terms e.g. $\tau h \frac{\partial J_{i-0.5}^{j+1}}{\partial \psi_i^{j+1}}$ and $\tau h \frac{\partial J_{i-0.5}^{j+1}}{\partial \psi_{i-1}^{j+1}}$. Due to the conservation of fluxes these derivatives are added to the diagonal in line i for the derivative to ψ_i and the negative of it to the off-diagonal entry in column i in line $i-1$. To save work it is enough to calculate the derivative of the flux terms once and add them to the appropriate diagonal and off-diagonal element. Then e.g. only the left, north and top side need to be treated (and the boundary faces).

However, as the two derivatives are usually not the same if the conductivity depends non-linearly on the potential, the resulting Jacobian is not symmetric. In contrast to Piccard iteration the convergence of Newton's method is quadratic close enough to the solution.

10.7.2 Solution of linear equations

- The matrix from the Picard iteration scheme is always symmetric and thus can be solved with a preconditioned conjugated gradients scheme.

- As the Jacobian resulting from Newton's method is generally non-symmetric solvers like GMRes or BiCGStab are necessary.

- Multigrid schemes have proofed very useful.

Inexact Newton's method / Inexact Picard iteration

- At the beginning of the iteration far from the correct solution of the non-linear equation system it is not necessary to solve the linear equations very precisely

- It is enough to obtain a certain minimal reduction e.g. 10^{-3}.

- Later the required defect reduction is adapted.

- For Picard iteration it is set to the minimum of the default reduction and the non-linear reduction in the last step

- For Newton's method minimum of the default reduction and the square of the non-linear reduction in the last step is used.

10.7.3 Convergence Test

- As with linear equation systems convergence criteria which are only based on the reduction in the last time step are dangerous as they could also just indicate a poor convergence

- Good convergence tests are based on a norm of the non-linear residuum $||\mathbf{f}(\psi^k)||_2$

- A sufficient reduction of the defect is demanded. However, this is limited by floating point precision, as there is no defect formulation as in the linear case.

10.7.4 Line Search

Both linearisation schemes are only valid in a certain region around the current iterate if the functions f are strongly non-linear. Therefore both methods are not globally convergent.

The sphere of convergence can be increased, by a line search, decreasing the fraction of the correction successively until an improvement is obtained.

Given: ψ^k and $\Delta\psi^{k+1}$;
Set $\alpha = 1.0$;
while $(||f(\psi^k + \alpha\Delta\psi^{k+1})|| \geq ||f(\psi^k)||$ and $\alpha \geq 2\alpha_{\mathsf{min}})$ **do**
    $\alpha = 0.5 * \alpha$;
end while

For Newton's method it has proofed advantageous to demand not only that there is a reduction, but that the reduction of the defect in the current step is smaller smaller than $1. - 0.25 * \alpha$.

10.7.5 Upwinding

- As a consequence of the non-linearity of Richards' equation it can become effectively hyperbolic.

- This requires a stabilisation by upwinding.

- Algorithm:
    Given: ψ_i^k and ψ_{i-1}^k;
    Determine sign of $\frac{\psi_i^k - \psi_{i-1}^k}{h_{i-0.5}} - \rho g \mathbf{e_z}$;
    Take the potential in upwind direction as ψ_{upwind};
    Calculate $K_i(\psi_{\mathsf{upwind}})$ and $K_{i-1}(\psi_{\mathsf{upwind}})$;
    Calculate $K_{i-0.5}$ as weighted harmonic mean of $K_i(\psi_{\mathsf{upwind}})$ and $K_{i-1}(\psi_{\mathsf{upwind}})$;

10.7.6 Time Step Adaptation

- If there is still no convergence the time step can be reduced e.g. by a factor of two.

- It is harder to determine when to increase the time step again. Typical criteria are a reduction of the defect in the first iteration of Newton's method by at least a certain fraction (e.g. 0.01) or a convergence of the iteration scheme in a maximal number of steps (e.g. three). This are of course purely empirical criteria.

- It is also possible to control the time step based on an estimation of the time discretisation error (which is rarely done). A simple version of this is to demand that the maximal change of the water content in an element is below a certain limit.

10.7.7 Mass Balance

- A very valuable tool to check the correctness of the implementation is the calculation of a global mass balance.

- As the scheme is locally mass conservative it should also be globally mass conservative up to the precision of the calculations.

- To get a global mass balance it is necessary to sum the mass over all elements (which is easy for a cell-centred Finite-Volume scheme) and subtract the initial mass and the cumulative flow over all boundaries of the domain.

- The interpretation of the resulting mass balance error is complicated by the fact that the error could be analysed relative to the initial mass, the final mass or the fluxes. Thus it is best to log not only the mass balance error but also the components of its computation.

10.8 Special Boundary Conditions

Of course the usual boundary conditions can be used (Dirichlet, Neumann). However, as long as the soil remains unsaturated and does not get completely dry, it is possible to specify Neumann boundary conditions on all boundaries, as the potential is fixed by the current potential and the water content.

10.8.1 Limited Flux Boundary Condition

- The application of pure Neumann boundary conditions for given fluxes can lead to physically unrealistic results (very high pressures) or solver breakdowns (if a soil is completely dried out by evaporation).

- An alternative are switching boundary conditions, which switch from Neumann to Dirichlet b.c. and back.

- One typical case is infiltration into a soil caused by rain. At the surface the flux is given by the rate of rainfall. The switching conditions are:
  - switch from Newton to Dirichlet if the potential at the surface gets positive.
  - switch from Dirichlet to Newton if the flux is larger then the specified flux.

  This implies the assumption, that excess infiltration goes away as surface run-off instantaneously.

- This boundary condition can also be written in a different way:

- calculate the flux using the specified potential in the Dirichlet boundary condition.

- use the minimum of the Neumann flux and the calculated flux as boundary flux.

- Similar boundary conditions can be used to simulate the lower boundary of free draining lysimeters, where the given flux and the limiting matrix potential are zero. Thus outflow can only occur if the soil is saturated.

- For evaporation a lower boundary for the potential and a flux out of the profile are given. The aim is to prevent the potential to fall below the threshold and the evaporation to be at most the prescribed rate (condensation is allowed).

10.8.2 Gravity Flow Boundary Condition

- At lower boundaries it is possible to prescribe an outflow boundary condition, where the gradient of potential is assumed to be zero and flow is only driven by the gravity term.

- However, this boundary condition can become instable if the flux inside the domain is directed to the surface.

10.9 Multiphase Flow

To describe multiphase flow in porous media we use n mass balance equations.

$$\frac{\partial \Phi S_\alpha}{\partial t} - \nabla \cdot \left[\bar{K} \frac{k_{\mathsf{rel}}(S_\alpha)}{\mu_\alpha} \left(\nabla p_\alpha - \rho_\alpha g \vec{e}_z \right) \right] + r_\alpha = 0 \tag{84}$$

$$S_\alpha = f(p_1, \ldots, p_n), \qquad \sum_{i=1}^{n} S_i = 1 \tag{85}$$

- Due to the correspondence of water potential per volume and water pressure it would be straightforward to use the pressure of the phases as independent variables However, the pressure of a phase is not defined if the phase is absent.

- Another possibility is to use the pressure of one phase (which is assumed not to vanish completely) and the saturation of the second phase. The saturation is still defined if it is zero. The only problem is that in contrast to the pressure the saturation can be discontinuous at material boundaries.

- There are other choices of primary variables and resulting formulations of multiphase flux laws.

10.10 Sample Simulations

10.10.1 Heterogeneity - Ponding

The example simulations show two different scenarios in which ponding of water occurs in a heterogeneous soil. Two materials are involved a loam and a sand (Figure 27).

The first situation is obvious. The saturated hydraulic conductivity is exceeded in the lower loamy horizon leading to ponding in the overlying sand (Figure 28).
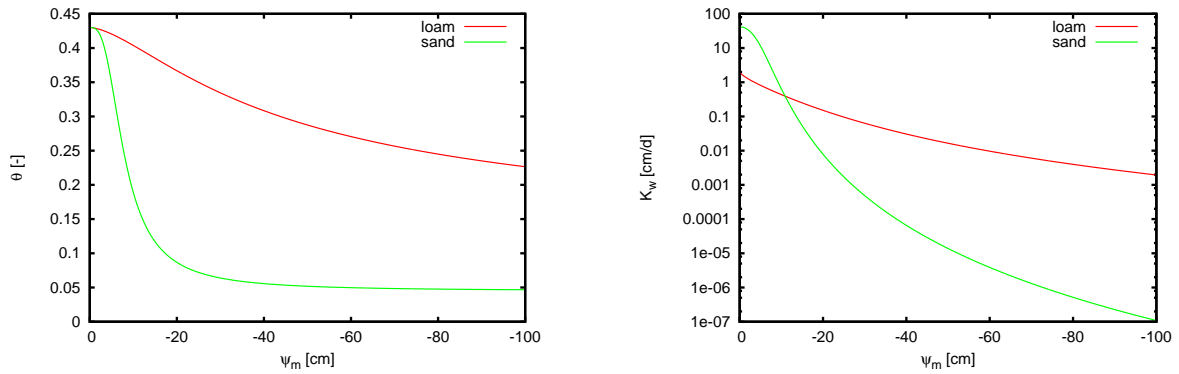
Figure 27: Soil water characteristic (left) and hydraulic conductivity function (right) for the sand and the loam.

The second situation is a bit less self-evident. The flux rate is far below the saturated hydraulic conductivity in both materials. However, the unsaturated conductivity in the underlying sand is at the same potential below the conductivity of the loam, resulting in a ponding of water until the potential is reached at which the unsaturated hydraulic conductivity is high enough (Figure 29).

10.10.2 Steep Fronts

As shown above Richards' equation can get effectively hyperbolic at steep infiltration fronts. This is illustrated by two examples.

The first example (Figure 30) shows the development of water content and potential during a constant rate infiltration in a homogeneous sand. The elements for which the sign condition in the jacobian is violated is always directly at the infiltration front. While the minimal time step without upwinding is 1.5 seconds, it is 60 seconds with upwinding.

The second example (Figure 31) shows the same for a heterogeneous sand packing. The infiltration front is much more complex. However, the elements with a violated sign condition still nicely ressemble the position of the front. The difference between the minimal time step with and without upwinding is even larger (360 seconds compared to 4.3 seconds).
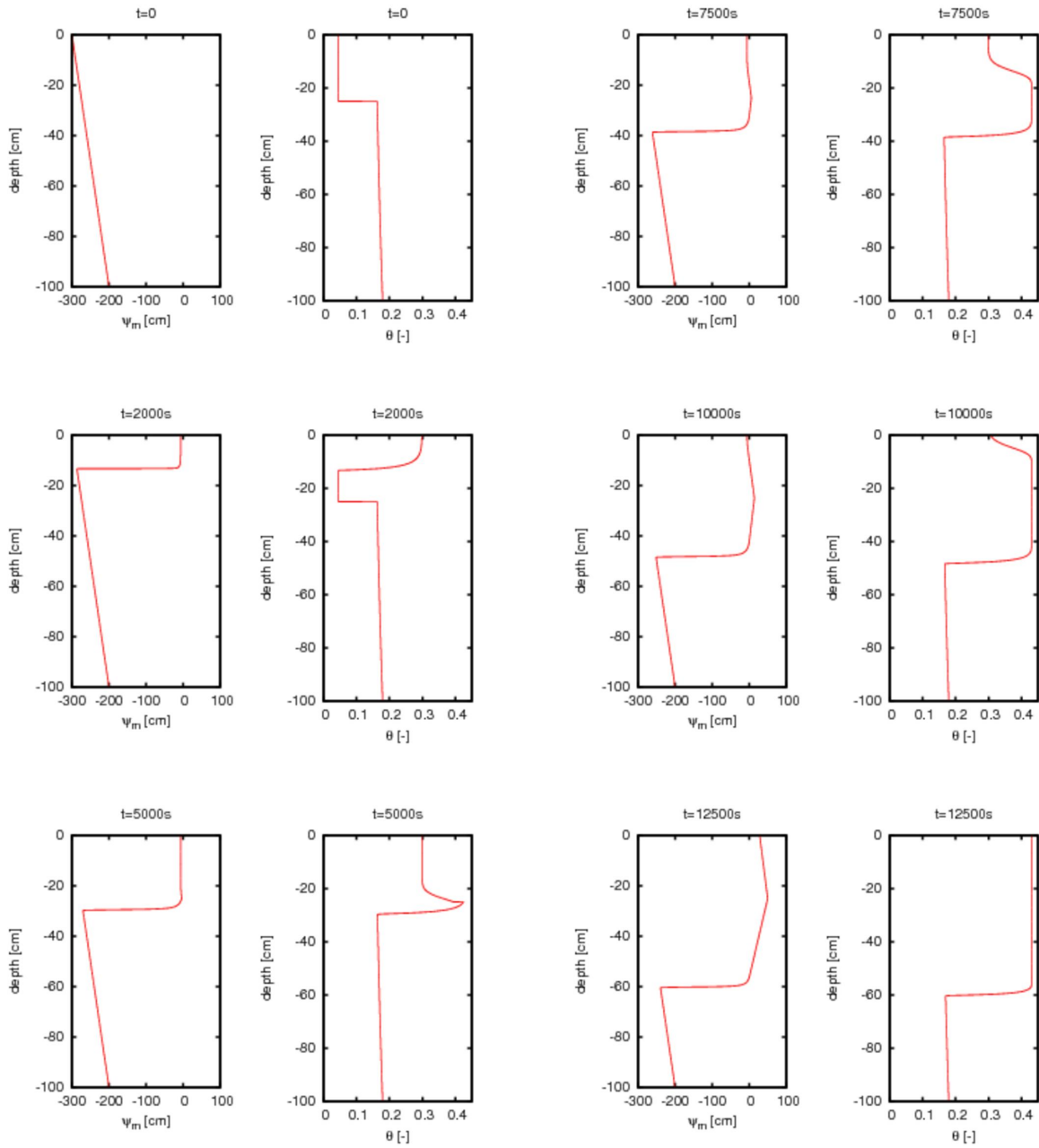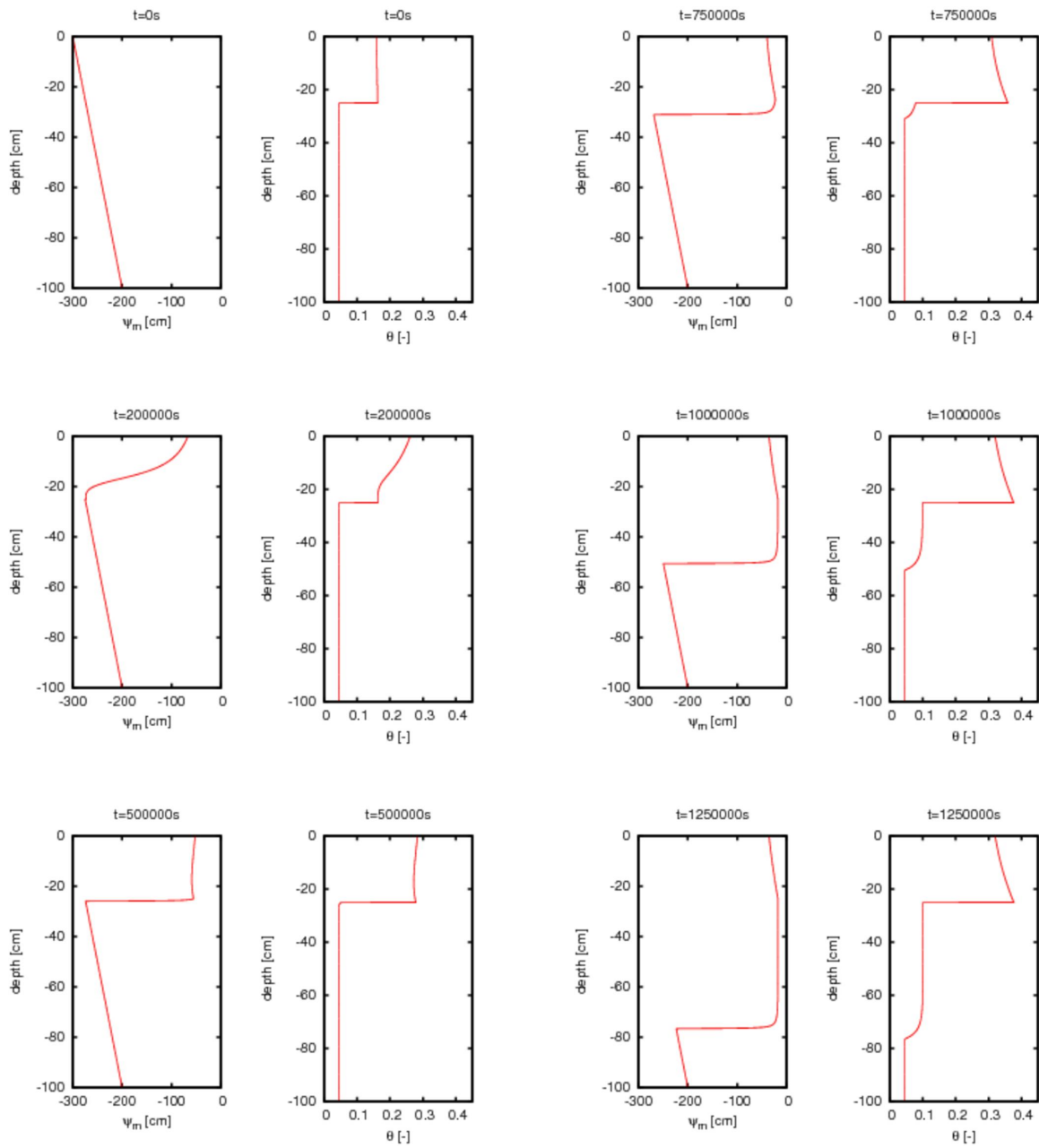
Figure 28: Infiltration in a sand over a loam

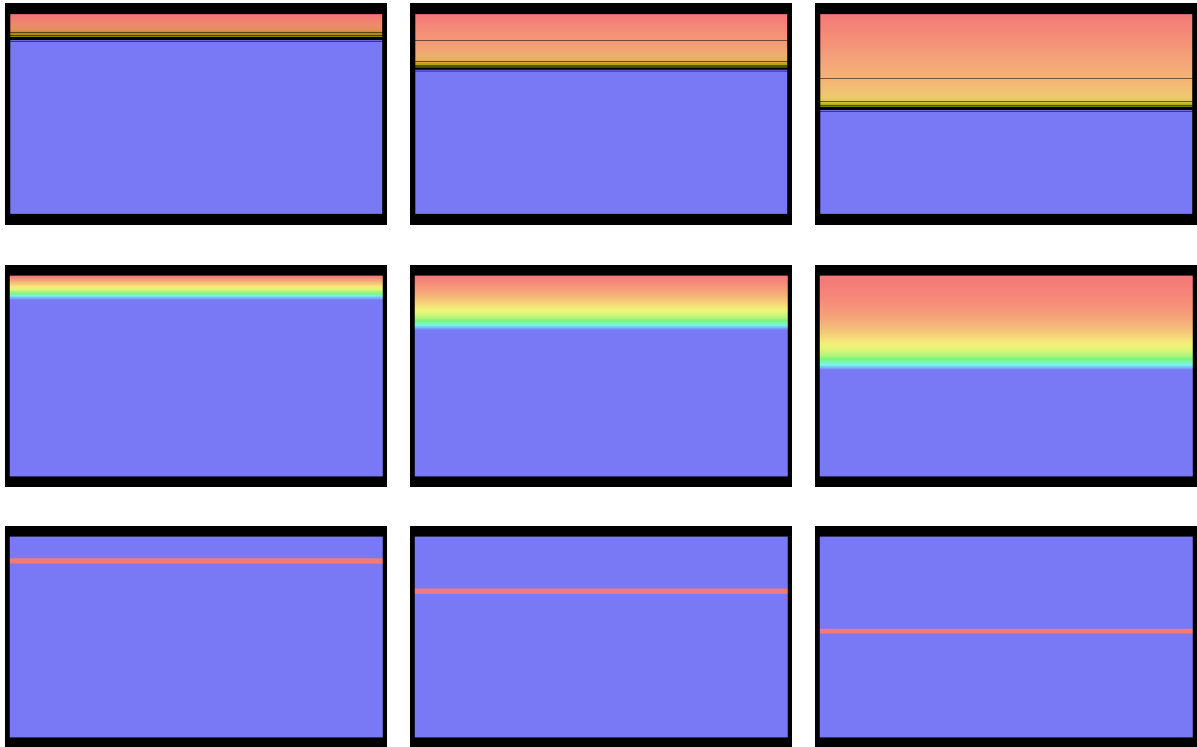Figure 29: Infiltration in a loam over a sand

Figure 30: Horizontal infiltration in a homogeneous porous medium. Potential (upper), water content (middle) and region where the sign condition is violated in the Jacobian (lower) for t=300 s, 900 s and 1800 s.
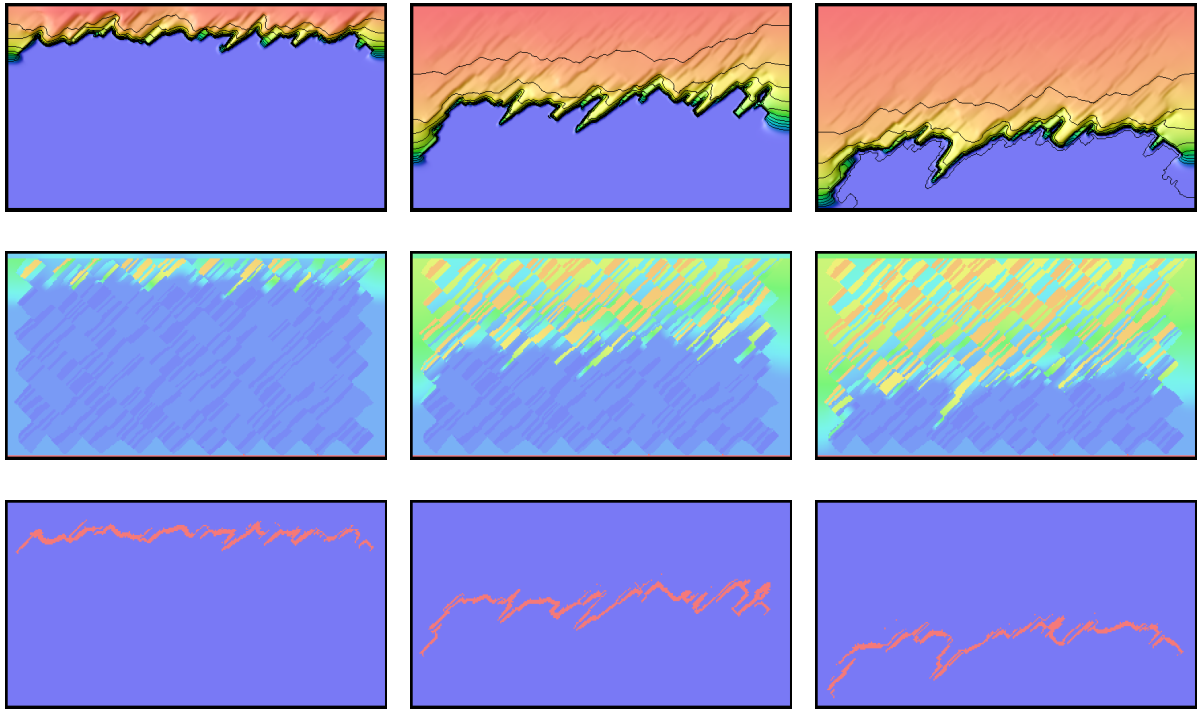
Figure 31: Horizontal infiltration in a heterogeneous porous medium. Potential (upper), water content (middle) and region where the sign condition is violated in the Jacobian (lower) for t=2000 s, 10000 s and 15000 s.

References

[Bea61]   J. Bear. On the tensor form of dispersion in porous media. *Geophysical Prospecting*, 66(4):1185–1197, 1961.

[Buc07]   E. Buckingham. Studies on the movement of soil moisture. Bulletin 38, U.S. Department of Agriculture, Bureau of Soils, Washington, DC, 1907.

[BVIV11] M. Bechtold, J. Vanderborght, O. Ippisch, and H. Vereecken. Efficient random walk particle tracking algorithm for advective dispersive transport in media with discontinuous dispersion coefficients and water contents. *Water Resour. Res.*, 2011.

[DAD05]   F. Delay, P. Ackerer, and C. Danquigny. Simulating solute transport in porous or fractured formations using random walk particle tracking: A review. *Vadose Zone Journal*, 4:360–379, 2005.

[dV52]   D. A. de Vries. The thermal conductivity of granular materials. *Annexe Bul. Inst. Intern. du Froid.*, 1992(1):115–131, 1952. copied.

[dV63]   D. A. de Vries. Thermal properties of soils. In W. R. van Wijk, editor, *Physics of Plant Environment*, pages 210–235. North Holland, Amsterdam, 1963. copied.

[ICR98]   O. Ippisch, I. Cousin, and K. Roth. Wärmeleitung in porösen Medien - Auswirkungen der Bodenstruktur auf Wärmeleitung und Temperaturverteilung. *Mitt. Dtsch. Bodenkundl. Ges.*, 87:405–408, 1998.

[Lev02]   R. J. Leveque. *Finite Volume Methods for Hyperbolic Problems*. Cambridge University Press, 2002.

[LFT96]   E. M. LaBolle, G. E. Fogg, and A. F. B. Tompson. Random-walk simulation of transport in heterogeneous porous media: Local mass-conservation problem and implementation methods. *Water Resour. Res.*, 32(3):583–593, 1996.

[Lim06]   D. H. Lim. Numerical study of nuclide migration in a nonuniform horizontal flow field of a high-level radioactive waste repository with multiple canisters. *Nucl. Technol.*, 156(2):222–245, 2006.

[Mil59]   R. J. Millington. Gas diffusion in porous media. *Science*, 130:100–102, 1959.

[MQ61]   R. J. Millington and J. P. Quirk. Permeability of porous solids. *Trans. Faraday. Soc.*, 57:1200–1207, 1961.

[Ran06]   R. Rannacher. Einführung in die Numerische Mathematik (Numerik 0). `http://numerik.iwr.uni-heidelberg.de/~lehre/notes`, 2006.

[Ric31]   L. A. Richards. Capillary conduction of liquids through porous mediums. *Physics*, 1:318–333, 1931.

[SAM93]   K. Semra, P. Ackerer, and R. Mose. Three dimensional ground-water quality modeling in heterogeneous media. In L. C. Wrobel and C. A. Brebbia, editors, *Water pollution II: Modeling, measuring and prediction*, pages 3–11. Computational Mechanics Publications, Southampton, UK, 1993.

[Sch61]   A. E. Scheidegger. General theory of dispersion in porous media. *Geophysical Prospecting*, 66(10):3273–3278, 1961.

[SFGGH06] P. Salamon, D. Fernandez-Garcia, and J.J. Gomez-Hernandez. A review and numerical assessment of the random walk particle tracking method. *J. Contam. Hydrol.*, 87(3-4):277–305, 2006.

[Uff85]   G. J. M. Uffink. A random-walk method for the simulation of macrodispersion in a stratified aquifer. In *IUGG 18th general assembly*, volume 65 of *IAHS symposia*, pages 26–34, 1985.