Streamline methods for parabolic differential equations.

Master of Science Thesis in Applied Mathematics

Jørn Hafver

Department of Mathematics
University of Bergen

August 11, 2008

Preface

First of all thanks to my inspiring supervisor Jan Martin Nordbotten for many a good cup of coffee at Knøderen, sometimes even accompanied by scientific discussions. To all my friends and fellow students at the institute, thank you for these two years. I hope to see you all again. Please give me a call if you end up in Jæren milking cows.

Contents

Objective

Parabolic advection-diffusion equations arise when modelling flow in porous media. We will in this thesis discuss two different problem set-ups from which these types of equations arise.

- Groundwater contamination with diffusion/dispersion
- Fractional-flow formulation of immiscible two-phase flow

Streamline methods equipped with time-of-flight coordinates are attractive alternatives or supplements to traditional solution methods of advection-diffusion equations. This is particulary the case when cross-streamline diffusive effects can be neglected. In this case the possibly 3-dimensional equations can be reduced to 1-dimensional equations along the streamlines. If cross-streamline effects need to be taken into account, these effects can be simulated on background grids through mappings which introduce significant numerical diffusion.

We propose a method to take care of the cross-streamline diffusive effects along normallines in 2D. It is based on operator splitting, reducing the 2D-equations to 1-dimensional equations along streamlines and normallines.

Outline

In chapter 1 we derive the basic equations for one-phase and two-phase porous media flow. For the case of two-phase flow, we vertically average the equations, reducing them from 3D to 2D in chapter 2. Streamlines, normallines and time-of-flight coordinates are introduced in chapter 3, and the spatial derivatives in the advection-diffusion equations are replaced by time-of-flight coordinates along streamlines and normallines.

Chapter 4 gives a brief introduction to some of the methods available for solving these types of equations. The drawback of conventional streamline methods when it comes to cross-streamline effects is mentioned.

In chapter 5 we introduce a method based on dimensional operator splitting of the time-of-flight equations into parts along streamlines and parts along normallines, and introduce an operator splitting coupling based on scaling.

In chapter 6 the vertically averaged two-phase equation is solved on streamline-normalline grids for CO_2 injection in a confined aquifer, and the solutions are compared to a reference solution. We compare the resulting errors for a number of different setups, varying streamline/normalline density, refinement and global time steps to identify the errors associated with the splitting and coupling.

In chapter 7 we consider a confined aquifer with background flow, an injection well that injects polluted water, and a water production well. The method of mapping the solution onto a background grid to handle the dispersion is here compared to the method of taking care of the dispersive terms along normallines.

Chapter 8 is a summary chapter, while Appendix A gives an introduction to the terminology and notation throughout this thesis. Appendix B outlines Pollock's streamline tracing method on Cartesian grids, and Appendix C explains and outlines the implementation of some of the built-in Matlab codes applied.

Chapter 1

Introduction

In this chapter we give an introduction to the basic equations of porous media flow. Both one-phase and two-phase equations are derived, along with the *IMPES* formulation of two-phase flow.

1.1 Conservation Laws

The principle of mass conservation is central in any field of flow dynamics. Figure 1.1 shows a fixed reference volume Ω. The rate of change of total mass inside this volume needs to be balanced by the rate of mass entering the volume through the edge $\partial\Omega$ and the rate of mass added/removed by possible sources or sinks.

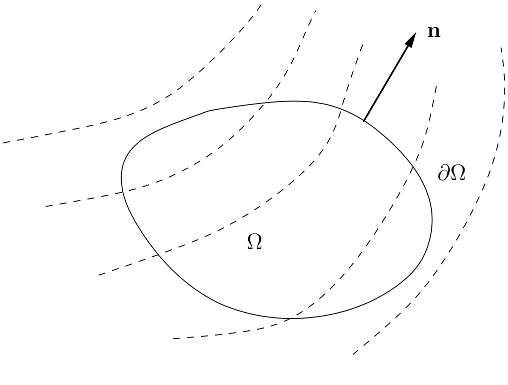


Figure 1.1: The figure shows a fixed reference volume in 2D-space.

$$\begin{bmatrix}\text{Rate of change}\\ \text{inside }\Omega\end{bmatrix} = \begin{bmatrix}\text{Mass entering}\\ \text{per time unit}\end{bmatrix} + \begin{bmatrix}\text{Mass being generated}\\ \text{per time unit}\end{bmatrix}$$

Mathematically speaking, the following expresses conservation of some quantity c in the area Ω:

$$\frac{\partial}{\partial t}\int_{\Omega} c\mathrm{d}x + \oint_{\partial\Omega} \mathbf{F}\cdot\mathbf{n}\mathrm{d}s = \int_{\Omega} Q\mathrm{d}x \tag{1.1}$$

where $\mathbf{F}$ is the boundary flux, $\mathbf{n}$ is the outward unit normal vector, and Q is a source term. The theorem of Gauss yields

$$\int_{\Omega}\left(\frac{\partial}{\partial t}c + \nabla\cdot\mathbf{F}\right)\mathrm{d}x = \int_{\Omega} Q\mathrm{d}x \tag{1.2}$$

This is the integral form of the continuity equation. The area Ω is arbitrary such that

$$\frac{\partial}{\partial t}c + \nabla\cdot\mathbf{F} = Q \tag{1.3}$$

This is the differential form of the continuity equation. It's complexity highly depends on the structure of the flux function $\mathbf{F}$, the source term Q, and possible nonlinear coupling between the velocity $\mathbf{v}$ and c. Equation (1.3) is only valid if the terms involved are sufficiently smooth. If they are not, equation (1.2) is still valid.

1.2 Basics of Flow in Porous Media

The fluids trapped in reservoirs can appear in multiple phases and each phase can consist of several components. Components may even transfer between phases when state variables change. For instance, gas may boil out of the oil phase if the pressure drops, however there is usually not much miscibility between oil and water. It is also known that CO_2 in contact with water over a long time and under high pressure can produce minerals. For those new to reservoir mechanics, Appendix A gives an overview and a brief introduction to the common terminology and notation used in this thesis.

We will only consider flow in *confined aquifers*. An aquifer is defined in the book of Freeze and Cherry [8] as "a saturated permeable geologic unit that can transmit significant quantities of water under ordinary hydraulic gradients". Hydraulic gradient is groundwater terminology for pressure gradients. A confined aquifer is an aquifer which is confined between two layers of impermeable mass, for instance clay.

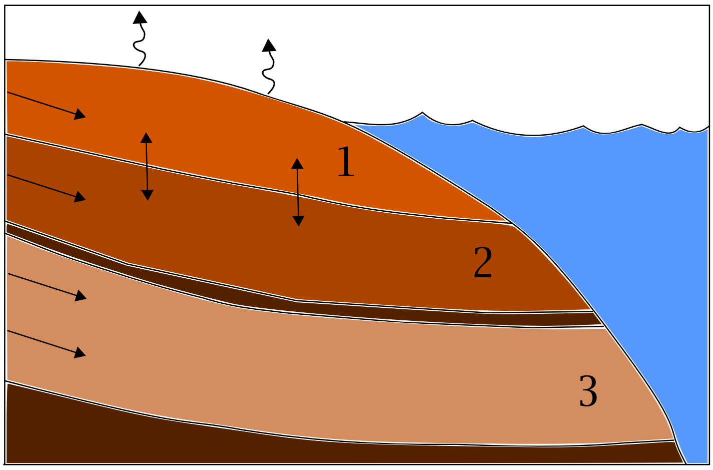


Figure 1.2: Figure illustrating how groundwater movement is affected by geological layers. Water is entering layer 1 through rivers or rain, and leaving it through evaporation. Water may transfer between layer 1 and layer 2. There may be layers of impermeable rock so that water cannot transfer between layer 2 and layer 3. Layer 3 is a confined aquifer.

1.2.1 Darcy's Law

Darcy's law was originally considered an empirical result, according to the book of Muskat [15] first stated in its simplest form by Henry Darcy in 1856. It is an expression for the average fluid flow rate in a point. When talking about points in porous media, we mean a small reference volume large enough to capture a representative amount of pores. Later it has been shown that Darcy's law follows from the Navier-Stokes equation, see for instance Freeze and Cherry [8]. The Darcy velocity is a phase variable, and can be expressed as

$$\mathbf{v}_\alpha = -\mathbf{K}\lambda_\alpha(\nabla p_\alpha + \gamma_\alpha \mathbf{k}) \tag{1.4}$$

Here λ_α is the mobility, defined as $\lambda_\alpha = k_{r\alpha}/\mu_\alpha$, and γ_α is defined as $\gamma = -\rho_\alpha g$.

1.3 Groundwater Contamination

In this section we consider the spreading of some contaminant dissolved in groundwater, a one-phase, two-component problem. The concentration of the contaminant is given by the amount of contaminant mass per unit volume of water. The concentration c is a conserved quantity, and we can use the conservation law derived in the previous section. The flux function $\mathbf{F}$ will in our setup consist of a purely convective part, but also a dispersive part,

$$\mathbf{F} = \mathbf{F}_{\mathrm{c}} + \mathbf{F}_{\mathrm{d}} \tag{1.5}$$

$\mathbf{F}_{\mathrm{c}}$ follows the average linear velocity, defined as Darcy velocity divided by porosity and can be written $\mathbf{F}_{\mathrm{c}} = c\bar{\mathbf{v}}$, where $\bar{\mathbf{v}} = \mathbf{v}/\phi$ is the average linear velocity.

In general, and highly dependent on which substances are being studied, the transport of contaminants will also be determined by adsorption and radioactive decay. Adsorption is a process in which the contaminant sticks to the pores, leaving some of it behind, perhaps also changing the pore geometries significantly, depending on the contaminant and rock properties. This is what makes polymer injection for enhanced oil recovery work, by sticking to the pores and thereby allowing more oil to pass through due to capillary pressure effects. We will however neglect the effects of adsorption and radioactive decay, and rather focus on hydrodynamic dispersion.

1.3.1 Hydrodynamic Dispersion

Dispersion is important because it makes the substance of interest arrive at for instance water wells at an earlier time than if the disperson is neglected. If advection was the only process acting on the fluid, the contaminant would move like a plug. We say that hydrodynamic dispersion consists of *molecular diffusion* and *mechanical dispersion*, $\mathbf{F}_{\mathrm{d}} = \mathbf{F}_{\mathsf{diff}} + \mathbf{F}_{\mathsf{disp}}$. Molecular diffusion is due to the Brownian motion of molecules in a fluid and is usually modelled as

$$\mathbf{F}_{\mathsf{diff}} = -D\nabla c \tag{1.6}$$

where D is the diffusion coefficient. In porous media this may be a tensor, however it will be treated as a scalar in this text.

For the mechanical dispersion however, the diffusion coefficient is replaced by a dispersivity tensor. This is because the mechanical dispersion is usually greater in the direction parallel to the direction of flow than it is in the direction perpendicular to the direction of flow. According to Heimsund [10], three of the major contributing effects to mechanical dispersion are Taylor diffusion, stream splitting and tortousity effects. These effects are illustrated in figure 1.3.

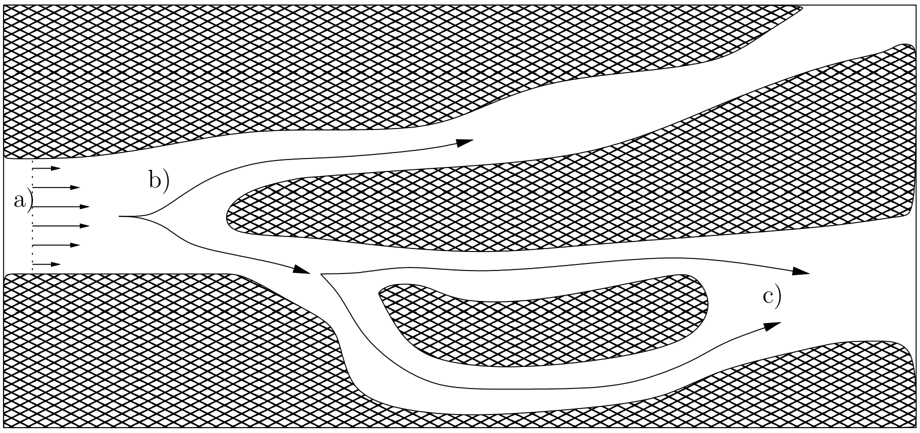


Figure 1.3: The different effects of mechanical dispersion. a) Taylor diffusion, b) stream splitting, c) tortuosity effects

A common assumption is that the dispersive flux is linearly dependent of the magnitude of average linear velocity, and is given in for instance [8] or [10]:

$$\mathbf{F}_{\text{disp}} = \|\bar{\mathbf{v}}\| \left(-d_l \frac{\partial c}{\partial l_l} \frac{\mathbf{v}}{\|\mathbf{v}\|} - d_t \frac{\partial c}{\partial l_t} \frac{\mathbf{n}}{\|\mathbf{n}\|} \right) \tag{1.7}$$

where $\partial/\partial l_l$ means differentiation in the direction parallel to and $\partial/\partial l_t$ means differentiation in the direction perpendicular to the velocity field in curvilinear coordinates. d_l and d_t are two generally different dispersion coefficients. Usually dispersion is greater in the direction of flow, and $d_l = O(10d_t)$.

Figure 1.4 shows how dispersion determines the motion of a tracer. A tracer plug in an aquifer with continuous background flow is followed from $t = t_0$ to $t = T$, and dispersion effects will make the area affected by the tracer larger (grey area), however with smaller concentrations.

1.3.2 Cartesian Coordinates

Later on we will compare our methods based on solving the equations along streamlines and normallines to methods based on dealing with the hydrodynamical dispersion on cartesian background grids. We will therefore need

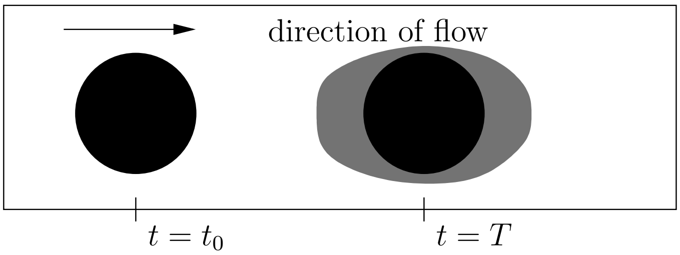


Figure 1.4: Illustrating the effects of dispersion on an amount of tracer. The black circles are the tracer at $t = t_0$, and at $t = T$ if there is no hydrodynamical dispersion.

this flux written in cartesian coordinates. Inspired by figure 1.5 define the tensors

$$\widetilde{L} = \frac{\mathbf{v}\mathbf{v}^T}{\|\mathbf{v}\|^2} \quad , \quad \widetilde{T} = \frac{\mathbf{n}\mathbf{n}^T}{\|\mathbf{n}\|^2} \tag{1.8}$$

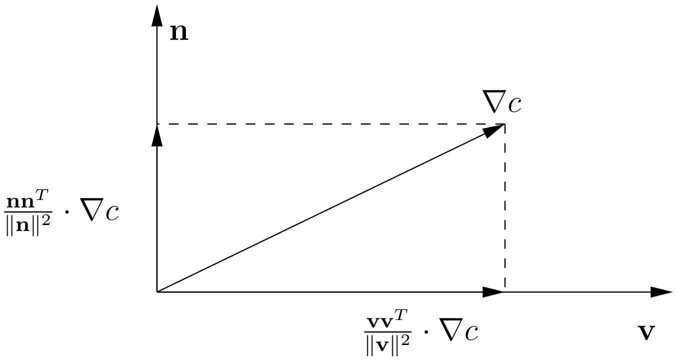

$\mathbf{n}$
∇c
$\frac{\mathbf{n}\mathbf{n}^T}{\|\mathbf{n}\|^2} \cdot \nabla c$
$\frac{\mathbf{v}\mathbf{v}^T}{\|\mathbf{v}\|^2} \cdot \nabla c$
$\mathbf{v}$

Figure 1.5: Decomposing the gradient of the concentration

The flux in equation (1.7) can then be written

$$\mathbf{F}_{\text{disp}} = \|\bar{\mathbf{v}}\| \left(-d_l \widetilde{L} \cdot \nabla c - d_t \widetilde{T} \cdot \nabla c \right) \tag{1.9}$$

Finally, define the hydrodynamic diffusion-dispersion tensor

$$\widetilde{D} = D\widetilde{I} + \|\bar{\mathbf{v}}\| \left(-d_l \widetilde{L} + d_t \widetilde{T} \right) \tag{1.10}$$

so that the total dispersive flux $\mathbf{F}_{\text{d}} = \mathbf{F}_{\text{diff}} + \mathbf{F}_{\text{disp}}$ can be written

$$\mathbf{F}_{\mathrm{d}} = -\widetilde{D} \cdot \nabla c \tag{1.11}$$

Assuming that c is continuous and putting this into the law of mass conservation gives the Cartesian form of the 2D hyperbolic-parabolic advection-dispersion equation for groundwater contamination in a homogeneous, isotropic aquifer,

$$\frac{\partial c}{\partial t} + \nabla \cdot (c\bar{\mathbf{v}}) = \nabla \cdot (\widetilde{D} \cdot \nabla c) + Q \tag{1.12}$$

Or, since $\nabla \cdot \mathbf{v} = 0$ for incompressible flow,

$$\frac{\partial c}{\partial t} + \bar{\mathbf{v}} \cdot \nabla c = \nabla \cdot (\widetilde{D} \cdot \nabla c) + Q \tag{1.13}$$

1.4 Two-Phase Flow

Advection-Diffusion equations also arise when modelling two-phase flow. In this section we consider immiscible fluids, and first introduce the standard fractional flow model. The two phases could for instance be oil and water or water and gas.

The conserved quantity in equation (1.3) is now the total mass of each phase, $\phi S_\alpha \rho_\alpha$, and the flux function $\mathbf{F}_\alpha = \rho_\alpha \mathbf{v}_\alpha$. The law of mass conservation becomes

$$\frac{\partial}{\partial t}(\phi \rho_\alpha S_\alpha) + \nabla \cdot (\rho_\alpha \mathbf{v}_\alpha) = Q_\alpha \tag{1.14}$$

We assume that the porosity is constant in time, and further that we are dealing with incompressible fluids, so that also ρ_α are constants. Equation (1.14) can then be written

$$\phi \frac{\partial S_\alpha}{\partial t} + \nabla \cdot \mathbf{v}_\alpha = \frac{Q_\alpha}{\rho_\alpha} \tag{1.15}$$

In addition to these equations, we also have Darcy's law, which gives 8 (3D) or 6 (2D) equations for the 10 (3D) or 8 (2D) unknowns $S_\alpha, \mathbf{v}_\alpha$ and p_α. To close the system, we first notice that $S_w + S_{nw} = 1$. We also need an expression that relates the two phase pressures to eachother. Here we use the capillary pressure relation $p_c = p_{nw} - p_w$.

1.4.1 The Saturation Equation

The equations (1.15) are summed over the two phases to get $\nabla \cdot (\mathbf{v}_w + \mathbf{v}_{nw}) \equiv \nabla \cdot \mathbf{v} = Q_w/\rho_w + Q_{nw}/\rho_{nw} \equiv Q$, since $S_w + S_{nw} = 1$. Here we have introduced the total velocity $\mathbf{v} = \mathbf{v}_{nw} + \mathbf{v}_w$.

Multiplying Darcy's law for each phase with the mobility of the other,

$$\lambda_{nw}\mathbf{v}_w = -\mathbf{K}\lambda_w\lambda_{nw}(\nabla p_w + \gamma_w \mathbf{k}) \tag{1.16}$$

$$\lambda_w\mathbf{v}_{nw} = -\mathbf{K}\lambda_w\lambda_{nw}(\nabla p_{nw} + \gamma_{nw} \mathbf{k}) \tag{1.17}$$

subtracting equation (1.16) from equation (1.17) gives

$$\lambda_w\mathbf{v}_{nw} - \lambda_{nw}\mathbf{v}_w = -\mathbf{K}\lambda_{nw}\lambda_w(\nabla p_c + (\gamma_{nw} - \gamma_w)\mathbf{k}) \tag{1.18}$$

We are interested in an expression that only contains the total velocity and the saturation of the wetting fluid. So we use that $\mathbf{v}_{nw} = \mathbf{v} - \mathbf{v}_w$ and get

$$\lambda_w\mathbf{v} - \mathbf{v}_w(\lambda_w + \lambda_{nw}) = -\mathbf{K}\lambda_{nw}\lambda_w(\nabla p_c + (\gamma_{nw} - \gamma_w)\mathbf{k}) \tag{1.19}$$

With the definitions $S = S_w$ and $f(S) = \lambda_w/(\lambda_w + \lambda_{nw})$, we have

$$\mathbf{v}_w = f(S)\mathbf{v} + \mathbf{K}\lambda_{nw} f(S)(\nabla p_c + (\gamma_{nw} - \gamma_w)\mathbf{k}) \tag{1.20}$$

and this we put into the continuity equation for the wetting phase:

$$\phi\frac{\partial S}{\partial t} + \nabla \cdot (f(S)\mathbf{v} + \mathbf{K}\lambda_{nw} f(S)(\nabla p_c + (\gamma_{nw} - \gamma_w)\mathbf{k})) = Q_w/\rho_w \tag{1.21}$$

This is the saturation equation.

1.4.2 The Pressure Equation

Darcy's law, (1.4) is summed over the two phases:

$$\mathbf{v} = \mathbf{v}_w + \mathbf{v}_{nw} = -\mathbf{K}\lambda_w(\nabla p_w + \gamma_w\mathbf{k}) - \mathbf{K}\lambda_{nw}(\nabla p_{nw} + \gamma_{nw}\mathbf{k}) \tag{1.22}$$

taking divergence on both sides and remembering that $\nabla \cdot \mathbf{v} = Q$,

$$\nabla \cdot \mathbf{v} = -\nabla \cdot \{\mathbf{K}\lambda_w(\nabla p_w + \gamma_w\mathbf{k}) + \mathbf{K}\lambda_{nw}(\nabla p_{nw} + \gamma_{nw}\mathbf{k})\} = Q \tag{1.23}$$

This is the pressure equation. Here as well the capillary pressure relation may be introduced, and the equation solved for p_w.

1.4.3 Solution Strategy

The pressure equation (1.22) and the saturation equation (1.21) are called the *fractional-flow* model for immiscible two-phase flow. One robust approach to solving these equations is a fully implicit method in which both equations are solved implicitly. This is however computationally expensive due to the nonlinear coupling of the equations.

Another, less expensive method is to solve the two equations sequentially. The *IMPES* method is short for *IM*plicit *P*ressure *E*xplicit *S*aturation. This way both the fundamentally different equations can be discretized by different methods.

To solve the equations sequentially means to start out with an initial saturation distribution, use this distribution to compute the saturation-dependent coefficients in the pressure equation. Then the pressure equation is solved forward one time-step for capillary pressure and velocity, and then these are supposed to be constant when solving the saturation equation the same time-step. See for instance [4] for more on this.

Chapter 2

Vertically Averaged Equations

In many reservoirs or aquifers the height of the layer is several orders of magnitude smaller than the horizontal distances. Instead of simulating the transport on a full 3D-model, for computing efficiency it could be useful to average the equations in the z-direction. This kind of averaging was widely used before the breakthrough of modern day computers [3], but is still useful if one needs efficient simulators for solving multiple, similar problems. We will in this section derive vertically averaged equations for two-phase immiscible flow in a three-dimensional domain Ω of constant thickness H.

2.1 Assumptions

With the assumption of no mixing, the fluid with the lowest mass density will be on top of the one with the highest mass density, separated by a sharp interface, given by $z(t) = h(x, y, t)$. Let the phase subscript α now denote T, B for top and bottom fluid, respectively. We assume a no-flow boundary condition at the top and bottom of $\partial\Omega$, and that the domain is infinite in the horizontal directions.
We will consider the case of CO_2-injection into confined aquifers, so that the CO_2 will displce water and stay on top of it. Residual water saturations then need to be taken into account. The saturations of the top and bottom fluid can then be written

$$S_T(x, y, z, t) = \begin{cases} 0, & 0 \leq z \leq h(x, y, t) \\ 1 - SrB, & h(x, y, t) \leq z \leq H \end{cases} \tag{2.1}$$

$$S_B(x, y, z, t) = \begin{cases} 1, & 0 \leq z \leq h(x, y, t) \\ SrB, & h(x, y, t) \leq z \leq H \end{cases} \tag{2.2}$$

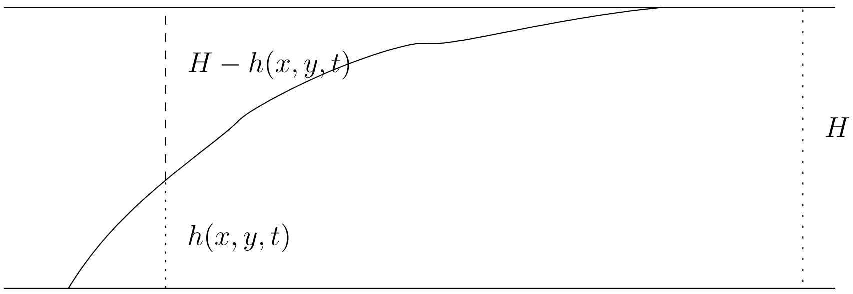


Figure 2.1: Sharp interface between the bottom and top fluid.

The continuity equation gives 2 equations for the unknowns h, S_T, S_B. Assume a pressure distribution in the z-direction on the form

$$\frac{\partial p_\alpha}{\partial z} = \rho_\alpha g \tag{2.3}$$

where g is the gravity constant. We further assume pressure continuity at the interface h, and by integration of equation (2.3) in the z-direction obtain the following relationship between the pressures and h

$$p_T(x, y, H, t) = p_B(x, y, 0, t) + (\rho_T - \rho_B)gh - \rho_T gH \tag{2.4}$$

2.2 Continuity Equation

Now we average the terms in the continuity equation for each of the fluids in the z-direction.

$$\frac{1}{H}\int_0^H \phi \frac{\partial S_\alpha}{\partial t} \mathrm{d}z + \frac{1}{H}\int_0^H \nabla \cdot \mathbf{v}_\alpha \mathrm{d}z = \frac{1}{H}\int_0^H \frac{q_\alpha}{\rho_\alpha} \mathrm{d}z \tag{2.5}$$

The goal is to write this equation in x- and y-coordinates only, with the coefficients averaged in the z-direction. Porosity is assumed independent of z.

Since everything in the first term of equation (2.5) besides S_α is assumed independent of time, the time differentiation can be moved outside of the integration. This term for the bottom and top fluids respectively, can be written

$$\frac{1}{H}\int_0^H \phi \frac{\partial S_B}{\partial t} \mathrm{d}z = \frac{\partial}{\partial t}\frac{1}{H}\int_0^H \phi S_B \mathrm{d}z = \frac{\partial}{\partial t}\frac{1}{H}\phi\left(h + S_{rB}(H - h)\right) \tag{2.6}$$

$$\frac{1}{H}\int_0^H \phi\frac{\partial S_T}{\partial t}\mathrm{d}z = \frac{\partial}{\partial t}\frac{1}{H}\int_0^H \phi S_T\mathrm{d}z = \frac{\partial}{\partial t}\frac{1}{H}\phi\left((1-S_{rB})(H-h)\right) \tag{2.7}$$

Considering figure 2.1, and equations 2.6,2.7 it is natural to define the averaged saturations for the bottom and top fluids, respectively

$$\bar{S}_B \equiv \frac{(1-S_{rB})(H-h)}{H} \tag{2.8}$$

$$\bar{S}_T \equiv \frac{h+S_{rB}(H-h)}{H} \tag{2.9}$$

This means that the first term of equation (2.5) regardless of phase can be written

$$\frac{1}{H}\int_0^H \phi\frac{\partial S_\alpha}{\partial t}\mathrm{d}z = \phi\frac{\partial \bar{S}_\alpha}{\partial t} \tag{2.10}$$

We now return to equation (2.5), and for the second term look for a $\bar{\mathbf{v}}_\alpha$ such that

$$\frac{1}{H}\int_0^H \nabla\cdot\mathbf{v}_\alpha\mathrm{d}z = \nabla\cdot\bar{\mathbf{v}}_\alpha \tag{2.11}$$

This $\bar{\mathbf{v}}_\alpha$ must not be confused with the average linear velocity $\bar{\mathbf{v}}$ introduced for one phase flow. First of all, apply Darcy's law to $\mathbf{v}_\alpha$,

$$\frac{1}{H}\int_0^H \nabla\cdot\mathbf{v}_\alpha\mathrm{d}z = \frac{1}{H}\int_0^H \nabla\cdot\left(-\frac{\mathbf{K}k_{r\alpha}}{\mu_\alpha}(\nabla p_\alpha+\gamma_\alpha\mathbf{k})\right)\mathrm{d}z \tag{2.12}$$

Equation (2.3) leads to the following simplification on the right hand side

$$\nabla p_\alpha+\gamma_\alpha\mathbf{k} = \frac{\partial p_\alpha}{\partial x}\mathbf{i}+\frac{\partial p_\alpha}{\partial y}\mathbf{j} = \nabla p'_\alpha \tag{2.13}$$

where p'_α is defined as

$$p'_T(x,y,t) = p_T(x,y,H,t) \tag{2.14}$$

$$p'_B(x,y,t) = p_B(x,y,0,t) \tag{2.15}$$

Further assume that the vertical components of the permeability tensor can be neglected, and replace the full 3-dimensional permeability tensor $\mathbf{K} = k_{ij}\mathbf{e}_i\mathbf{e}_j, \quad i,j = 1,2,3$ with the 2-dimensional version of it, $\mathbf{K}' =$

$k_{ij}\mathbf{e}_i\mathbf{e}_j, \quad i,j = 1,2$, using Einstein summation notation and letting the basis vectors be denoted $\mathbf{e}_1 = \mathbf{i}$, $\mathbf{e}_2 = \mathbf{j}$ and $\mathbf{e}_3 = \mathbf{k}$.

We will also assume that the permeability is independent of z, and we further notice that $\nabla p'_\alpha$ is independent of z since

$$\frac{\partial}{\partial z}\nabla p'_\alpha = \nabla\left(\frac{\partial p'_\alpha}{\partial z}\right) = 0 \tag{2.16}$$

Because of the assumption of no mixing, the relative permeabilities have simple expressions,

$$k_{rB} = \begin{cases} 1 & 0 \le z \le h(x,y,t) \\ 0 & h(x,y,t) \le z \le H \end{cases} \tag{2.17}$$

$$k_{rT} = \begin{cases} 0 & 0 \le z \le h(x,y,t) \\ k_{rT}(S_{rB}) & h(x,y,t) \le z \le H \end{cases} \tag{2.18}$$

Upon defining the averaged relative permeabilities

$$\bar{k}_{rB} = \frac{1}{H}\int_0^H k_{rB}\mathrm{d}z = \frac{h}{H} \tag{2.19}$$

$$\bar{k}_{rT} = \frac{1}{H}\int_0^H k_{rT}\mathrm{d}z = k_{rT}(S_{rB})\frac{H-h}{H} \tag{2.20}$$

we finally get an expression for the vertically averaged phase velocities to be used in equation (2.11),

$$\bar{\mathbf{v}}_\alpha = -\frac{\mathbf{K}'\bar{k}_{r\alpha}}{\mu_\alpha}\nabla p'_\alpha \tag{2.21}$$

Defining the averaged injection rate,

$$Q_\alpha = \frac{1}{H}\int_0^H \frac{q_\alpha}{\rho_\alpha}\mathrm{d}z \tag{2.22}$$

and combining equations (2.5), (2.10), (2.11) and (2.22) results in the vertically averaged continuity equation for each phase

$$\phi\frac{\partial \bar{S}_\alpha}{\partial t} + \nabla\cdot\bar{\mathbf{v}}_\alpha = Q_\alpha \tag{2.23}$$

2.3 IMPES formulation

Now we can combine equations (2.21) and (2.23) in a pressure equation and a saturation equation, just as in the previous section. The derivation is the same, although the capillary pressure relation is replaced by an expression linking the gradients of the phase pressures to the gradient of the water saturation. To find this expression, first take the gradient on both sides of equation (2.4).

$$\nabla p'_T = \nabla p'_B + (\rho_T - \rho_B) g \nabla h \tag{2.24}$$

Equation (2.8) can then be rearranged to give h as a function of S,

$$h = \frac{H(S - S_{rB})}{1 - S_{rB}} \tag{2.25}$$

Equation (2.24) can then be written

$$\nabla p'_T = \nabla p'_B + \gamma \nabla S \tag{2.26}$$

where

$$\gamma = \frac{H(\rho_T - \rho_B)g}{(1 - S_{rB})} \tag{2.27}$$

Now, the pressure equation can be written

$$-\nabla \cdot (\bar{\lambda}\mathbf{K}'\nabla p'_B + \gamma \bar{\lambda}_T \mathbf{K}' \nabla S) = Q \tag{2.28}$$

where $\bar{\lambda}_\alpha = \bar{k}_{r_\alpha}/\mu_\alpha$ and $\bar{\lambda} = \bar{\lambda}_B + \bar{\lambda}_T$. It is practical to express the averaged mobilities as functions of S, and equations (2.19), (2.20) and (2.25) allow us to write

$$\bar{\lambda}_B = \frac{\bar{k}_{rB}}{\mu_B} = \frac{h}{\mu_B H} = \frac{S - S_{rB}}{\mu_B(1 - S_{rB})} \tag{2.29}$$

and

$$\bar{\lambda}_T = \frac{\bar{k}_{rT}}{\mu_T} = k_{rT}(S_{rB})\frac{H - h}{\mu_T H} = k_{rT}(S_{rB})\frac{1 - S}{\mu_T(1 - S_{rB})} \tag{2.30}$$

The averaged saturation equation then becomes

$$\phi \frac{\partial S}{\partial t} + \nabla \cdot (f(S)\bar{\mathbf{v}} + g(S)\mathbf{K}'\nabla S) = Q_B \tag{2.31}$$

where

$$f(S) = \frac{\bar{\lambda}_B}{\bar{\lambda}_B + \bar{\lambda}_T} = \frac{(S - S_{rB})\mu_T}{(S - S_{rB})\mu_T + (1 - S)k_{rT}(S_{rB})\mu_B} \tag{2.32}$$

and

$$g(S) = \gamma f(S)\bar{\lambda}_T \tag{2.33}$$

The averaged total velocity $\bar{\mathbf{v}} = \bar{\mathbf{v}}_B + \bar{\mathbf{v}}_T$ must again not be confused with the average linear velocity for one-phase flow. This is the only place in the text where the bar notation is used. Later on the equations will be assumed two-dimensional from the start.

Chapter 3

Streamlines

The equations we have put up can be expressed in coordinates parallel and perpendicular to the direction of flow. This is especially effective when it comes to solving the hyperbolic parts of the equations. This part can be decoupled from a 2D-equation into multiple 1D-equations along the streamlines.

Given a velocity field $\mathbf{v}(\mathbf{x}, t)$, it's field lines at a given time are called the *streamlines* at that time. That is, streamlines are instantaneously tangent to the velocity field at every point in space.

There is no time-dependence in the velocity field since the streamlines were supposed to be instentaneously tangent to it. If the velocity field varies with time, the streamlines vary with time.

Define a curvilinear coordinate s that is the distance a particle has travelled along a streamline. The *time-of-flight* - the the time it takes a particle to travel along a streamline given the Darcy velocity $\mathbf{v}(\mathbf{x})$ is defined in for instance [19] or [12] as

$$\tau = \int \frac{\phi}{\|\mathbf{v}\|} \mathrm{d}s \tag{3.1}$$

This gives the expression

$$\frac{\|\mathbf{v}\|}{\phi} = \frac{\mathrm{d}s}{\mathrm{d}\tau} \tag{3.2}$$

such that the streamlines are given by

$$\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}\tau} = \frac{\mathbf{v}(\mathbf{x})}{\phi} \tag{3.3}$$

By the chain rule and the definition of directional derivatives we then have what makes the concept of time-of-flight-coordinates so successful,

$$\phi\frac{\partial}{\partial\tau} = \phi\frac{\mathrm{d}s}{\mathrm{d}\tau}\frac{\partial}{\partial s} = \|\mathbf{v}\|\frac{\partial}{\partial s} = \mathbf{v}\cdot\nabla \tag{3.4}$$

A similar relationship is found for ν, a time-of-flight coordinate for the normal field.

$$\phi\frac{\partial}{\partial\nu} = \mathbf{n}\cdot\nabla \tag{3.5}$$

The normal field is a vector field that is perpendicular to the velocity field in any point, $\mathbf{v}\cdot\mathbf{n} = 0$.

A *streamtube* is (in 2D) the area bounded by two streamlines. An important idea in streamline simulation is to consider each streamline as the center of a streamtube [19]. The volume of a fraction of the streamtube is known - $\Delta V = Q_t\Delta\tau$, where Q_t is the total flow rate and $\Delta\tau$ is the time needed for a particle to travel across the volume. The center streamline is then parameterized using the time-of-flight coordinate. In this way, streamlines with small τ represents fast flowing regions - streamtubes with small volumes. Large τ represents streamtubes with large volumes - slow flowing regions.

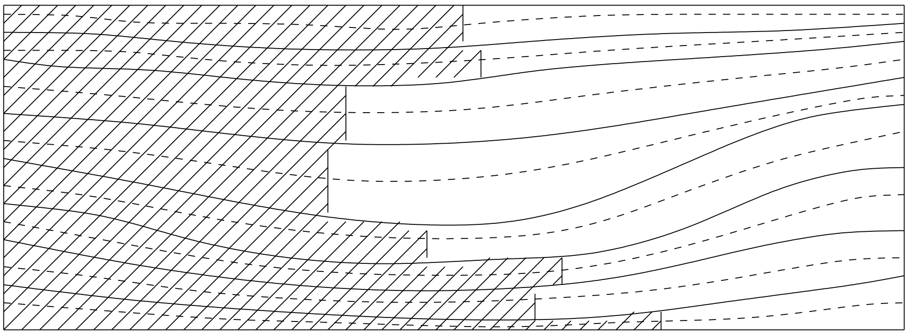


Figure 3.1: Filling a streamtube with a given volume is the same as walking along the center streamline to a known time-of-flight

Let us define the following transformations:

$$\xi : \mathbb{R}^2 \times [0,\infty) \mapsto \mathbb{R}^2 \tag{3.6}$$

$$\eta : \mathbb{R}^2 \times [0,\infty) \mapsto \mathbb{R}^2 \tag{3.7}$$

Such that $\xi(\mathbf{x}_0,\tau)$ is the parameterization of a streamline starting in the point $\mathbf{x}_0$, and $\eta(\mathbf{x}_1,\nu)$ is the parameterization of a normalline starting in the point $\mathbf{x}_1$. This notation will be used throughout this thesis.

3.1 The Convection-dispersion Equation in Time-of-flight Coordinates

We get the equation for convective/dispersive groundwater flow in time-of-flight coordinates if we insert the advective, diffusive and dispersive fluxes into the continuity equation and replace the directional derivatives with time-of-flight derivatives according to equation (3.4),

$$\begin{aligned}\frac{\partial c}{\partial t} + \nabla \cdot (c\bar{\mathbf{v}}) - \nabla \cdot \Bigg\{ \Bigg((\|\bar{\mathbf{v}}\| d_l + \phi D) \frac{\phi}{\|\mathbf{v}\|} \frac{\partial c}{\partial \tau} \frac{\mathbf{v}}{\|\mathbf{v}\|} \Bigg) \qquad (3.8)\\ + \Bigg((\|\bar{\mathbf{v}}\| d_t + \phi D) \frac{\phi}{\|\mathbf{n}\|} \frac{\partial c}{\partial \nu} \frac{\mathbf{n}}{\|\mathbf{n}\|} \Bigg) \Bigg\} = Q\end{aligned}$$

For an arbitrary scalar function h, the following relationships hold

$$\nabla \cdot (h\mathbf{v}) = \mathbf{v} \cdot \nabla h + h \nabla \cdot \mathbf{v} = \frac{\partial h}{\partial \tau} \qquad (3.9)$$

$$\nabla \cdot (h\mathbf{n}) = \mathbf{n} \cdot \nabla h + h \nabla \cdot \mathbf{n} = \frac{\partial h}{\partial \nu} + h \nabla \cdot \mathbf{n} \qquad (3.10)$$

since $\nabla \cdot \mathbf{v} = 0$ for incompressible flow, but $\nabla \cdot \mathbf{n}$ is not necessarily zero.

Equation (3.8) can then be written

$$\begin{aligned}\frac{\partial c}{\partial t} + \frac{\partial c}{\partial \tau} - \phi \frac{\partial}{\partial \tau} \Bigg((\|\bar{\mathbf{v}}\| d_l + \phi D) \frac{\phi}{\|\mathbf{v}\|^2} \frac{\partial c}{\partial \tau} \Bigg) \qquad (3.11)\\ -\phi \frac{\partial}{\partial \nu} \Bigg((\|\bar{\mathbf{v}}\| d_t + \phi D) \frac{\phi}{\|\mathbf{n}\|^2} \frac{\partial c}{\partial \nu} \Bigg) - (\|\mathbf{v}\| d_t + \phi D) \frac{\phi}{\|\mathbf{n}\|^2} \frac{\partial c}{\partial \nu} \nabla \cdot \mathbf{n} = Q\end{aligned}$$

Note that this equation is on conservation form and is reduced to a 1D-problem along the streamlines if dispersion and diffusion effects are neglected. Furthermore, if $\nabla \cdot \mathbf{n} = 0$, the equation is on conservation form in (τ, ν)-space.

3.2 The Saturation Equation in Time-of-flight Coordinates

The same can be done with the averaged saturation equation,

$$\phi \frac{\partial S}{\partial t} + \nabla \cdot (f(S)\mathbf{v}) + \nabla \cdot (g(S)\mathbf{K}\nabla S) = Q \qquad (3.12)$$

The second term can as in the previous section be written

$$\nabla \cdot (f(S)\mathbf{v}) = \phi \frac{\partial f(S)}{\partial \tau} \tag{3.13}$$

The gradient of the saturation can also be decomposed:

$$\nabla S = \frac{\nabla S \cdot \mathbf{v}}{\|\mathbf{v}\|^2}\mathbf{v} + \frac{\nabla S \cdot \mathbf{n}}{\|\mathbf{n}\|^2}\mathbf{n} = \frac{\phi}{\|\mathbf{v}\|^2}\frac{\partial S}{\partial \tau}\mathbf{v} + \frac{\phi}{\|\mathbf{n}\|^2}\frac{\partial S}{\partial \nu}\mathbf{n} \tag{3.14}$$

We further assume that the permeability tensor is isotropic: $\mathbf{K} = k(\mathbf{x})$. With $\tilde{g}(S, \mathbf{x}) \equiv k(\mathbf{x})g(S)$, we can now write the third term in (3.12) as follows:

$$\begin{aligned}
&\nabla \cdot (\tilde{g}(S,\mathbf{x})\nabla S) = \\
&\nabla \cdot \left(\mathbf{v}\frac{\tilde{g}(S,\mathbf{x})\phi}{\|\mathbf{v}\|^2}\frac{\partial S}{\partial \tau}\right) + \nabla \cdot \left(\mathbf{n}\frac{\tilde{g}(S,\mathbf{x})\phi}{\|\mathbf{n}\|^2}\frac{\partial S}{\partial \nu}\right) = \\
&\phi\frac{\partial}{\partial \tau}\left(\frac{\tilde{g}(S,\mathbf{x})\phi}{\|\mathbf{v}\|^2}\frac{\partial S}{\partial \tau}\right) + \phi\frac{\partial}{\partial \nu}\left(\frac{\tilde{g}(S,\mathbf{x})\phi}{\|\mathbf{n}\|^2}\frac{\partial S}{\partial \nu}\right) + \frac{\tilde{g}(S,\mathbf{x})\phi}{\|\mathbf{n}\|^2}\frac{\partial S}{\partial \nu}\nabla \cdot \mathbf{n}
\end{aligned} \tag{3.15}$$

Now we can re-write equation (3.12):

$$\begin{aligned}
&\frac{\partial S}{\partial t} + \frac{\partial f(S)}{\partial \tau} + \frac{\partial}{\partial \tau}\left(\frac{\tilde{g}(S,\mathbf{x})\phi}{\|\mathbf{v}\|^2}\frac{\partial S}{\partial \tau}\right) \\
&+ \frac{\partial}{\partial \nu}\left(\frac{\tilde{g}(S,\mathbf{x})\phi}{\|\mathbf{n}\|^2}\frac{\partial S}{\partial \nu}\right) + \frac{\tilde{g}(S,\mathbf{x})\phi}{\|\mathbf{n}\|^2}\frac{\partial S}{\partial \nu}\nabla \cdot \mathbf{n} = \frac{Q}{\phi}
\end{aligned} \tag{3.16}$$

Note that this equation is two point *degenerate*, because the diffusive flux function g vanishes when $S = 0$ and $S = 1$. In this case there is no diffusion, and we should expect to lose the smoothness of the solution at the saturation endpoints.

3.3 Divergence Free Normal Field Obtained by Scaling

As mentioned, our equations can be written on conservation form if $\nabla \cdot \mathbf{n} = 0$. A simple way to define $\mathbf{n}$ is to let $\mathbf{n} = R \cdot \mathbf{v}$, where R is the 90 degree counter-clockwise rotation matrix,

$$R = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \tag{3.17}$$

There is no reason to choose this definition of $\mathbf{n}$, except for its simplicity. Instead, define a scaling function ψ such that

$$\mathbf{n}^* = \psi R\mathbf{v} = \psi\mathbf{n} \tag{3.18}$$

such that

$$\begin{aligned}\nabla \cdot \mathbf{n}^* &= \nabla \cdot (\psi \mathbf{n}) \\ &= \mathbf{n} \cdot \nabla \psi + \psi \nabla \cdot \mathbf{n} = 0 \qquad (3.19)\end{aligned}$$

According to equation (3.5), this first order PDE can be reduced to a first order ODE along the normallines.

$$\phi \frac{\partial \psi}{\partial \nu} + \psi \nabla \cdot \mathbf{n} = 0 \qquad (3.20)$$

with solution

$$\psi = Ce^{-\int_o^\nu \nabla \cdot \mathbf{n}/\phi \mathrm{d}x} \qquad (3.21)$$

Using (3.18), and solving equation (3.20) will give us a divergence free normal field.

If $\nabla \times \mathbf{v} = 0$ and $\mathbf{n}^* = \begin{pmatrix} -v_2 \\ v_1 \end{pmatrix}$, then $\nabla \cdot \mathbf{n}^* = 0$ since $\nabla \times \mathbf{v} = \left(\frac{\partial v_2}{\partial x} - \frac{\partial v_1}{\partial y} \right) \mathbf{k} = -(\nabla \cdot \left(\frac{\mathbf{n}^*}{\psi} \right))\mathbf{k}$. This is the case with one-phase flow of a fluid with constant viscosity in a homogeneous, isotropic porous medium, since then there is no relative permeability. Then the Darcy velocity is a gradient, and hence has zero curl.

Chapter 4

Solution Methods

The averaged saturation equation for two-phase, immiscible flow, and the groundwater transport equation are both parabolic advection/diffusion equations. The first one is nonlinearly coupled to the pressure equation, and is in itself nonlinear. Furthermore, it is degenerate. The second one is linear, and not coupled to other equations.

For simplicity of notation, consider the two-phase averaged saturation equation

$$\phi \frac{\partial S}{\partial t} + \nabla \cdot (\mathbf{v} f(S)) + \nabla \cdot (\tilde{g}(\mathbf{x}, S) \nabla S) = 0 \tag{4.1}$$

When considering the advective/dispersive groundwater equations, the saturation S is replaced by the concentration c, the fractional flow function $f(S)$ is replaced by c, and the scalar function $\tilde{g}(\mathbf{x}, S)$ is replaced by the hydrodynamic dispersion tensor $\tilde{D}$ as defined in chapter 1.

There are a number of different ways to solve equation (4.1) numerically. Traditional, straightforward methods may be computationally very expensive since the time steps and spatial discretization needs to be small in order to keep numerical diffusion at a minimum. Therefore, a number of different operator splitting methods have been introduced in order to reduce computational cost and allow for long time steps. In this chapter we will review some of these methods, in addition to a streamtube-based method by Cirpka et al.

4.1 Operator Splitting

One simple approach is to do a dimensional splitting, mentioned by for instance LeVeque in [13]. Equation (4.1) can then be reduced to 1D equations, one equation for the x-direction

$$\phi \frac{\partial S}{\partial t} + \frac{\partial}{\partial x}(v_1 f(S)) + \frac{\partial}{\partial x}\left(\tilde{g}(\mathbf{x}, S)\frac{\partial S}{\partial x}\right) = 0 \tag{4.2}$$

and one for the y-direction

$$\phi \frac{\partial S}{\partial t} + \frac{\partial}{\partial y}(v_2 f(S)) + \frac{\partial}{\partial y}\left(\tilde{g}(\mathbf{x}, S)\frac{\partial S}{\partial y}\right) = 0 \tag{4.3}$$

The splitting would then be

$$S(\mathbf{x}, n\Delta t) \approx [\mathcal{P}^y_{\Delta t} \circ \mathcal{P}^x_{\Delta t}]^n S_0(\mathbf{x}) \tag{4.4}$$

Here, $\mathcal{P}^x_{\Delta t}$ is the exact solution operator that integrates equation (4.2) one global time step Δt, and $\mathcal{P}^y_{\Delta t}$ is the same for equation (4.3).

To approximate the exact solution operators 1D finite volume methods as the one presented in the next chapter could be applied directly, or the equations could be further split into purely hyperbolic and parabolic parts individually. The main advantage with this strategy is to make use of the extensive theory that exists for purely hyperbolic equations. Consider first the splitting of equation (4.2) into a hyperbolic part

$$\phi \frac{\partial S}{\partial t} + \frac{\partial}{\partial x}(v_1 f(S)) = 0 \tag{4.5}$$

and a parabolic part

$$\phi \frac{\partial S}{\partial t} + \frac{\partial}{\partial x}\left(\tilde{g}(\mathbf{x}, S)\frac{\partial S}{\partial x}\right) = 0 \tag{4.6}$$

Here one can apply an appropriate one-dimensional solver for equation (4.5), and for instance use a finite difference method for equation (4.6). According to Espedal and Karlsen [7], finite differences can be applied on equations with even strong degeneracy. $\mathcal{P}^x_{\Delta t}$ in the operator splitting algorithm (4.4) is then replaced by $\mathcal{H}^x_{\Delta t} \circ \mathcal{S}^x_{\Delta t}$, where $\mathcal{S}^x_{\Delta t}$ is the exact solution operator of equation (4.5), and $\mathcal{H}^x_{\Delta t}$ is the same for equation (4.6). Apply the same operator splitting with the same notation to equation (4.3), and the full, semi-discrete dimensional operator splitting algorithm of equation (4.1) becomes

$$S(\mathbf{x}, n\Delta t) \approx [\mathcal{H}^y_{\Delta t} \circ \mathcal{S}^y_{\Delta t} \circ \mathcal{H}^x_{\Delta t} \circ \mathcal{S}^x_{\Delta t}]^n S_0(\mathbf{x}) \tag{4.7}$$

4.1.1 Corrected Operator Splitting

If hyperbolic solvers that don't require a CFL-condition are used for the hyperbolic updates, again according to Espedal and Karlsen [7] this method is too diffusive around any self-sharpening fronts. However, this diffusive effect can be overcome by introducing a residual flux term

$$f_{\text{res}} = f - f_c \tag{4.8}$$

The error when using large-time-step methods like the front-tracking method used by the mentioned authors in [7] turns out to be because of Oleinik's entropy condition. We will not go into much detail regarding this, but rather refer to any good book on numerical methods for hyperbolic conservation laws, for instance Aavatsmark [2] or LeVeque [13]. The term f_c in (4.8) denotes the envelope of f dictated by the entropy condition.

The residual flux term f_{res} can then be included in the parabolic term, so that instead of equation (4.6), the equation

$$\phi \frac{\partial S}{\partial t} + \frac{\partial}{\partial x}(v_1 f_{res}) + \frac{\partial}{\partial x}\left(\tilde{g}(\mathbf{x}, S)\frac{\partial S}{\partial x}\right) = 0 \tag{4.9}$$

is solved for the parabolic update. Various other variations around this kind of flux splitting has been taken by various authors, and the details are far beyond the scope of this text. However, let it be mentioned that the idea of corrected operator splitting for the 1D hyperbolic terms could also be used in streamline contexts. In fact, according to [7] the corrected operator splitting algorithm presented above is genuinely one-dimensional, and has the advantage that it does not require a priori knowledge of f_c. Therefore either a dimensional splitting as the one presented above or a streamline method as presented below must be applied.

4.2 Conventional Streamline Methods

In the previous section we considered an approach consisting of a dimensional splitting of equation (4.1), and then solving the resulting 1D equations individually. Another way to go is to start with a splitting of the physical processes, resulting in 2D purely hyperbolic and parabolic equations. The hyperbolic equation will then read

$$\phi \frac{\partial S}{\partial t} + \nabla \cdot (\mathbf{v} f(S)) = 0, \tag{4.10}$$

and the parabolic equation

$$\phi\frac{\partial S}{\partial t} + \nabla \cdot (\tilde{g}(\mathbf{x}, S)\nabla S) = 0 \tag{4.11}$$

When using streamlines equipped with time-of-flight coordinates, there is no need for the additional dimensional splitting of the hyperbolic equation, because equation (4.10) is replaced by

$$\phi\frac{\partial S}{\partial t} + \frac{\partial f(S)}{\partial \tau} = 0 \tag{4.12}$$

If the effects leading to the parabolic terms are neglected, this is a very attractive approach. The reason for this is two-sided. First of all, we have the obvious advantage of only having to solve a 1D equation for each streamline. Secondly, even if we are only solving 1D equations along streamlines, as mentioned in section (3) the time-of-flight coordinates allows us to let the streamlines represent the center lines of streamtubes. This makes the method mass conservative, as opposed to the similar approach that consists of solving equations along streamlines but using regular arc length derivatives.

One drawback here is that one has to trace the streamlines, and find the time-of-flight coordinates. Pollock's tracing algorithm takes care of both in one step, and is described for 2D, rectangular grids in appendix B. This algorithm has later been extended, and is now applicable even on irregular 3D grids [9].

Another and more serious drawback with this method is its inability to efficiently take care of cross-streamline effects. If equation (4.11) needs to be taken into account, one way is to neglect the transverse diffusive effects in (4.11), and incorporate the parallel diffusive effects into equation (4.12). This is of course not very satisfying, but could in some situations be good enough if the transverse effects indeed are negligible. Another approach is to map the solutions of equation (4.12) onto a background grid, and solve equation (4.11) on this. We propose instead a method that uses normallines to handle cross-streamline diffusive effects, and this is explained in the next chapter.

4.2.1 Mapping to and from Background Grids

For simplicity of presentation we consider here only cartesian grids with grid spacing Δx. Given the saturation $S_i^j \equiv S(\xi_i^j)$ at points $\xi_i^j \equiv \xi(\mathbf{x}_0^j, \tau_i)$ along streamlines, we construct a function $h(x, y)$ such that $\forall(i, j),\ h(\xi_i^j) = S_i^j$. Let the operator that takes this action on S_i^j be called $\mathcal{R}_{\Delta x}$. Let S_ξ denote the discrete solution defined on the streamlines.

$$h(x,y) = \mathcal{R}_{\Delta x} S_\xi \tag{4.13}$$

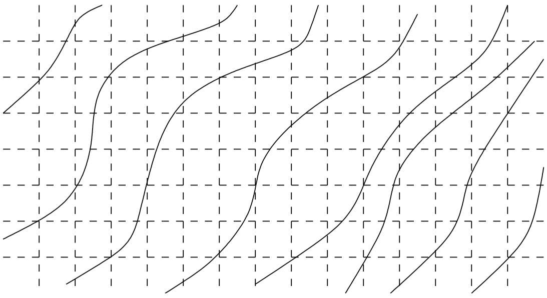


Figure 4.1: Streamlines and a background grid. To calculate diffusion one must map saturations from the streamlines onto the background grid, and then back again afterwards.

Furthermore, let $\mathcal{T}_{\Delta x}$ be an operator that has the opposite effect and interpolates the function values of $h(x,y)$ given in the cartesian grid points,

$$S_\xi = \mathcal{T}_{\Delta x} h(x,y) \tag{4.14}$$

Now, let $\mathcal{P}_\xi$ be the solution operator associated with equation (4.12), and $\mathcal{P}_{\Delta x}$ be the same for equation (4.11). Then the solution algorithm for equation (4.1) using streamlines for the hyperbolic part and a cartesian grid for the parabolic part can be written

$$S_\xi^n = [\mathcal{T}_{\Delta x} \circ \mathcal{P}_{\Delta x} \circ \mathcal{R}_{\Delta x} \circ \mathcal{P}_\xi]^n S_\xi^0 \tag{4.15}$$

In Chapter 5 we will apply this method to a groundwater flow problem. Details regarding how the mapping between streamlines and background grid is performed in practice are found in Appendix C.

The biggest problem with this approach is that the numerical diffusion associated with the mapping could well be greater than the physical diffusion one is trying to model. If this is the case, then there is not much point in solving the parabolic equation at all. Another problem is mass conservation. When interpolating the solution like this, we have no guarantee that the total mass stays the same. More sophisticated mapping methods than straightforward ones are being developed, see for instance Mallison et. al. [14].

4.3 Streamline-oriented Grids

To solve the equations transformed to streamline and normalline coordinates, one way to go is by using grids generated by streamtubes. In [6], Cirpka et al. describes how to generate such grids, and in [5], the same authors describe how the equations are solved. This is an approach that is restricted to stationary flow fields, and therefore cannot be directly applied to the two-phase equations if the problem is such that the nonlinear interplay between the saturation and pressure equations imply that the velocity field is time-dependent. However, it is well suited for solving the groundwater flow equations when the flow is stationary.

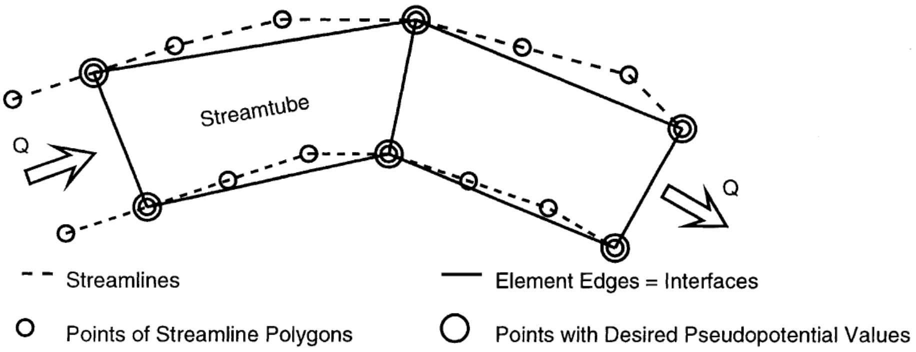

Streamtube
Q
Q
Streamlines
Element Edges = Interfaces
Points of Streamline Polygons
Points with Desired Pseudopotential Values

Figure 4.2: Cell elements generated by two streamlines being considered the boundary of a streamtube. The figure is from an article of Cirpka et al., [6].

The authors are using cell-centered finite volume methods where the cells are being generated by streamlines. We will not go into much detail regarding the generation of such grids, bur rather refer to [6]. The grids are generated such that the cell edges in streamline coordinates are parallel and perpendicular to the direction of flow with respect to the cell-center. Two such cell elements are illustrated in figure 4.3. The authors are further considering reactive transport of multiple components, with which we are not concerned. So for our problem consisting of a single non-reactive component they would solve the equation

$$\phi\frac{\partial c}{\partial t}+\frac{\partial(vc)}{\partial l_l}-\frac{\partial}{\partial l_l}\left((v\alpha_l+\phi D_m)\frac{\partial c}{\partial l_l}\right)-\frac{\partial}{\partial l_t}\left((v\alpha_t+\phi D_m)\frac{\partial c}{\partial l_t}\right)=Q \quad (4.16)$$

where $\partial/\partial l_l$ means differentiation in the direction parallel to the direction of flow, and $\partial/\partial l_t$ means differentiation in the direction perpendicular to it.

The solution procedure for a cell over one time step is to first calculate the advective fluxes over all cell edges connected to the cell of interest. Then the dispersive fluxes are calculated based on the updated concentrations. For details on how the fluxes are calculated, we refer to [5].

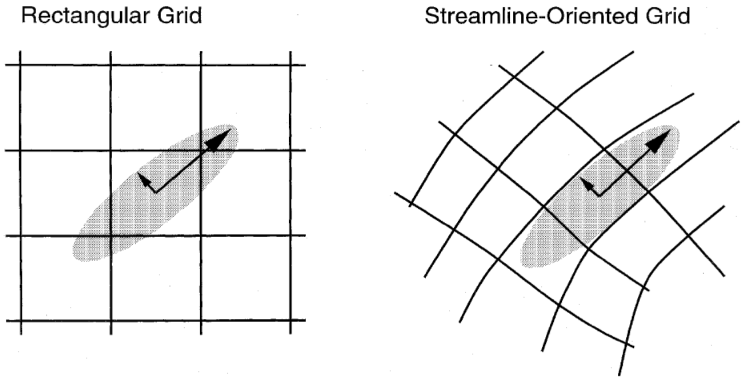


Figure 4.3: Dispersive transport on rectangular on streamline-oriented grids. The figure is from another article of Cirpka et al., [5].

Figure 4.3 illustrates the advantage of using streamline-oriented grids for the calculation of dispersive transport. When using rectangular grids a 9 point stencil must be used, while for streamline-oriented grids, one only needs to consider the boundary fluxes.

Chapter 5

A Streamline-normalline Method

As mentioned in the previous chapter, streamline methods are very attractive if cross-streamline effects can be neglected. These effects can be simulated on background grids, but this may lead to great numerical diffusion. In this chapter we propose a method that intends to deal with cross-streamline effects, while at the same time reducing the 2D equations to one-dimensional sub-problems along streamlines and normallines. That is, we split the time-of-flight formulations of either the saturation equation or the groundwater contamination equation into a part along streamlines,

$$\frac{\partial S}{\partial t} + \frac{\partial f(S)}{\partial \tau} + \frac{\partial}{\partial \tau}\left(\frac{\tilde{g}(S,\mathbf{x})\phi}{\|\mathbf{v}\|^2}\frac{\partial S}{\partial \tau}\right) = 0 \tag{5.1}$$

and a part along normallines,

$$\frac{\partial S}{\partial t} + \frac{\partial}{\partial \nu}\left(\frac{\tilde{g}(S,\mathbf{x})\phi}{\|\mathbf{n}\|^2}\frac{\partial S}{\partial \nu}\right) + \left(\frac{\tilde{g}(S,\mathbf{x})\phi}{\|\mathbf{n}\|^2}\frac{\partial S}{\partial \nu}\right)\nabla \cdot \mathbf{n} = 0 \tag{5.2}$$

Again, for simplicity of presentation we consider only the saturation equation. We will only consider source terms as point sources, such that sources or sinks are considered boundary conditions for equation (5.1) along streamlines starting or ending in a source or a sink.

Let $\mathcal{P}^{\tau}_{\Delta t}$ and $\mathcal{P}^{\nu}_{\Delta t}$ be the exact solution operators of (5.1) and (5.2) respectively. Then the semi-discrete splitting can be written

$$S(\mathbf{x}, n\Delta t) \approx [\mathcal{P}^{\nu}_{\Delta t} \circ \mathcal{P}^{\tau}_{\Delta t}]^n S_0(\mathbf{x}) \tag{5.3}$$

To approximate these solution operators we use control volumes. If we use the intersection points between streamlines and normallines as discretization points in the one-dimensional equations, no mapping is needed when going from the streamline equation to the normalline equation. The problem here is that it may take a great number of streamlines and normallines in order to get a satisfactory grid.

That this method is semi-discrete means that the solution operators $\mathcal{P}^{\tau}_{\Delta t}$ and $\mathcal{P}^{\nu}_{\Delta t}$ are taken to be exact. In practice and in general we need to approximate these with discretizations. Furthermore, after applying the discrete versions of the solution operators, we need operators to transfer the solutions from the range of one discrete solution operator to the domain of the other. As the discrete solution operators, we will use 1D control volume methods to solve the one-dimensional equations where the time-of-flight coordinates acts as the space variables. The next section describes how this is done.

5.1 1D Control Volume Methods

Consider a 1D advection-diffusion-equation on the form

$$\frac{\partial u}{\partial t} + \frac{\partial f(u)}{\partial x} = \frac{\partial}{\partial x}\left(g(u,x)\frac{\partial u}{\partial x}\right) \tag{5.4}$$

Given a set of grid points x_i on the x-axis, and t^n on the t-axis, we divide the domain into *control volumes* as shown in figure 5.1, and then integrate over such a volume following Aavatsmark in [2]. To simplify notation, we also define the expressions $u^n = u^n(x) = u(t^n, x)$ and $u_i = u_i(t) = u(t, x_i)$.

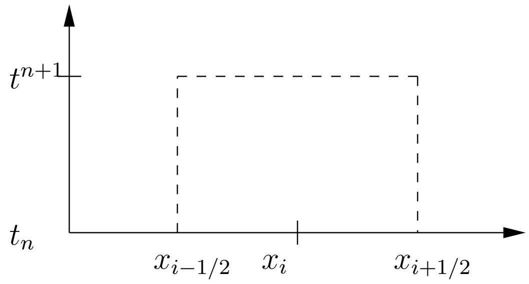


Figure 5.1: A 1D control volume

$$\begin{aligned}\int_{t^n}^{t^{n+1}}\int_{x_{i-1/2}}^{x_{i+1/2}}\frac{\partial u}{\partial t}\mathrm{d}x\mathrm{d}t+\int_{t^n}^{t^{n+1}}\int_{x_{i-1/2}}^{x_{i+1/2}}\frac{\partial f(u)}{\partial x}\mathrm{d}x\mathrm{d}t=\\ \int_{t^n}^{t^{n+1}}\int_{x_{i-1/2}}^{x_{i+1/2}}\frac{\partial}{\partial x}\left(g(u,x)\frac{\partial u}{\partial x}\right)\mathrm{d}x\mathrm{d}t\end{aligned} \tag{5.5}$$

$\Rightarrow$

$$\begin{aligned}\int_{x_{i-1/2}}^{x_{i+1/2}}u^{n+1}-u^n\mathrm{d}x+\int_{t^n}^{t^{n+1}}f(u_{i+1/2})-f(u_{i-1/2})\mathrm{d}t=\\ \int_{t^n}^{t^{n+1}}\left(g(u_{i+1/2},x_{i+1/2})\left[\frac{\partial u}{\partial x}\right]_{x=x_{i+1/2}}\right)-\\ \left(g(u_{i-1/2},x_{i-1/2})\left[\frac{\partial u}{\partial x}\right]_{x=x_{i-1/2}}\right)\mathrm{d}t\end{aligned} \tag{5.6}$$

This equation is on conservation form, and if we replace these integrals with approximations of them, we get a mass conserving discretization scheme. We therefore let $\Delta x_i = x_{i+1/2} - x_{i-1/2}$ and $\Delta t^{n+1} = t^{n+1} - t^n$ and define the following expressions:

$$u_i^n \approx \frac{1}{\Delta x_i}\int_{x_{i-1/2}}^{x_{i+1/2}}u^n\mathrm{d}x \tag{5.7}$$

$$\phi_{i+1/2}^{n+1/2} \approx \frac{1}{\Delta t^{n+1}}\int_{t^n}^{t^{n+1}}f(u_{i+1/2})\mathrm{d}t \tag{5.8}$$

$$\psi_{i+1/2}^{n+1/2} \approx \frac{1}{\Delta t^{n+1}}\int_{t^n}^{t^{n+1}}\left(g(u_{i+1/2},x_{i+1/2})\left[\frac{\partial u}{\partial x}\right]_{x=x_{i+1/2}}\right)\mathrm{d}t \tag{5.9}$$

Equations (5.8) and (5.9) are called the *advective numerical flux* and the *diffusive numerical flux*, respectively. Equation (5.6) can then be written

$$\frac{u_i^{n+1}-u_i^n}{\Delta t^{n+1}}+\frac{\phi_{i+1/2}^{n+1/2}-\phi_{i-1/2}^{n+1/2}}{\Delta x_i}=\frac{\psi_{i+1/2}^{n+1/2}-\psi_{i-1/2}^{n+1/2}}{\Delta x_i} \tag{5.10}$$

For the flux in (5.8) we use Godunov's method. Define

$$\phi_{i+1/2}^{n+1/2}=\begin{cases}\min\limits_{u\in[u_i^n,u_{i+1}^n]} f(u) & \text{for } u_i^n \le u_{i+1}^n\\ \max\limits_{u\in[u_i^n,u_{i+1}^n]} f(u) & \text{for } u_{i+1}^n \le u_i^n\end{cases} \tag{5.11}$$

For the diffusive flux in (5.9) we use

$$\psi_{i+1/2}^{n+1/2} = g\left(\frac{u_i^n + u_{i+1}^n}{2}, x_{i+1/2}\right)\frac{u_{i+1}^n - u_i^n}{x_{i+1} - x_i} \tag{5.12}$$

The equation (5.1) is on the form (5.4), and can therefore be written on conservation form. The same goes for equation (5.2) if $\nabla\cdot\mathbf{n} = 0$. If $\nabla\cdot\mathbf{n} \neq 0$, there will be an additional term in equation (5.4) on the form $h(u,x)\frac{\partial u}{\partial x}$. This term is then discretized by

$$\left[h(u,x)\frac{\partial u}{\partial x}\right]_{x=x_i} \approx h(u(t,x_i),x_i)\frac{u(t,x_{i+1}) - u(t,x_i)}{x_{i+1} - x_i} \tag{5.13}$$

5.2 Coupling the Solution Operators

As already mentioned, since the solution operators are only acting on streamlines and normallines, we need some way of taking the solution from the range of one solution operator to the domain of the other. Consider figure 5.2. Given a streamline $\xi(\mathbf{x}_M,\tau)$ and a normalline $\eta(\mathbf{x}_N,\nu)$, the intersection points are given by $\mathbf{x}_{M,N} = \xi(\mathbf{x}_M,\tau) = \eta(\mathbf{x}_N,\nu)$. $\mathbf{x}_{M,N}$ need not be unique, nor necessarily exist. Assume now that we have found an intersection point, which for simplicity may be considered unique, and let τ_i and ν_j be such that $\mathbf{x}_{M,N} = \xi(\mathbf{x}_M,\tau_i) = \eta(\mathbf{x}_N,\nu_j)$.

Define the discrete solution $\bar{S}_{M,N}^n$ in the point $\mathbf{x}_{M,N}$ at the time t^n to be the average of the solution inside the dotted area Ω,

$$\bar{S}_{M,N}^n = \frac{1}{A(\Omega)}\int_\Omega S(t^n,\mathbf{x})\mathrm{d}x\mathrm{d}y \tag{5.14}$$

where $A(\Omega)$ is the area of Ω. If we now think of Ω as the intersection between the streamtube corresponding to the streamline $\xi(\mathbf{x}_M,\tau)$ and the streamtube corresponding to the normalline $\eta(\mathbf{x}_N,\nu)$, we can define the discrete solution in the intersection point along streamlines as

$$\bar{S}_\xi^n(\tau_i) = \frac{1}{\Delta\tau_i}\int_{\tau_{i-1/2}}^{\tau_{i+1/2}} S(t^n,\xi(\mathbf{x}_M,\tau))\mathrm{d}\tau \tag{5.15}$$

where $\Delta\tau_i = |\tau_{i+1/2} - \tau_{i-1/2}|$. Correspondingly, we can define the discrete solution in the intersection point along normallines as,

$$\bar{S}_\eta^n(\nu_j) = \frac{1}{\Delta\nu_j}\int_{\nu_{j-1/2}}^{\nu_{j+1/2}} S(t^n,\eta(\mathbf{x}_N,\nu))\mathrm{d}\nu \tag{5.16}$$

where $\Delta\nu_j = |\nu_{j+1/2} - \nu_{j-1/2}|$

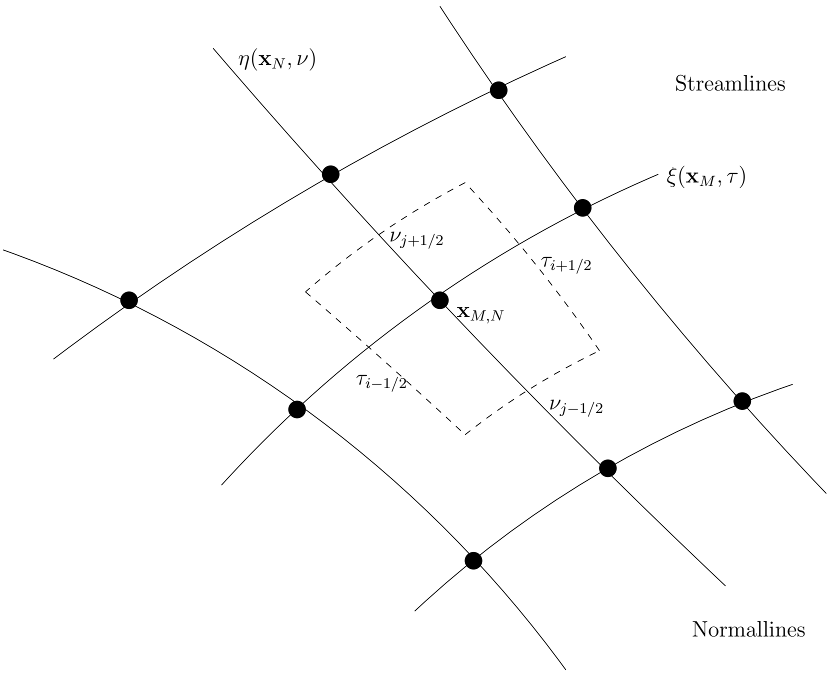


Figure 5.2: Streamlines, normallines and intersection points. The area inside the dotted line is a control volume.

Because we are using time-of-flight coordinates, the three expressions above all represent the averaged solution inside Ω, and it is justified to write

$$\bar{S}^n_{M,N} = \bar{S}^n_{\xi}(\tau_i) = \bar{S}^n_{\eta}(\nu_j). \tag{5.17}$$

This is the key to coupling the solution operators in a conservative way. After solving the equation along streamlines, make sure relationship (5.17) holds for all intersection points, and then solve along normallines. Make sure the relationship holds once more, and then solve along streamlines.

5.3 The Fully Discrete Method

The easiest way to implement this method is to simply use the intersection points along streamlines as grid points in the 1D control volume discretization. However, this may well lead to unnecessary numerical diffusion if the intersection points are far apart. To cope with this type of numerical diffusion using this straightforward approach, the only way to go is by increasing the number of streamlines and normallines, thereby quadratically increasing the number of control volumes and the computing cost. It is therefore instead natural to refine the one-dimensional grids along streamlines and normallines, giving a linearly increasing number of control volumes.

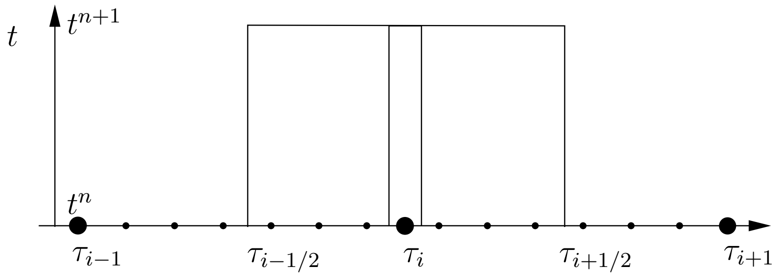


Figure 5.3: Dividing the 1D control volumes into smaller control volumes

Inspired by figure 5.3, let τ_i be the time-of-flight at which the streamline $\xi(\mathbf{x}_M, \tau)$ intersects a normalline. In between $\tau_{i-1/2}$ and $\tau_{i+1/2}$, place K_i additional points $\tau_i^k,\ k = 1, \ldots, K_i$. For each i, define

$$\Delta\tau_i^k = \tau_i^{k-1/2} - \tau_i^{k+1/2},\ k = 1, \ldots, K_i \tag{5.18}$$

where $\tau_i^{1/2} = \tau_{i-1/2}$ and $\tau_i^{K_i+1/2} = \tau_{i+1/2}$. Defining the τ_i^k in such a way assures that

$$\sum_k \Delta\tau_i^k = \Delta\tau_i \tag{5.19}$$

Now we can define the refined solution in the points τ_i^k by

$$S_\xi^n(\tau_i^k) = \frac{1}{\Delta\tau_i^k} \int_{\tau_i^{k-1/2}}^{\tau_i^{k+1/2}} S(t^n, \xi(\mathbf{x}_M, \tau))\mathrm{d}\tau \tag{5.20}$$

With these definitions we get that

$$\sum_k \Delta\tau_i^k S_\xi^n(\tau_i^k) = \Delta\tau_i \bar{S}_\xi^n(\tau_i) \tag{5.21}$$

Note that by the above definitions, $\bar{S}_\xi^n(\tau_i)$ is the mean value of the solution in the 2D control volume, and also the mean value of the discrete solutions defined in the refined grid points, $S_\xi^n(\tau_i^k)$.

Similar expressions are defined on the normallines, $\bar{S}_\eta^n(\nu_j) = \bar{S}_\xi^n(\tau_i)$ and $S_\eta^n(\nu_j^k)$ is defined in the same way as $S_\xi^n(\tau_i^k)$ and such that

$$\sum_k \Delta\nu_j^k S_\eta^n(\nu_j^k) = \Delta\nu_j \bar{S}_\eta^n(\nu_j) \tag{5.22}$$

Now we are ready to define our fully discrete method. We begin by solving the equation on the refined grid along the streamlines. Define an operator $\mathcal{M}_\xi$ that computes mean values in the following way,

$$\bar{S}_\xi^n = \mathcal{M}_\xi S_\xi^n : \bar{S}_\xi^n(\tau_i) = \frac{1}{\Delta\tau_i}\sum_k \Delta\tau_i^k S_\xi^n(\tau_i^k) \tag{5.23}$$

Now we need an operator that updates the solution on the fine grid along the normallines, keeping $\bar{S}_\eta^n(\nu_j) = \bar{S}_\xi^n(\tau_i)$ in the intersection points. This operator we call $\mathcal{D}_\eta$ and may be defined in a number of ways, but we choose to let it scale the solution on the fine grid by the same factor as the coarse grid is scaled after a time step,

$$S_\eta^n = \mathcal{D}_\eta \bar{S}_\xi^n : S_\eta^n(\nu_j^k) = \frac{\bar{S}_\xi^n(\tau_i)}{\bar{S}_\xi^{n-1}(\tau_i)} \cdot S_\eta^{n-1}(\nu_j^k) \tag{5.24}$$

The operator that computes the mean values for the normallines $\mathcal{M}_\eta$, and the one that distributes them on the fine grid along streamlines, $\mathcal{D}_\xi$ are defined in the same way. That is, let

$$\bar{S}_\eta^n = \mathcal{M}_\eta S_\eta^n : \bar{S}_\eta^n(\nu_j) = \frac{1}{\Delta\nu_j}\sum_k \Delta\nu_j^k S_\eta^n(\nu_j^k) \tag{5.25}$$

and

$$S_\xi^n = \mathcal{D}_\xi \bar{S}_\eta^n : S_\xi^n(\tau_i^k) = \frac{\bar{S}_\eta^n(\nu_j)}{\bar{S}_\eta^{n-1}(\nu_j)} \cdot S_\xi^{n-1}(\tau_i^k) \tag{5.26}$$

Using the notation above, we can write our fully discrete method

$$S_\xi^n = [\mathcal{D}_\xi \circ \mathcal{M}_\eta \circ \mathcal{P}_\eta \circ \mathcal{D}_\eta \circ \mathcal{M}_\xi \circ \mathcal{P}_\xi]^n S_\xi^0 \tag{5.27}$$

Algorithm 1 below shows how the method is implemented.

Algorithm 1

```
n=0
Define $S_\xi^0$, $\bar{S}_\xi^0$,$S_\eta^0$ and $\bar{S}_\eta^0$ according to initial values
while $n \leq$ number of time steps do
  for $i = 1$ to number of streamlines do
    if $n \neq 0$ then
      Update $\bar{S}_\xi^n$ such that (5.17) holds at all intersection points
      Update $S_\xi^n$ for normalline $i$ according to (5.24)
    end if
    Update $S_\xi^n$ for streamline $i$ by solving equation (5.1)
    Update $\bar{S}_\xi^n$ for streamline $i$ according to (5.23)
  end for
  for $i = 1$ to number of normallines do
    Update $\bar{S}_\eta^n$ such that (5.17) holds at all intersection points
    Update $S_\eta^n$ for normalline $i$ according to (5.26)
    Update $S_\eta^n$ for normalline $i$ by solving equation (5.2)
    Update $\bar{S}_\eta^n$ for normalline $i$ according to (5.25)
  end for
  $n = n + 1$
end while
```

Chapter 6

Simulation of CO_2 Injection in Confined Aquifers

To test the method of handling the cross-streamline diffusive effects along normallines, we will in this chapter consider a problem to which there exists reference solutions. A very simple case is the one of radial flow where the velocity is inversely proportional to the distance from the well. We show in the next section that this is the case. If the flow was purely radial, there would be no cross-streamline diffusive effects. Therefore, we add some rotation to the velocity field to be able to study the interesting parts of the equations.

6.1 Construction of the Test Case

We consider a homogeneous, isotropic, infinite reservoir with a CO_2 injection well in the origin. Since we are adding a strictly rotational term to the otherwise radial velocity field, it is natural to utilize the differential operators in polar coordinates. Let $\mathbf{v}_\alpha = v_{\alpha r}\mathbf{e}_r + v_{\alpha\theta}\mathbf{e}_\theta$ be the velocity field written in polar coordinates. Then

$$\nabla = \frac{\partial}{\partial r}\mathbf{e}_r + \frac{1}{r}\frac{\partial}{\partial\theta}\mathbf{e}_\theta \tag{6.1}$$

$$\nabla\cdot\mathbf{v}_\alpha = \frac{1}{r}\frac{\partial}{\partial r}(rv_{\alpha r}) + \frac{1}{r}\frac{\partial}{\partial\theta}v_{\alpha\theta} \tag{6.2}$$

The velocity field is further assumed to be symmetric in the sense that $v_{\alpha r} = v_{\alpha r}(r)$ and $v_{\alpha\theta} = v_{\alpha\theta}(r)$, making the second term on the right hand side of equation (6.2) disappear. The injection well is considered a point source, and is isolated from the domain. The law of mass conservation (1.15) can then be written

$$\frac{1}{r}\frac{\partial}{\partial r}\left(r v_{\alpha r}\right) = \phi \frac{\partial S_\alpha}{\partial t} \tag{6.3}$$

This equation is summed over the two phases, giving

$$r v_r = C \tag{6.4}$$

where v_r is the radial part of the total velocity and C is a constant of integration. Hence the radial part of the total velocity is inversely proportional to the distance from the well under the symmetry conditions assumed, and our velocity field will be analytically given by

$$\mathbf{v} = \frac{\alpha}{x^2+y^2}\begin{pmatrix} x \\ y \end{pmatrix} + \frac{\alpha}{(x^2+y^2)^\beta}\begin{pmatrix} y \\ -x \end{pmatrix} \tag{6.5}$$

We further let the normal field be given by $\mathbf{n} = R\mathbf{v}$, R again being the 90 degree counter clockwise rotation matrix,

$$\mathbf{n} = \frac{\alpha}{x^2+y^2}\begin{pmatrix} -y \\ x \end{pmatrix} + \frac{\alpha}{(x^2+y^2)^\beta}\begin{pmatrix} x \\ y \end{pmatrix} \tag{6.6}$$

Then α is a parameter used to set the well injection rate, and β scales the rotation. Since $v_{rad} = \alpha/r$, we get a simple expression for the well injection rate, $Q_{well} = \alpha 2\pi H$. Also, the expression

$$\|\mathbf{v}\|^2 = \|\mathbf{n}\|^2 = \alpha^2\left\{\frac{1}{r^2} + \frac{1}{r^{4\beta-2}}\right\} \tag{6.7}$$

is needed in the equations.

Figure 6.1 shows the streamlines and normallines for different values of β. There are 10 streamlines and 10 normallines in each case.

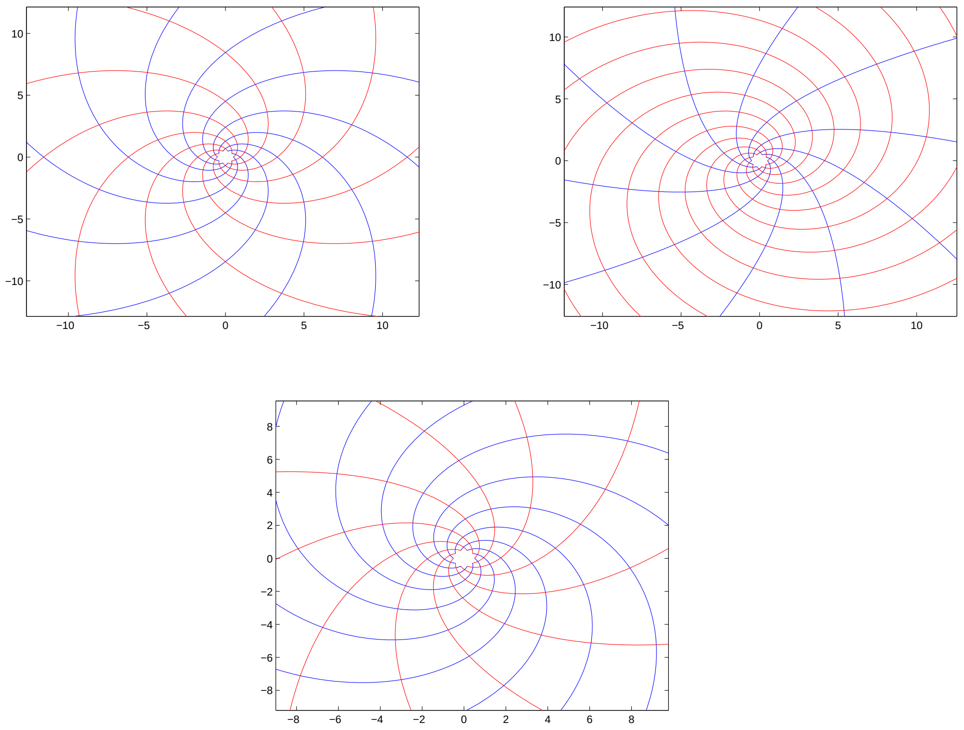


Figure 6.1: Top left: $\beta = 1$, top right: $\beta = 1.2$, bottom: $\beta = 0.9$. Blue lines are the streamlines, red lines are the normallines.

6.1.1 Model Parameters

We will use the following values for the parameters in our equations.

Residual water saturation	S_r	0
CO$_2$ viscosity	μ_T	0.061 mPa s
Water viscosity	μ_B	0.511 mPa s
CO$_2$ density	ρ_T	733 kg m^{-3}
Water density	ρ_B	1099 kg m^{-3}
Permeability	$k(\mathbf{x})$	0.02 Darcy
Aquifer height	H	15 m
Gravity constant	g	9.8 m/s^2
Porosity	ϕ	0.15

This will make the flux-functions f and g look like this:

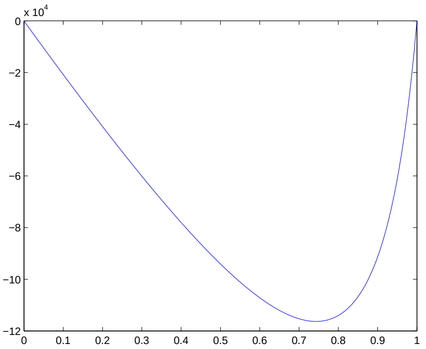


Figure 6.2: $\tilde{g}(S, \mathbf{x})$

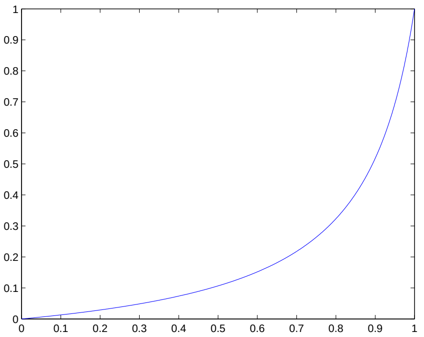


Figure 6.3: $f(S)$

6.2 Mass Conserving Code

In this section we aim to show that the numerical setup really is mass conserving when $\nabla \cdot \mathbf{n} = 0$, that is when $\beta = 1$. The injection well is considered a point source, but the streamlines need to be started a distance from the point source, so we consider the well to be a cylinder of radius 0.5m. The boundary condition $S = 0$ cannot be applied to the streamlines because diffusive effects will have some water entering into the well. To avoid this, we let a cylinder of CO$_2$ be present around the well before injection starts. We let the radius of this cylinder be large enough so that no water is allowed to enter the well. The numerically integrated volume at any time step should then be the starting volume of CO$_2$ plus total injected volume.

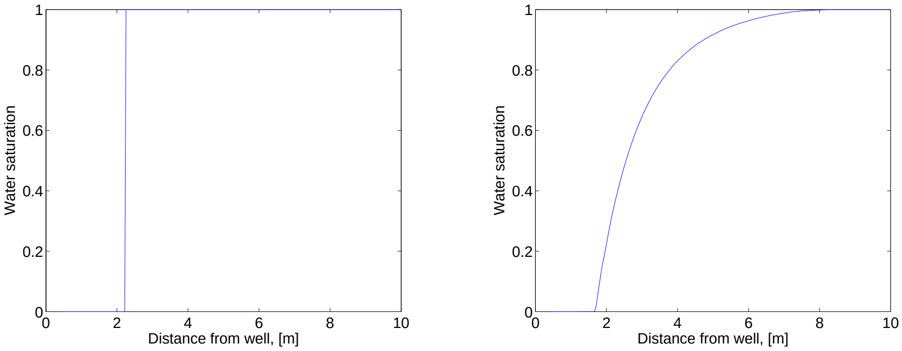


Figure 6.4: Left: $t = 0$, Right: $t = 34000$

Figure 6.4 shows what the test case looks like. As we see, diffusion effects has some water enter into the volume that initially was pure CO_2, and this is the reason why the initial cylinder should be present at the start of the injection. The injection rate is set to $94.2488\text{m}^3/\text{day}$.

In figure 6.5 below are two graphs showing the development of the mass balance error (integrated volume minus injected volume) and the relative mass balance error for a case with 20 streamlines and 20 normallines. We see that there is a small error which initially increases a bit before decreasing. There is some error associated with the positions of the intersection points and corresponding time-of-flight coordinates, and we expect these small mass balance errors to be a result of this.

The grid used to produce the graphs below was refined with 4 additional points between each intersection point along both streamlines and normallines. Changing either the refinement or the number of streamlines and normallines had little effect on the mass balance errors, the order of magnitude and behavior was unchanged. What had an effect was changing the accuracy in the numerical integration of the streamlines.

6.2.1 Nonzero Divergence

As mentioned before, when $\nabla \cdot \mathbf{n} \neq 0$, the time-of-flight formulation of the saturation equation cannot be written on conservation form without solving an additional set of differential equations to obtain a scaling function for the normal field. A simple calculation shows that for our setup

$$\nabla \cdot \mathbf{n} = \alpha \frac{2}{r^{2\beta}}(1-\beta) \tag{6.8}$$

so that $\nabla \cdot \mathbf{n}$ is at the highest around the well and decreases with r. Figure 6.6 shows a plot of $\nabla \cdot \mathbf{n}$ as a function of r and β.

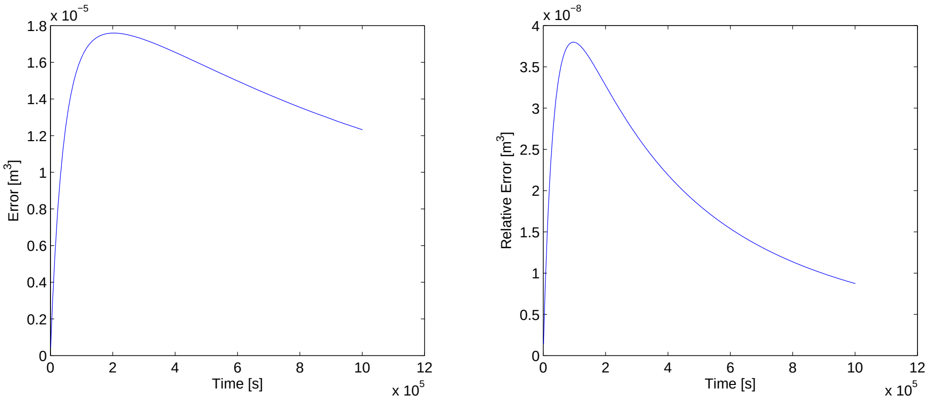


Figure 6.5: Mass balance error and corresponding relative error for a refined grid with 20 streamlines and 20 normallines.

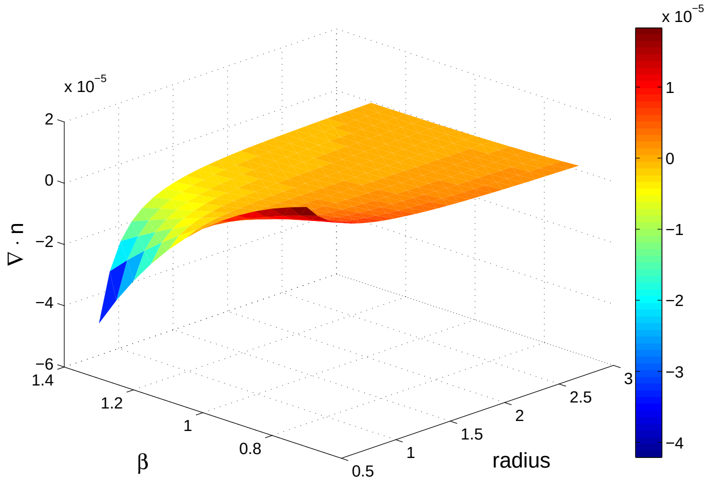


Figure 6.6: $\nabla \cdot \mathbf{n}$ as a function of r and β.

Table 6.1 shows the mass balance error and corresponding relative error for 5 different values of β at time $t = 10^6$ s. The table shows that mass indeed is lost or gained if the flow is rotational.

6.3 Simulations

In [16] the authors show that the radial problem of CO_2 being injected into an aquifer has a self-similar solution satisfying a nonlinear, second-order differential equation when assuming incompressible, immiscible fluids and a

β	Error	Rel. error
1	$1.2323 \cdot 10^{-5}$	$-8.7338 \cdot 10^{-9}$
1.1	-0.8142	$-5.8873 \cdot 10^{-4}$
1.2	-0.7686	$-5.5283 \cdot 10^{-4}$
0.9	1.3918	$9.3939 \cdot 10^{-4}$
0.8	1.8502	$1.3309 \cdot 10^{-3}$

Table 6.1: Mass balance errors and the corresponding relative errors for different values of β at $t = 10^6$ s.

homogeneous, isotropic confined aquifer. This similarity solution is used as a reference solution to our streamline-normalline method. The reference case will be the one where the injection rate $Q_{well} = 94\text{m}^3/\text{day}$. As it is seen from figure 6.2, the diffusion term vanishes at the points where the water saturation is either 0 or 1. We should then perhaps expect our numerical methods to have problems around these points.

The boundary condition for the start of the streamlines is now time-dependent, and we let $S_\xi^n(r_0) = S(n\Delta_\tau t, r_0)$ where S_ξ^n is the discrete solution and $S(t, r)$ is the interpolated reference solution. $\Delta_\tau t$ is the time step used for solving the equation along streamlines.

6.3.1 Adjusting the Number of Streamlines

First, we look at how the numerical solution approximates the reference solution when the number of streamlines is the only thing being changed. Figure 6.7 shows the numerical solution and reference solution for four different setups, and we see that as the number of streamlines/normallines are increased, the numerical solution seems to better approximate the reference solution. In all the cases in this section, $\nabla \cdot \mathbf{n} = 0$.

As shown in figures 6.9-6.11, the errors decrease when the number of streamlines and normallines increase. An important factor here is that this way of decreasing the numerical diffusion leads to a quadratic increase in the number of control volumes used in the computations. That is, if the number of intersection points along a streamline is m, and the number of streamlines are n, the number of 1D control volumes summed over all streamlines are $m \cdot n$ (and the number of 1D control volumes summed over streamlines and normallines would be $2 \cdot m \cdot n$, but the factor 2 stays the same, so we consider only the number of control volumes along streamlines). If the number of streamlines and normallines are doubled to $2n$, the number of intersection points will also be doubled. This gives a total number of $2n \cdot 2m$ 1D control

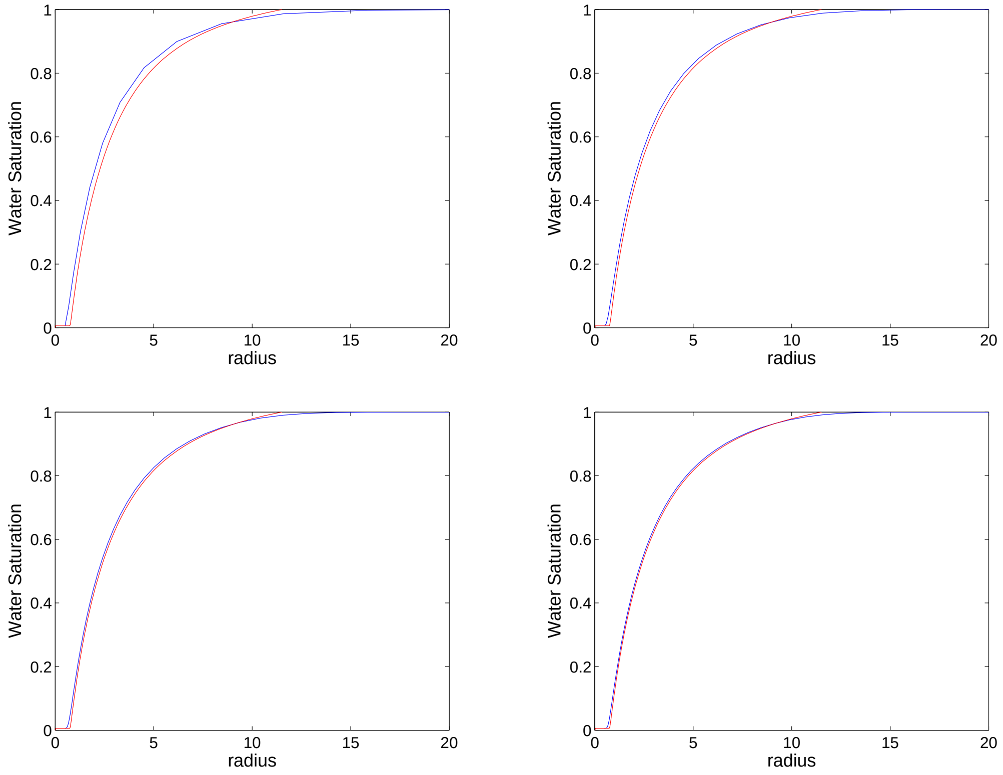


Figure 6.7: Numerical solution and reference solution for four different setups. The number of streamlines are the same as the number of normallines in each case. Top left: 10, top right: 20, bottom left: 30, bottom right: 40. The figures show the simulations at time $t = 10^5$s

volumes summed over streamlines.

6.3.2 Adjusting the Operator Splitting Steps

We saw in the previous section how the numerical solutions approached the exact solution as the number of streamlines and normallines was increased. This is of course not unexpected, since an increased number of control volumes for the 1D solution operators should give less numerical diffusion. However, this way of increasing the accuracy is expensive. In this section we first consider the same setup as before, but take the grid with 20 streamlines and 20 normallines, and further refine this grid.

The refinement is done in the following way. Given the time-of-flight coordinates for the k intersection points along streamlines $\tau_1, \tau_2 \ldots \tau_k$, we linearly place a number of additional points in between each intersection point. The number of additional points is in our implementation always the same between all the different intersection points along a streamline. Furthermore, if we stick to placing an even number of additional points between two intersection points, we automatically fulfill the relationship (5.18). The possible advantage with this approach has already been mentioned. Refining the 1D grids instead of increasing number of streamlines gives a linear increase in the total number of 1D control volumes, while as figure 6.7 shows, reducing the numerical diffusion.

Figure 6.8 shows that the numerical solution approaches the reference solution as the refinement increases. However, just by looking at the figures 6.7 and 6.8 it is not possible to quantify the errors being made in any other way than that the error seems to be decreasing. In figures 6.9-6.11 the ℓ^2-norm of the vector $S_\xi - S_{ref}$ for different grids, refinements and global time steps is plotted. Here S_ξ is the numerical solution in the intersection points as defined in (5.15). S_{ref} is the reference solution interpolated in the intersection points. In the figures, the "refinement factor" 2 means no refinement, and generally refinement factor n means $n-2$ additional points between two streamlines or two normallines. In all of the plots below, the refinement factor is the same along streamlines and normallines.

Figure 6.9 shows the behavior of the error on a grid with 10 streamlines and 10 normallines, figure 6.10 on a grid with 20 streamlines and 20 normallines, and figure 6.11 on a grid with 40 streamlines and 40 normallines. Consider first figure 6.9, where the numbers 1-3 indicates different types of errors being made. The first kind of error we can call E_1. We should expect a big part of this error to be due to the control volume formulation. The discrete solution value in a point is defined as the average solution in a time-of-flight interval. Therefore, the average solution in such an interval should

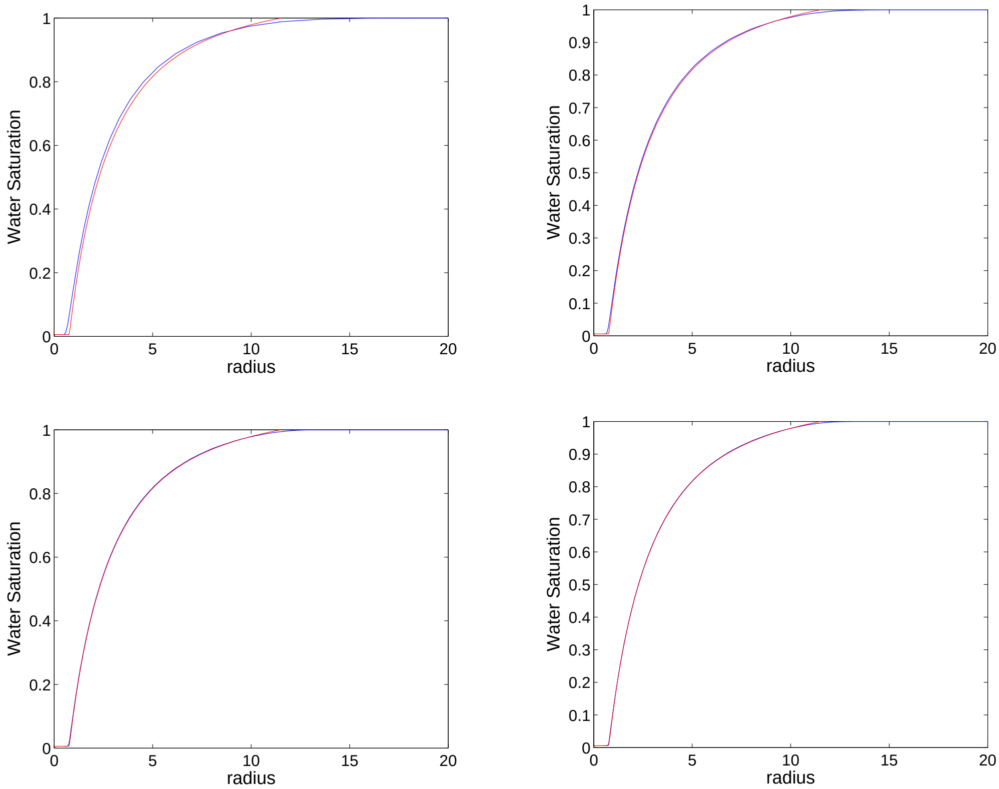


Figure 6.8: Numerical solution and reference solution for four different setups. In each case there are 20 streamlines and 20 normallines. The difference is that there is further refinement along the streamlines and normallines. Top left: no additional points between intersection points, top right: 2 additional points between 2 intersection points, bottom left: 8 additional points between 2 intersection points, bottom right: 18 additional points.

be different from the reference solution in the intersection points. We see that E_1 decreases as the number of streamlines and normallines increase, as it should.

The error E_2 can be associated with the numerical diffusion that arises when the 1D equations are solved as this error decreases when the refinement factor increases. The error E_3 decreases when the global time step is reduced. This term then seems to represent the operator splitting error. Notice how E_3 is of the same order of magnitude for all grids and all refinement factors.

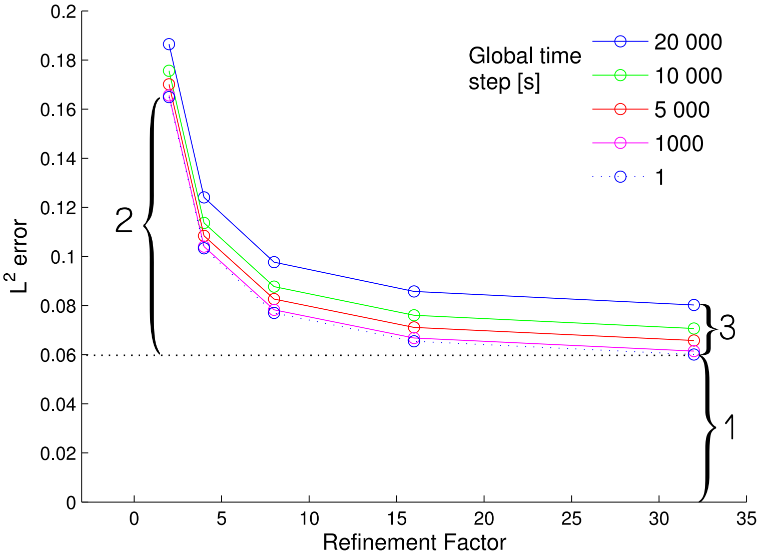


Figure 6.9: Plot of ℓ^2-norm of $S_\xi - S_{ref}$ at time $t = 10^5$s on a grid with 10 streamlines and 10 normallines for different refinement factors.

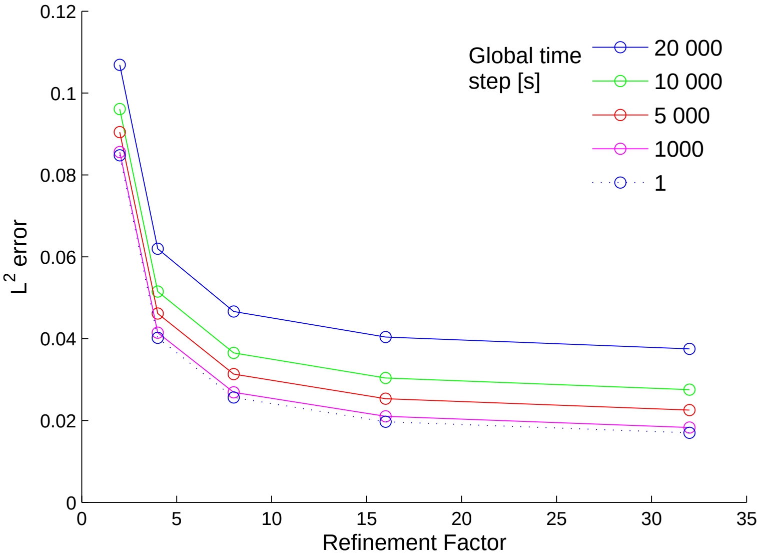


Figure 6.10: Plot of ℓ^2-norm of $S_\xi - S_{ref}$ at time $t = 10^5$s on a grid with 20 streamlines and 20 normallines for different refinement factors.

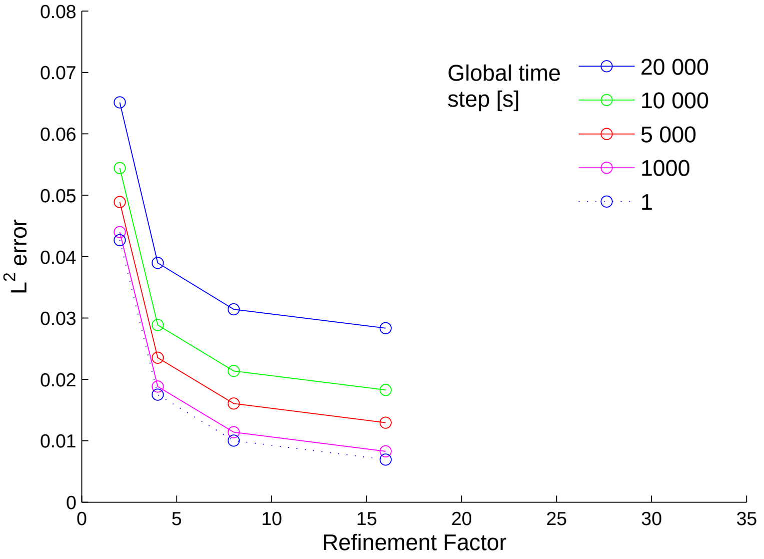


Figure 6.11: Plot of ℓ^2-norm of $S_\xi - S_{ref}$ at time $t = 10^5$s on a grid with 40 streamlines and 40 normallines for different refinement factors.

There is an additional type of error that has yet to be discussed, the error associated with the discrete operator splitting coupling. This error might be a part of any of the errors $E_1 - E_3$ since it may depend on a number of factors - global time steps, refinement, and positions of intersection points. As figure 6.12a shows, the ℓ^2-error is actually less when the streamlines are refined but the normallines are not, compared to the case where both are being refined. This indicates that the scaled coupling might not be suitable for the normallines, at least not in all cases.

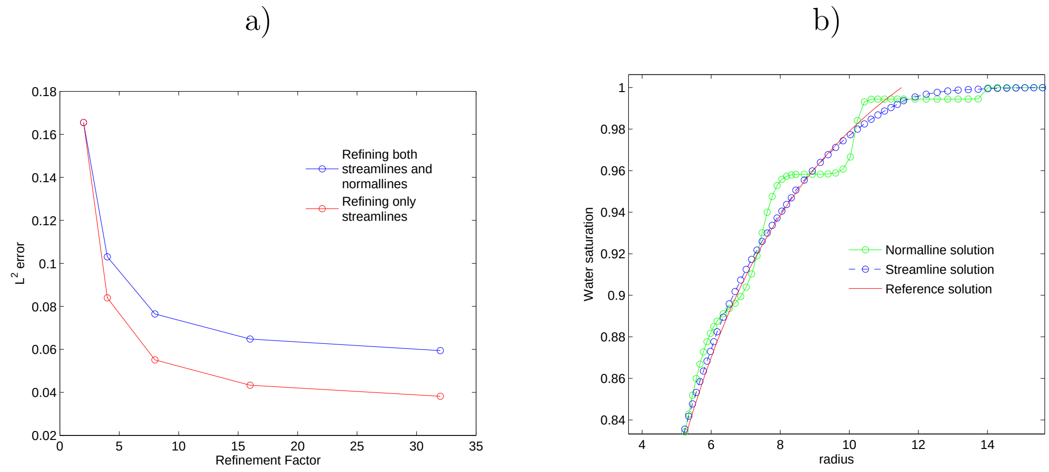


Figure 6.12: a) ℓ^2-error at time $t = 10^5$s for a grid with 10 streamlines and 10 normallines. b) Discrete refined solutions and reference solution near $S = 1$ at $t = 10^5$s. Refinement factor for b) is 16 for both streamlines and normallines. The global time step is 1000s.

Consider figure 6.12b. This figure shows the refined solution with refinement factor 16 along streamlines and normallines, as well as the reference solution for water saturations close to 1. When the mean solution for an intersection point is computed after solving along the streamlines, the solution is projected onto the normallines based on the normalline solution at the previous time step. When the refined solution is constant, as is the case when the mean water saturation for an intersection point is equal to 1, the solution in the refined grid points are the same as the mean solution in the intersection point. This, in combination with the degeneracy of the parabolic term makes the shape of the normalline solution imply artificially high diffusive numerical fluxes in the direction of flow. Notice how the shape of the numerical solution on streamlines follow the reference solution to a much higher degree than the solution on normallines in figure 6.12b.

Figure 6.13 shows the same case as figure 6.12a, only with another reconstruction method for the normallines. For this case, instead of using the scaled mapping we use a linear reconstruction, assuming that solution values along normallines vary linearly between two intersection points. No information regarding the refined solution at the previous time step is used. This will take care of the stair-case shape of normalline solutions in the neighborhood of $S = 1$, and hopefully approximate the reference solution to a higher degree. This method is however not mass-conserving, possibly inducing errors.

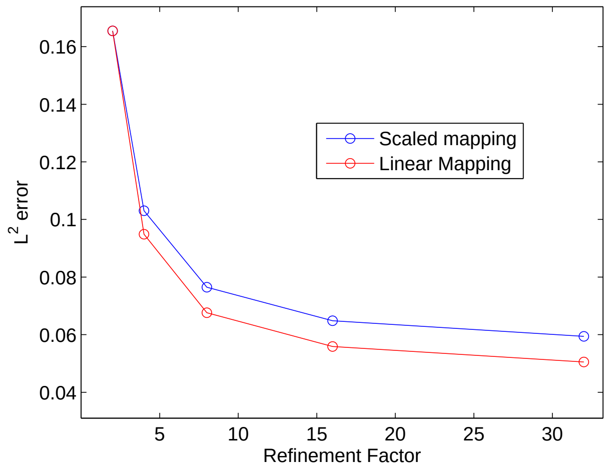


Figure 6.13: Linear and scaled reconstructions on normallines for the same setup as above. Scaled reconstruction used on streamlines.

Here we see that using the linear reconstruction of normalline solutions gives a better approximation to the reference solution, as perhaps anticipated. However, using no refinement of normallines at all is still a better alternative. The tests resulting in figures 6.12 and 6.13 show that the accuracy of the streamline/normalline method is highly sensitive to the methods used for the reconstruction of refined solutions on normallines. More sophisticated methods should be developed, perhaps adaptive methods using different reconstruction methods for different regions depending on the local degree of degeneracy, solutions at previous time steps and so on.

Chapter 7

Simulation of Groundwater Contamination

In this chapter we will construct a more realistic, physically motivated grid than the one in the previous chapter, and solve the equations that arise from groundwater contamination. We will compare the streamline/normalline methodology to the approach of mapping onto a background grid to take care of diffusive/dispersive terms.

As motivation, consider a factory that as a part of its production process ends up with contaminated water. The factory is given permission to inject this water into a confined aquifer, or injects it anyway, without permission. If there are any drinking water wells connected to the same aquifer, simulations of the spreading of contaminant is important to determine when and if the contaminant reaches the drinking water well. The factory could be polluting the drinking water even if it injects its contaminated water into a different aquifer than the one that holds the drinking water well. The two aquifers could for instance be connected through fractures or leaky abandoned wells.

7.1 Construction of the Test Case

However the contaminated water is allowed to reach the drinking water, its source will from now on be called an injector. Our setup will hence consist of an injector that is injecting polluted water into a confined aquifer and a drinking water well, illustrated in figure 7.1.

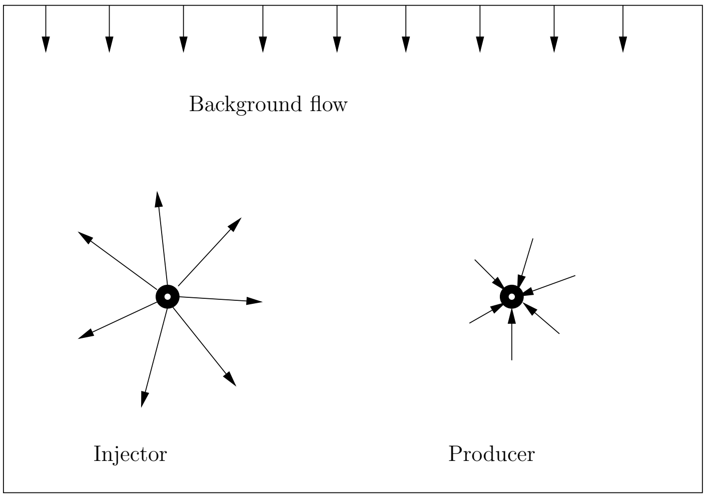


Figure 7.1: An injector and a producer in a confined aquifer with background flow. The boundaries are open.

As in the previous chapter, analytical expressions for the velocity field will be used. To find this we first put the Darcy velocity into the law of mass conservation for an incompressible fluid. Let q_1 be the injection rate of the injector, considered a point source, and let q_2 be the production rate of the producer, considered a point sink,

$$\nabla \cdot \left(-\frac{K}{\mu} \nabla p \right) = \frac{q_1 - q_2}{\rho} \tag{7.1}$$

Next, we decompose the pressure field into a solution of the homogeneous equation and two particular solutions, $p = p_h + p_1 + p_2$, where p_h, p_1 and p_2 are the solutions of the following equations:

$$\nabla \cdot \left(-\frac{K}{\mu} \nabla p_h \right) = 0 \tag{7.2}$$

$$\nabla \cdot \left(-\frac{K}{\mu} \nabla p_1 \right) = \frac{q_1}{\rho} \tag{7.3}$$

$$\nabla \cdot \left(-\frac{K}{\mu} \nabla p_2 \right) = -\frac{q_2}{\rho} \tag{7.4}$$

Then p will be a solution of equation (7.1). Define also $\mathbf{v} = \mathbf{v}_h + \mathbf{v}_1 + \mathbf{v}_2$. Then

$$\mathbf{v}_h = -\frac{K}{\mu} \nabla p_h \tag{7.5}$$

where ∇p_h is the background hydraulic gradient. This is the velocity originating from the background flow. The solutions of equations (7.3) and (7.4) are the same as they would have been if there was no background hydraulic gradient and hence no background flow. This means that the velocities $\mathbf{v}_1$ and $\mathbf{v}_2$ are inversely proportional to the injector and the producer, respectively, and that the steady-state velocity field is given by

$$\mathbf{v} = -\frac{K}{\mu}\nabla p_h + \frac{C_1}{r_1}\mathbf{e}_{r1} + \frac{C_2}{r_2}\mathbf{e}_{r2} \tag{7.6}$$

where r_1 and r_2 are the distance from the injector and the producer respectively, and $\mathbf{e}_{r1}$ and $\mathbf{e}_{r2}$ are the radial unit vectors pointing from the injector and the producer respectively. C_1 and C_2 are constants determined by the injection and production rates. Note that all 3 terms in $\mathbf{v}_h$ are irrotational, assuring that $\nabla \cdot \mathbf{n} = 0$ if we let the normal field $\mathbf{n} = R\mathbf{v}$ where R is the 90 degree counter clockwise rotation matrix.

7.1.1 Parameter Setup

To emphasize the difference in the methods of handling cross-streamline diffusive effects, we let the background Darcy velocity, injection and production rates be such that no streamline starting in the injector ends in the producing well. Then the concentration of contaminant that reaches the producer is expected to be highly dependent on the numerical methods used to capture transverse dispersion since cross-streamline effects are the only effects that could lead to contamination of the produced water.

To achieve this, we set as in the previous chapter the porosity ϕ to 0.15. The injection rate is set to 41.9m^3/day, and we let the producing well be placed 50 meters from the injection well, and let it's production rate be 20% of the injection rate. Let the background Darcy velocity be $\mathbf{v}_h = 3.13 \cdot 10^{-6}$m/s, resulting in streamline-normalline grids as in figure 7.2. Molecular diffusion is for simplicity neglected. The concentration of contaminant in the injected water is set to $c_0 = 10^{-4}$. We further consider two different cases for the longitudinal and transverse dispersivities α_l and α_t. In one case we let $\alpha_l =$ 1m and $\alpha_t = 0.1$m, and in the other we let $\alpha_l = 10$m and $\alpha_t = 1$m.

7.2 Simulations

In this section we solve equation (3.11) simplified according to the parameter setup and decomposed to a part along streamlines,

$$\frac{\partial c}{\partial t} + \frac{\partial c}{\partial \tau} - \phi \frac{\partial}{\partial \tau}\left(\|\bar{\mathbf{v}}\| d_l \frac{\phi}{\|\mathbf{v}\|^2}\frac{\partial c}{\partial \tau}\right) = 0 \tag{7.7}$$

and a part along normallines,

$$\frac{\partial c}{\partial t} - \phi \frac{\partial}{\partial \nu}\left(\|\bar{\mathbf{v}}\| d_t \frac{\phi}{\|\mathbf{n}\|^2}\frac{\partial c}{\partial \nu}\right) = 0 \tag{7.8}$$

The equations are solved using 1D finite volumes as described in Chapter 5. The injected concentration is implemented as a boundary condition for the streamlines, letting $c = c_0$ at the beginning of each streamline starting in the injection well. We also consider taking dispersion on a background grid, solving

$$\frac{\partial c}{\partial t} + \frac{\partial c}{\partial \tau} = 0 \tag{7.9}$$

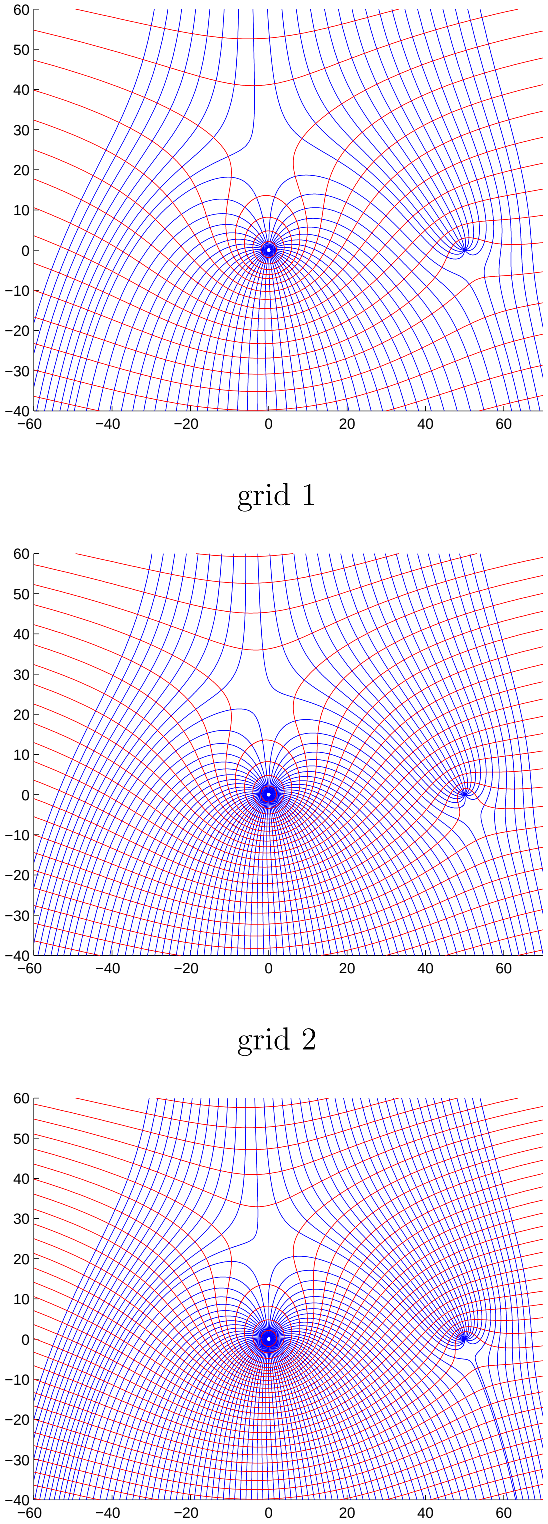
along streamlines, and solving

$$\frac{\partial c}{\partial t} - \nabla \cdot (\widetilde{D} \cdot \nabla c) = 0 \tag{7.10}$$

on a background Cartesian grid, using finite differences and a 9-point stencil, where central differences are used to approximate the derivatives in the second term of (7.10). The Cartesian grid has the same grid spacing - $\Delta x = 2$m in all simulations.

Figure 7.2 shows the three different streamline/normalline grids used for the simulations. There is no refinement along streamlines or normallines. This means that the grid points used for the solution of the one-dimensional equations are the intersection points between streamlines and normallines. The global time step is the same for all simulations - $\Delta t_G = 10^4$s.

In figure 7.3, 2D-plots of the concentration near the injector is shown for 3 different dime steps and both solution methods. While this kind of plot is useful when visualizing flow, it is not the best tool when it comes to comparing the different solution methods. We have implemented a global error estimate, taking the relative ℓ^1-norm of the difference between two solutions, arranging the solution in the different grid points in a vector. Figure 7.4 shows the ℓ^1 difference of the different solution methods for grid 1 and 2 relative to the streamline/normalline method on grid 3. Finally, in figure 7.5 we show the concentration in the producing well as a function of time for the different solution methods, grids and dispersivities.



grid 1

grid 2

grid 3

Figure 7.2: The three different streamline-normalline grids used for the simulations in this chapter. Blue lines are the streamlines, red lines are the normallines.

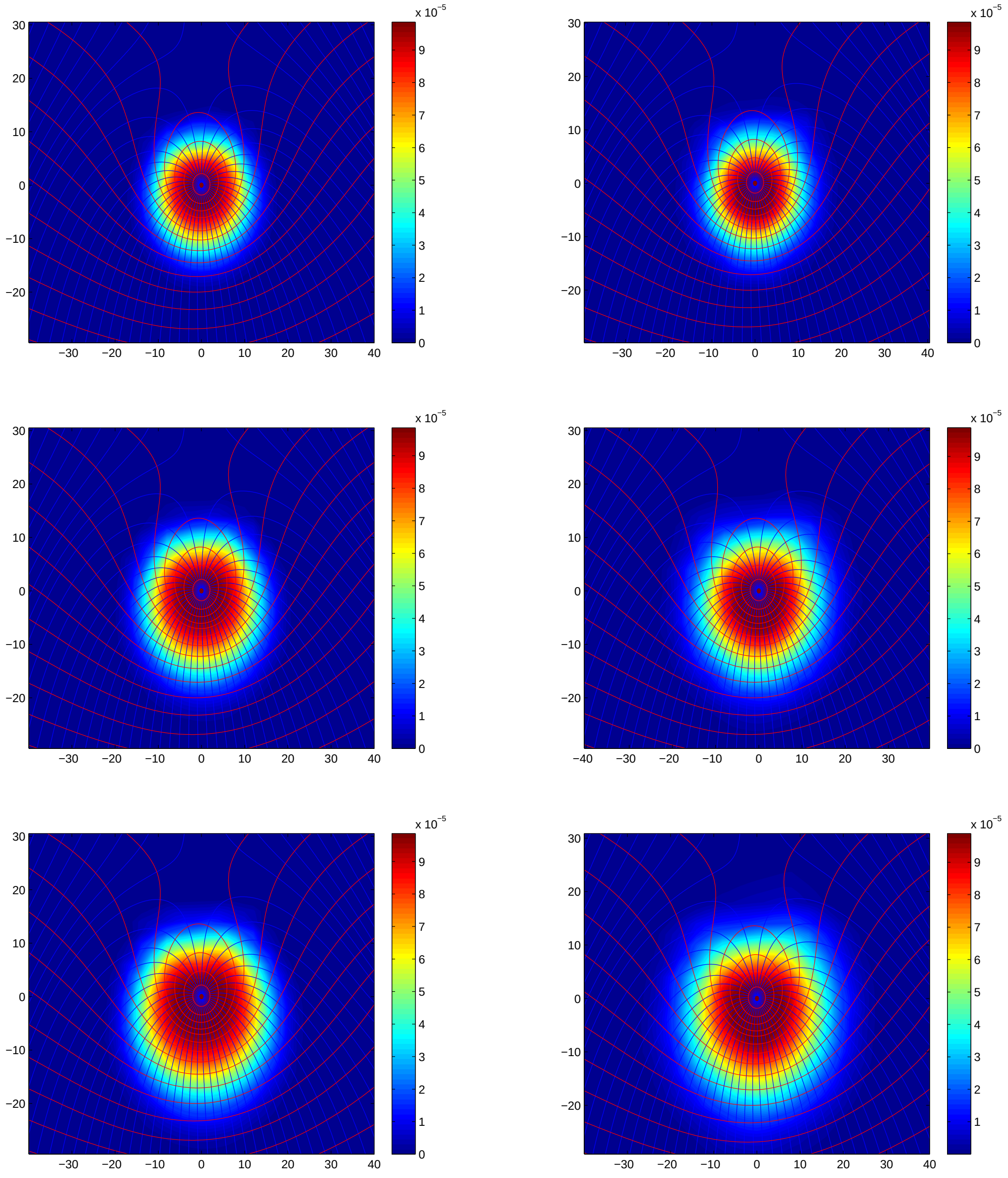


Figure 7.3: Simulations with dispersion on a background grid (right) and dispersion along normallines (left). Longitudinal dispersivity $\alpha_l = 1$m, and transverse dispersivity $\alpha_t = 0.1$m. The screenshots are at times $t = 1 \cdot 10^5$s, $t = 1.5 \cdot 10^5$s and $t = 2 \cdot 10^5$s

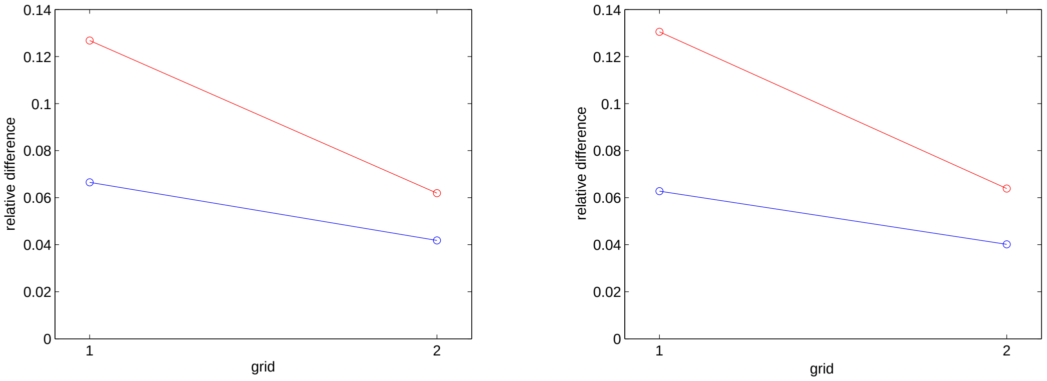


Figure 7.4: Relative ℓ_1 difference for different grids and dispersivity. Left: lowest dispersivity, right: highest dispersivity

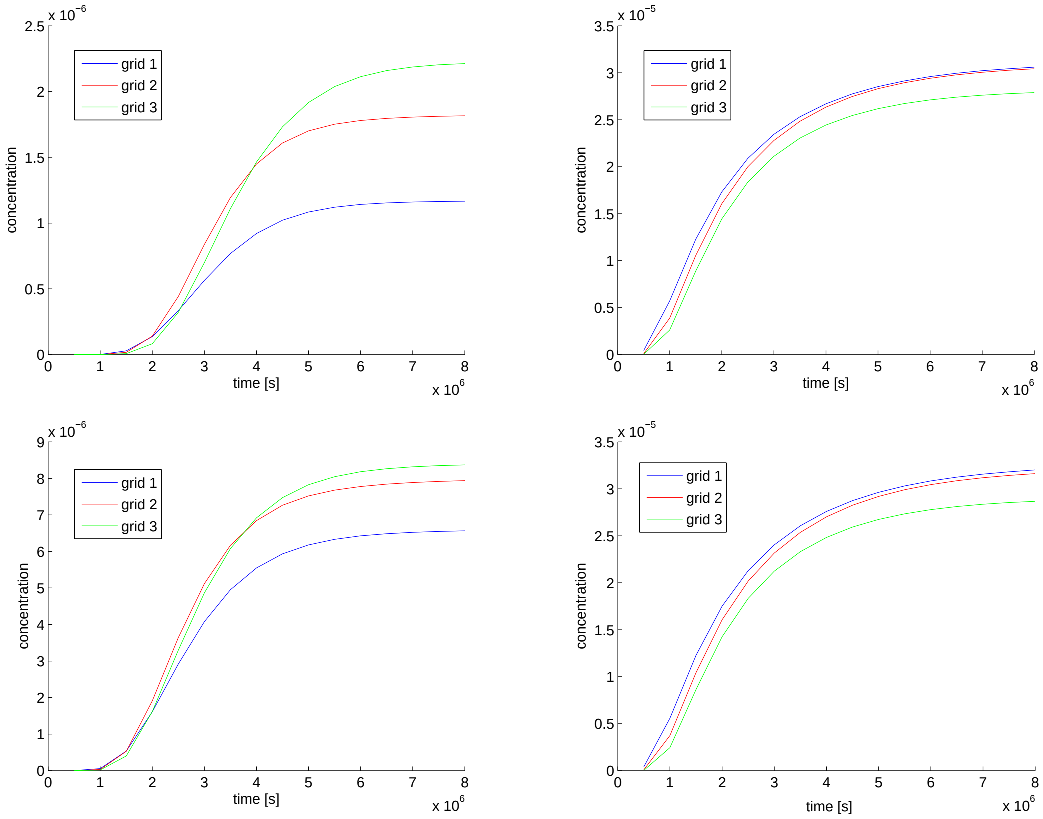


Figure 7.5: Concentration in producing well as a function of time for different grids. Left: dispersion along normallines. Right: dispersion on background grid. Top: $\alpha_l = 1\text{m}$, $\alpha_t = 0.1\text{m}$. Bottom: $\alpha_l = 10\text{m}$, $\alpha_t = 1\text{m}$.

7.2.1 Discussion

Figure 7.3 shows plots of the concentration at different times on the first of the grids in figure 7.2. In this case the longitudinal dispersivity is $\alpha_l = 1$m, and transverse dispersivity $\alpha_t = 0.1$m. The figure shows not much difference in the concentration profiles, but the mapping between streamlines and the background grid seems to be a bit more diffusive than the streamline/normalline method which has a sharper concentration front. The global error estimates in figure 7.4 show that the results are consistent, and that the solution appears to be converging globally when the number of streamlines and normallines increase.

Global error estimates may not be the right tool to use on these kind of problems. The numerical diffusive smearing of the solution introduced by the mapping between streamlines and the background grid leads to an overestimation of the concentration in the producing well. First of all, consider the two plots on the right hand side of figure 7.5. These are from simulations with dispersion on the background grid, and we see that the plots are nearly identical. Although the method involving background grids appeared to be convergent and correct, the numerical diffusion associated with the mapping is much greater than the physical dispersive effects that we are trying to capture. The standard method, as implemented here simply cannot be used to give any useful or predictable information regarding the producing well contaminant concentration.

The two plots on the left hand side of figure 7.5 are from simulations with dispersion along normallines. The plots show that the concentration in the producing well is much lower than it was when dispersion was taken on the background grid, and that it takes more time for the contaminant to reach the well. Furthermore they show that the concentrations are highly grid-dependent. For the case with $\alpha_t = 0.1$m, simulation on grid 1 predicts approximately half of the concentration that grid 3 predicts. This is because the problem is constructed to be a difficult one - we are trying to predict the concentration in a singularity. What enters the producing well or not is by default sensitive to streamline/normalline densities in the bordering regions between contaminated and clean water. Smaller dispersivity means higher grid sensitivity. When comparing the two figures on the left hand side of figure 7.5, we see that the case when $\alpha_t = 1$m is more easily captured than the case with the least transverse dispersivity, even if none of the solutions can be called a converged solution. The sensitivity of the solutions to grid changes is still far less than the sensitivity to changes in the physics of the problem, as opposed to the standard method.

In conclusion, we have shown that while the standard approach to including dispersion in streamline methods is convergent in a global sense, it is unable to give any useful information regarding the well-to-well transfer of contaminants. The streamline/normalline method presented significantly reduces the numerical diffusion by eliminating mapping errors.

Chapter 8

Summary and Conclusion

In this thesis we have studied and implemented a solution approach to advection-diffusion equations based on reformulating the equations in time-of-flight coordinates along streamlines and normallines. By doing a dimensional splitting of the resulting equations, one-dimensional equations was solved along streamlines and normallines individually for both one-phase and two-phase flow.

The basic equations was derived in chapters 1-3, and the streamline/ normalline method was developed in chapter 5. In Chapter 6 we compared solutions of a vertically averaged saturation equation using this streamline/normalline method to a reference solution of CO_2 injection in confined aquifers. We saw that the solutions approached the reference solution when the number of streamlines and normallines was increased, as well as when the one-dimensional grids was further refined along streamlines. However, we found that refining the one-dimensional grids along normallines had the opposite effect on the accuracy, showing at least for normallines the need for more sophisticated mappings than the scaled mapping outlined in chapter 5. We also saw that the method is non-conservative when the velocity field is rotational. In the case of rotational flow, the scaling (3.21) of the normal field should be considered.

In Chapter 7 we ran some simulations of groundwater contamination in a confined aquifer with background flow, a source of contaminant and a drinking water well. We compared the streamline/normalline method to the method of mapping from streamlines onto a background Cartesian grid for the calculation of dispersion. We saw that for the test case constructed and parameters used, the standard method for the inclusion of dispersion in streamline methods failed completely in giving any reliable results regarding the well-to-well transfer of contaminants. Using normallines for the calculation of transverse dispersivity eliminates the numerical diffusion associated with the mapping to and from background grids, thus giving more predictable results.

Although streamline simulation based on time-of-flight coordinates is widely used for 3-dimensional problems, the method discussed in this thesis is restricted to 2-dimensional problems. In 3D one would have to consider some highly complex geometric surfaces as ”normallines”. In 2D however, the method could very well serve as an alternative or supplement to existing methods.

Appendix A

Terminology and Notation

α : **Subscript denoting phase**. When writing for instance S_α, one means "the saturation of phase alpha". For two-phase flow, α could be wetting (w) or non-wetting (nw). We will not be considered much with **wettability**, but think of a drop of water on a table and a drop of oil the same size as that of water. The drop of water will be covering a bigger area of the table than the drop of oil. We say that water is wetting relative oil.

ϕ : **Porosity**. Rock parameter. Defined as the ratio between the volume of empty space and the volume of solid rock (the solid rock is called the *matrix*). However, this definition is not very practical, since there may well be volumes of empty space being isolated from the flow. ϕ is therefore usually interpreted as the effective porosity. In this interpretation one considers isolated empty space as a part of the matrix.

ρ : **Mass density**. Phase variable. In incompressible fluids the mass density is constant. Adding compressibility to the system complicates things, and one needs to add thermodynamical equations of state in order to incorporate the compressibility. We consider only the case of incompressible fluids, where the mass density is given.

S : **Saturation**. Phase variable. Portion of available space filled with the respective phase. The following relation always holds: $\sum_\alpha S_\alpha = 1$. Residual saturations need to be taken into account, since it is generally not possible to get for instance all the oil out in a drainage process (reduction of oil saturation in an oil/water system is called *drainage*, while increasing the oil saturation is called *imbibition*). This is highly dependent of pore geometries, and is usually empirically determined.

$\mathbf{v}$: **Velocity**. Phase variable. Darcy velocity is denoted $\mathbf{v}$ and average linear velocity is denoted $\bar{\mathbf{v}}$

K : **Permeability**. Rock parameter. In groundwater flow it is sometimes called the hydraulic conductivity. The permeability is usually related to the porosity, but higher porosity doesn't necessarily imply higher permeability. This is because the permeability is determined by not only available volume, but also pore sizes and geometries. These again rely on faults, fractures, different geolocial layers and so on. The permeability is therefore in it's most general an anisotropic, heterogenous tensor. Determining these rock parameters is crucial in real field cases, and is a major research issue. Traditionally it is done by seismic interpretation, core samples or well data history matching.

k_r : **Relative permeability**. Variable depending on rock and saturations. What the effective permeability for a phase is given a certain amount present of the other.

p : **Pressure**. Phase variable. Due to interfacial tension between the different phases and the rock matrix, the pressures are higher in phases with lower wettabilities and lower in phases with higher wettabilities [17]. The differences in two phase pressures are called the **capillary pressure** between the two respective phases, so in a three phase system we need two capillary pressure relations. Modelling these capillary pressures is a research subject in itself, and empirically determined functions are preferred when available.

Capillary pressure is at least a function of saturation, but experiments have shown that it is also dependent of the flow process itself. For example, the cappilary pressure between oil and water is higher for an imbibitions process than it is for a draining process. This effect is called *hysteresis*. See for instance [10] for more details.

μ : **Viscosity**. Phase variable. The viscosity is a measure of how “thick“ a fluid is. Syrup has high viscosity, water has low. We only consider the case of constant viscosity.

Appendix B

Streamline Tracing

If one knows the cell edge fluxes in a cartesian grid, it is in principle straightforward to trace streamlines using Pollock's method [18]. For simplicity, assume stationary flow, meaning that streamlines and particle paths coincide.

Pollocks method tells us how to trace a streamline inside a grid cell, given an entry point P_0. A particle in the point P_0 at time $t = 0$ follows the path given by $\mathbf{r}(t) = x(t)\mathbf{i} + y(t)\mathbf{j}, \mathbf{r}(0) = P_0$. Scale and move the cell if necessary so that it becoems $[0, 1] \times [0, 1]$, and let the time start at $t = 0$. Assume that the x-component of the velocity field, $u(x, y)$ varies linearly in x-direction, and assume that the y-component , $v(x, y)$ varies linearly in y-direction. We then have

$$\frac{dx}{dt} = u(x, y) = u(x) = G_x x + u_A = (u_B - u_A)x + u_A \tag{B.1}$$

$$\frac{dy}{dt} = v(x, y) = v(y) = G_y y + v_C = (v_D - v_C)y + v_C \tag{B.2}$$

Here A, B, C, D are respectively the left, right, bottom and top cell edges. The assumptions above makes the equations uncoupled, and if $G_x, G_y \neq 0$ we get the following expressions for the time t:

$$t = \frac{1}{G_x} = \ln\left(\frac{G_x x(t) + u_A}{G_x x(0) + u_A}\right) \tag{B.3}$$

$$t = \frac{1}{G_y} = \ln\left(\frac{G_y y(t) + v_C}{G_y y(0) + v_C}\right) \tag{B.4}$$

If $G_x, G_y = 0$ the expressions for t are simpler:

$$t = \frac{x(t) - x(0)}{u_A} \tag{B.5}$$

$$t = \frac{y(t) - y(0)}{v_C} \tag{B.6}$$

Now we have 3 unknowns, $t, x(t), y(t)$, but only 2 equations. t may still be found if we assume $t \in [0, \tau]$, and let τ be the smallest time we get by calculating the time a particle needs to travel from the entrance point to each of the 3 other boundaries.

For the left and right edge, let $x(t) = 0, x(t) = 1$ in (B.3) or (B.5), and for (B.4) or (B.6), let $y(t) = 0, y(t) = 1$. Now, let τ be the smallest of these times, so that the particle will exit the cell through the point $(x(\tau), y(\tau))$, which is then taken as the entry point for the cell which the particle enters.

When τ now is found, equations (B.3) or (B.5) can be solved for $x(t)$, and (B.4) or (B.6) can be solved $y(t)$. If $G_x, G_y \neq 0$:

$$x(t) = \frac{(G_x x(0) + u_A)e^{G_x t} - u_A}{G_x}, \qquad t \in [0, \tau] \tag{B.7}$$

$$y(t) = \frac{(G_y y(0) + v_C)e^{G_y t} - v_C}{G_y}, \qquad t \in [0, \tau] \tag{B.8}$$

If $G_x, G_y = 0$:

$$x(t) = x(0) + u_A t, \qquad t \in [0, \tau] \tag{B.9}$$

$$y(t) = y(0) + v_C t, \qquad t \in [0, \tau] \tag{B.10}$$

This way streamlines can be traced from an arbitrary starting point, all the way to the edge of the domain, or until it reaches a well. If there are production wells inside a cell, all the possible τ will be infinite, and the particle naturally cannot exit the cell.

This method has been extended to be used on irregular grids as well, by introducing transformations that injectively transforms the cells and their interior into unit cubes.

The figures below show a quarter-of-a-five well setup with a nonhomogeneous permeability field along with the corresponding streamlines. The pressure equation is solved using software following [1].

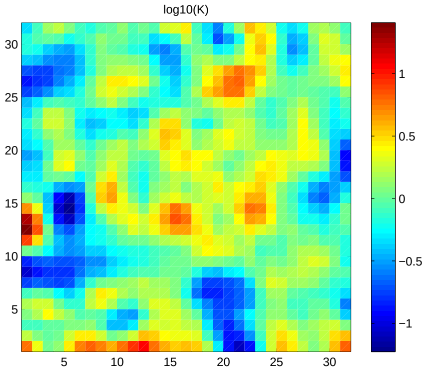


Figure B.1: Nonhomogenous permeability field

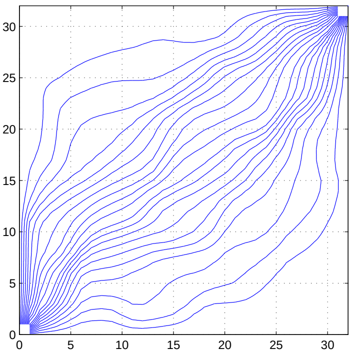


Figure B.2: And the corresponding streamlines

Appendix C

Application of Built-in Matlab Code

When implementing the different methods described in this thesis, a number of built-in Matlab codes have been used. This appendix explains some of the nontrivial ones.

ODE solvers

When finding the streamlines for the analytically given, stationary velocity or normal fields, we solve an equation on the form

$$\frac{\partial \mathbf{x}}{\partial \tau} = \mathbf{v}(\mathbf{x}) \tag{C.1}$$

using `ode45`, a fourth order Runge-Kutta solver. This is a one-step method, and can be written [11]

$$\mathbf{x}(\tau + h) = \mathbf{x}(\tau) + \frac{1}{6}(\mathbf{v}_1 + 2\mathbf{v}_2 + 2\mathbf{v}_3 + \mathbf{v}_4)) \tag{C.2}$$

where

$$\begin{aligned} \mathbf{v}_1 &= h\mathbf{v}(\mathbf{x}) \\ \mathbf{v}_2 &= h\mathbf{v}(\mathbf{x} + \tfrac{1}{2}\mathbf{v}_1) \\ \mathbf{v}_3 &= h\mathbf{v}(\mathbf{x} + \tfrac{1}{2}\mathbf{v}_2) \\ \mathbf{v}_4 &= h\mathbf{v}(\mathbf{x} + \mathbf{v}_3) \end{aligned}$$

After this equation has been integrated for a streamline, we end up with a streamline polygon. So another thing worth mentioning here is how we find the intersection between a streamline and a normalline polygon

Finding Intersection Points

There is no built-in matlab code to find the intersection between two polygons. However Matlab Central File Exchange (http://www.mathworks.com/matlabcentral/fileexchange) offers a function, `curveintersect` written by Sebastian Hölz of the Leibniz Institute of Marine Sciences.

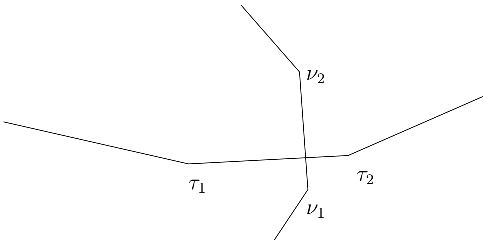


Figure C.1: Intersection of a streamline polygon and a normalline polygon

The time-of-flight coordinates corresponding to the intersection points are then interpolated by assuming that the time-of-flight varies linearly between two polygon points.

Mapping Data to and from Background Grids

When mapping data to and from background grids, the data given along streamlines at discrete time-of-flights need to be interpolated. For mapping from streamlines to a Cartesian grid, we use `griddata`. When going from the Cartesian grid to the streamlines, we use `interp2`. `griddata` is based on a *triangulation* of the interpolation points. In the book by Kincaid and Cheney [11], a triangulation is described as a collection of triangles $T_1, T_2, \ldots, T_m$ where the following rules must be satisfied:

- Each interpolation node must be the vertex of some triangle T_i
- Each vertex of a triangle in the collection must be a node
- If a node belongs to a triangle, it must be the vertex of that triangle

Matlab uses Delaunay triangulation as default. This has the additional property that

- No node is inside the circumcircle of any triangle.

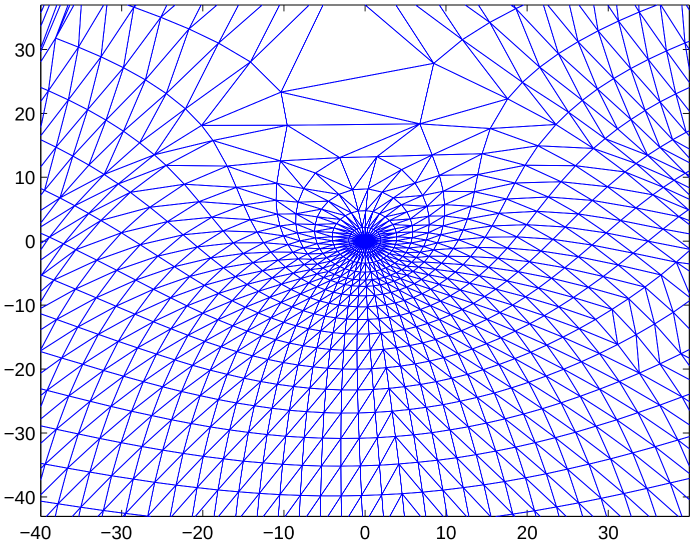


Figure C.2: Delaunay triangulation of intersection points.

Figure C.2 shows the Delaunay triangulation of the grid produced by intersection points in chapter 5. The solution is taken to be linear in each triangle, and such that it is continous. The solution values at the Cartesian grid points intersects this piecewise linear solution.

`interp2` uses the values in the Cartesian grid points to interpolate values along streamlines. It assumes that the solution is piecewise linear and continous. The solution values along the streamlines intersects this piecewise linear solution.

Bibliography

[1] Jørg E. Aarnes, Tore Gimse, and Knut-Andreas Lie. An introduction to the numerics of flow in porous media using matlab. SINTEF ICT, Dept. of Applied Mathematics, Oslo.

[2] Ivar Aavatsmark. Bevarelsesmetoder for hyperbolske differensial-ligninger. University of Bergen, 2004.

[3] Jacob Bear. On the aquifer's integrated balance equations. *Advances in Water Resources*, 1:15–23, 1977.

[4] Zhangxin Chen, Guanren Huan, and Yuanle Ma. *Computational Methods for Multiphase Flows in Porous Media.* SIAM, 2006.

[5] Olaf A. Cirpka, Emil O. Frind, and Rainer Helmig. Numerical methods for reactive transport on rectangular and streamline-oriented grids. *Advances in Water Resources*, 22:711–728, 1999.

[6] Olaf A. Cirpka, Emil O. Frind, and Rainer Helmig. Streamline-oriented grid generation for transport modelling in two-dimensional domains including wells. *Advances in Water Resources*, 22:697–710, 1999.

[7] Magne S. Espedal and Kenneth H. Karlsen. Numerical solution of reservoir flow models based on large time step operator splitting algorithms. *Lecture Notes, Department of Applied Mathematics*, 1999.

[8] R. Allan Freeze and John A. Cherry. *Groundwater.* Prentice Hall, Englewood Cliffs, NJ 07632, US, 1979.

[9] H. Hægland, H.K. Dahle, G.T. Eigestad, K.-A. Lie, and I. Aavatsmark. Improved streamlines and time-of-flight for streamline simulation on irregular grids. *Advances in Water Resources*, 30:1027–1045, 2007.

[10] Bjørn Ove Heimsund. *Mathematical and Numerical Methods for Reservoir Fluid Flow Simulation.* PhD thesis, University of Bergen, Norway, 2005.

[11] David Kincaid and Ward Cheney. *Numerical Analysis: Mathematics of Scientific Computing.* Brooks/Cole 3rd edition, 2001.

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

[13] Randall J. LeVeque. *Numerical Methods for Conservation Laws.* Birkhuser, 1998.

[14] B. Mallison, M. Gerritsen, and S. Matringe. Improved mappings for streamline based simulation. *SPE Journal*, 2006.

[15] Morris Muskat. *Physical Principles of Oil Production.* McGraw-Hill, 1949.

[16] Jan M. Nordbotten and Michael A. Celia. Similarity solutions for fluid injection into confined aquifers. *Journal of Fluid Mechanics*, 561:307–327, 2006.

[17] Øystein Pettersen. *Grunnkurs i reservoarmekanikk.* University of Bergen, Bergen, Norway, 1990.

[18] David W. Pollock. Semianalytical computation of path lines for finite-difference models. *Groundwater*, 26:743–750, 1988.

[19] Marco R. Thiele. Streamline simulation. *8th International Forum on Reservoir Simulation*, 2005.