Simulation of salt migrations in density dependent groundwater flow

E.S. van Baaren

Master's Thesis
Applied Mathematics

June 2007

Thesis Committee:
Dr.ir. F.J. Vermolen (Delft University of Technology)
Prof.dr.ir. C. Vuik (Delft University of Technology)
Dr.ir. W.J. Zaadnoordijk (Royal Haskoning)
Prof.dr.ir. A.W. Heemink (Delft University of Technology)

Abstract

In the Netherlands, the effects of the changing climate become more and more visible; the rain falls in higher intensities, the sea level rises and the maximum discharge of rivers increases. The question rises what the effects will be of these phenomena for the salt migration in the groundwater underneath the polders near the coast. The problem description of this thesis is to investigate the possibilities of modelling salt migrations in density dependent groundwater with modelling environment Triwaco.

The movement of the groundwater and the transport of solutes in the subsurface are coupled processes and the two equations must be solved jointly. This coupling starts with the flow equation which calculates the freshwater head for a given density pattern, then Darcy's law transforms the freshwater heads into velocities of the groundwater which are given to the transport equation. The transport equation determines the new densities for these velocities and returns these values to the flow equation. This process is called the coupled process and visiting both equations once is called a cycle. A new cycle can be made by repeating the process.

The flow equation is already solved in Triwaco and uses a finite element method for the simulation of groundwater flow in the lateral (2D) direction. Communication between aquifers (vertically) is described with a 1D finite difference method. The numerical method used to solve the transport equation has to fit easily in this used method for the flow equation.

For the two dimensional advective, dispersive and diffusive transport of salt, research is done on four numerical methods; the Standard Galerkin Approach, the SUPG pure advection algorithm by Mizukami, the SUPG classical upwind method and the Mizukami Hughes algorithm. For the solute transport between the aquifers (third dimension) a finite difference method is used in the simulations.

Numerical experiments are done for two ways of coupling the transport and flow equation. In the first coupling both the flow and transport equation are solved in Matlab. The benchmark problem of the rotating brackish zone works well with the developed software in Matlab. In the second coupling the flow equation is solved with Triwaco and coupled to the transport equation solved in Matlab. When some problems with calculating the velocities in Triwaco are solved, no big problems are expected for simulation of density dependent groundwater flow or salt migrations with Triwaco. The advantage of the flexible finite element grid used in Triwaco can now also be used for the salt transport of density dependent groundwater flow.

Contents

Chapter 1

Introduction

In the Netherlands, the effects of the changing climate become more and more visible; the rain falls in higher intensities, the sea level rises and the maximum discharge of rivers increases. The question rises what the effects will be of these phenomena in the future. An important issue is for example the salt concentration in the groundwater that will change due to these effects and that will influence the groundwater flow. It is important to be able to answer this question on a regional scale. For a farmer it is important to know whether ditches bordering his fields become too salt to be used as drinking water for his cattle and for the waterworks it is interesting to know if they can still use a certain source for tap water in ten years [1, 2].

1.1 Hydrology

The hydrological cycle is shown in Figure 1.1. From this cycle it can be seen that a number of situations can cause salt water intrusion or extraction from the groundwater. One of these situations is the sea level rise which can change the boundary between the salt groundwater and the fresh groundwater. The IPCC [3], the climate panel of the United Nations, expects a sea level rise between the 18 and 59 centimeters till 2100. The KNMI [4] on the other hand predicted a rise of $35 - 85$ centimeters for this century in the Netherlands, mainly due to the melting of glaciers and ice caps, the change of the discharge of rivers and the change in temperature.

Another cause for the change in salt concentration are the alternating periods with much precipitation an no precipitation due to the climate change. During periods with much precipitation the salt groundwater can dilute or the boundary between the salt and freshwater can move and during dry periods large quantities of the freshwater will evaporate. On regional scale there will not only be natural influences in this cycle. In Figure 1.2 it can be seen that wells may have a large influence on the density and flow of the water.

Another problem is the drop of the ground in the Netherlands due to for example the winning of natural gas, the winning of salt and groundwater-abstractions. According to Figure 1.3 [6] the drop of the ground will be 80 centimeters between 2007 and 2050 in some regions in the Netherlands.

The influences of these changes are mainly noticeable for the waterworks

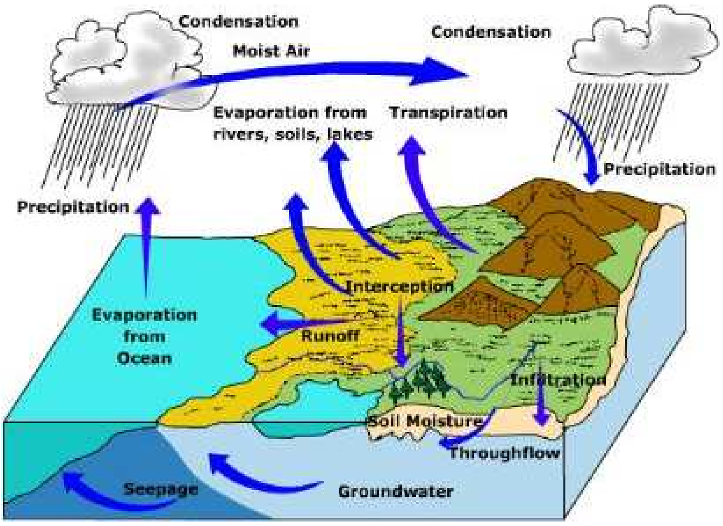


Figure 1.1: Hydrological cycle [5].

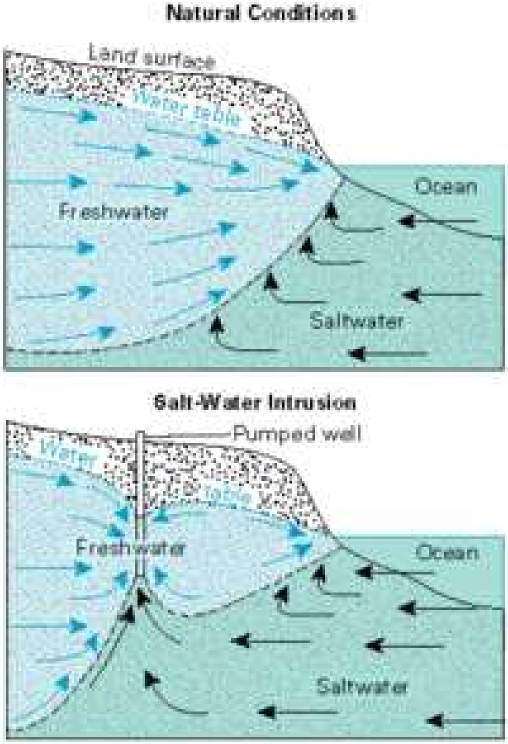


Figure 1.2: A well in the groundwater.

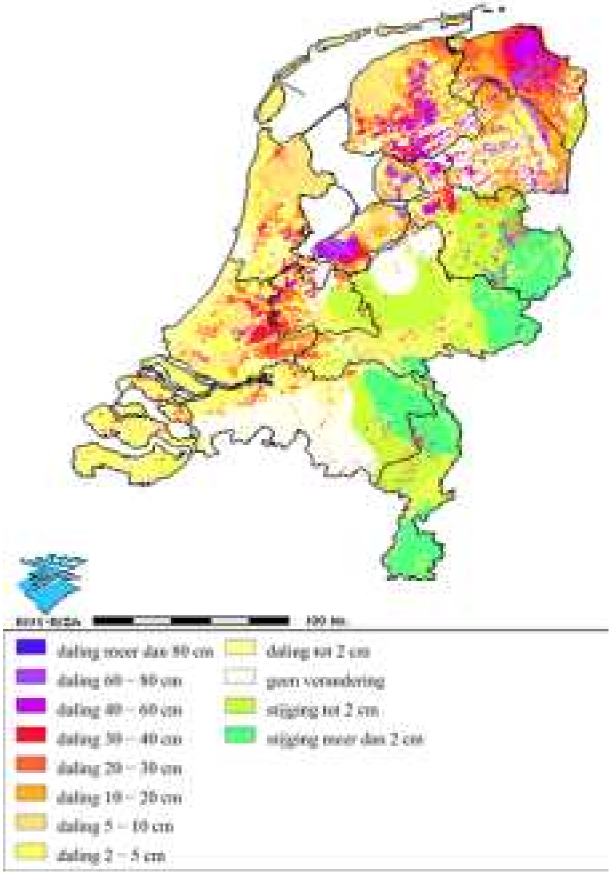


Figure 1.3: The drop of the ground till the year 2100 in the Netherlands will be between the 2 *cm* (green) and 80 *cm* (purple) [6].

and the ecology. The agriculture and horticulture will soon notice that the groundwater becomes more and more brackish, rare plants will become extinct and above a certain concentration, salt water cannot be used for drinking water.

1.2 Geology

In the subsurface aquifers are separated by aquitards, as can be seen in Figure 1.4. An aquifer is a body of rock or sediment that is sufficiently porous and permeable to store, transmit and yield significant quantities of groundwater to wells and springs. It is assumed that aquifers have a relatively small slope. The flow of the water takes place in all directions in the aquifers.

An aquitard is a geologic formation that is not permeable enough to yield significant amounts of water to wells, but on a regional scale can supply significant water to the underlying or overlaying aquifers. In an aquitard only vertical velocity is assumed, the horizontal velocity of the flow is zero. As shown in Figure 1.4, the number of aquitards is assumed to be equal to the number of aquifers minus one. Below the last aquifer an aquiclude can be found, which is an impermeable body of rock that may absorb water slowly but does not transmit it. The first aquifer is only assumed for the model to be the first subsurface layer.

Porosity and permeability are properties of the material of the subsurface. The porosity of a material is the percentage of the volume of that material that can be occupied by water. For example the porosity of soil will be higher than the porosity of rock. The typical porosity of some common sediments and rocks can be found in Table 1.1. The permeability of a geologic formation is its ability to transmit water. There are several factors that affect permeability, including pore size. In general, fine grained sediments will have lower permeability than coarse grained sediments. For some subsurface materials their permeability in descending order can be found in Table 1.2. It is assumed that the porous subsurface is fully saturated with water, no other fluids or gasses that cannot mix with water are present [7, 8, 9, 10].

Table 1.1: Typical porosity of some common sediments and rocks [10].

Material	Porosity
Soil	55%
Gravel and sand	20-50%
Clay	50-70%
Sandstone	5-30%
Limestone	10-30%
Fractured igneous rocks	10-40%

Table 1.2: Some subsurface materials in descending order of permeability [10].

Gravel (High Permeability)
Sand
Silt
Clay
Shale (Low Permeability)

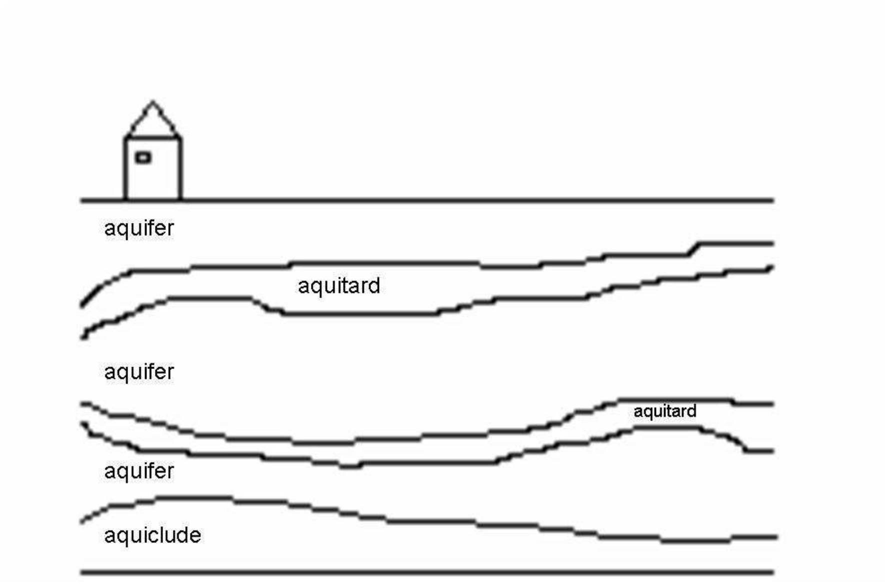


Figure 1.4: Aquifers and aquitards in the subsurface.

1.3 Triwaco

Triwaco is a software package developed by Royal Haskoning. Triwaco offers an integrated modelling environment for modelling flow through the unsaturated and saturated zone, drainage, infiltration and surface water flow. Since the first version in 1984 it has been developed to a modelling environment which is used to support policy development, research and planning in the field of groundwater, surface water and ecology.

The Triwaco package contains a finite element simulator for saturated ground water flow which is called FLAIRS. FLAIRS calculates the groundwater heads and fluxes in a groundwater domain of aquifers and aquitards. The resulting system usually is non-linear due to a non-linear topsystem and aquifer transmissivities which depend on the head.

FLAIRS calculates the lateral flow in aquifers with a two dimensional finite element method. Communication between aquifers (vertically) is described with a 1D finite difference method. The finite element grid is generated by the module TESNET. Boundaries and node densities are inputted into TESNET. In addition it is also possible to enter points or polygons for wells or watercourses and fault zones. Around wells so called 'support circles' can be defined, which are used to automatically create a very dense grid around wells. The streamlines and velocity of the groundwater is calculated with Trace or TraWin. More information about the used methods in Triwaco can be found in Appendix E [11, 12].

1.4 Sustainability

In this project sustainability is an integrated part of the research beside the mathematics, geology and hydrology. The differences in salt concentrations are mainly due to climate changes which result in sealevel rise or different rain fall, and human behaviour which results in a drop of the ground due to the mining of gas or brackish water due to the mining of freshwater by the waterworks. The simulations of salt migration in density dependent groundwater flow can be used for research to these effects of climate changing or human behaviour. The constructed model can also be used during the policymaking; the effects of possible solutions in order to keep an well-balanced environment can be modeled.

A good example to illustrate the necessity of the use of the constructed model is the freshwater lens in the dunes in the Netherlands. In Figure 1.2 can be seen that because of the lower density of freshwater compared to salt water, a freshwater lens exists beneath the dunes on top of the salt water. When freshwater mining takes place, the freshwater lens can be destroyed as can be seen in the second figure of Figure 1.2. The sealevel rise can also influence the shape of the freshwater lens and a combination of mining and sealevel rise can destroy the freshwater lens. Solutions have to be found in order to maintain the freshwater lens beneath the dunes for future generations. One of these possible solutions is an injection of freshwater (rainwater for example) in the dunes in order to repair the lens. With the constructed mathematical model the influence of this possible solution can be modeled during the decision making in order to decide the best location and capacity of the source.

Another sustainable application of the calculation of salt migration in density dependent groundwater flow is the location of a new national park. The construction of a national park takes years and is usually meant to stay there for more than a century. It is expected that the effects of the climate change become more and more visible and this mathematical model can predict the salt concentrations in the groundwater in one century. In this way can be predicted if the chosen location is a good location for the new national park or that some plants are unable to survive when the groundwater becomes to salt.

1.5 Problem description

The problem description of this thesis is to investigate the possibilities of modelling salt migration in density dependent groundwater with modelling environment Triwaco.

Dissolved salt is transported in the ground by the flow of groundwater (advection), molecular diffusion and mechanical dispersion. Dispersive transport describes the dilution or mixing of a solute due to different velocities of groundwater, for example friction in pores, varying travel path lengths and pore sizes. Molecular diffusion is defined as the transport of matter solely by the random motions of individual molecules (Brownian motion).

Advective transport can be caused by density differences of the groundwater. Freshwater will stay on top of salt water because the density of freshwater is less than the density of salt water, but when the salt water is on top of the freshwater, a flow of the water will occur caused by these density differences. Because of these density differences the flow will change and because the flow changes, the density of the water changes again. The process of salt migration in the groundwater is a coupled process between the transport of salt and the flow of groundwater.

The transport of salt is described by the transport equation. In this thesis research is done on numerical methods to solve 3D advective, dispersive and diffusive transport of salt in the groundwater. The structure of this numerical method has to fit in the method used to solve the flow equation in Triwaco.

Chapter 2 gives a derivation of the groundwater flow equation (Section 2.1) and the transport equation (Section 2.2). The coupling between the flow and transport equation is given in Section 2.5.

In Chapter 3 the numerical solution methods are discussed. In Section 3.1 the numerical grid is presented, in Section 3.2 different numerical methods to solve the transport equation spatially are discussed and in the Sections 3.3 and 3.4 the spatial discretization methods for the transport equation are presented. The transport equation is time dependent, the temporal discretization is shown in Setion 3.4. The numerical method that solves the flow equation in Matlab is shown in Section 3.7.

Chapter 4 shows the numerical experiments. First, in Section 4.1 the 2D experiments of different finite element methods for the transport equation are presented. In Section 4.2 some 3D experiments of the combined finite element and finite difference method for the transport equation are given. Section 4.3 presents the results of the rotating brackish zone example, which is a coupling between the transport and flow equation. The flow equation of this coupling is solved in Matlab (Section 4.3.1) as well as in Triwaco (Section 4.3.2). In Section 4.4 numerical experiments of an example are presented. The example consists of a well pumping freshwater from an aquifer fed from below with saline water.

The conclusions, recommendations and discussion are given in Chapter 5. In the appendix a list of symbols, a list of hydrological as well as mathematical definitions and a description of the developed software is presented.

Chapter 2

Model

Salt transport in groundwater can be described by the transport equation and the velocity of the water by the flow equation. In order to calculate the salt transport with time dependent velocities or in order to calculate the velocity of the groundwater with variable and time dependent density, these equations are coupled. The incompressible and laminar Groundwater flow equation is derived in Section 2.1 and coupled to the transport equation in Section 2.2. The description of this coupled process can be found in Section 2.5.

2.1 Groundwater flow

In the used model groundwater flow is expressed in terms of the equivalent freshwater head and fluid density instead of fluid pressure and fluid density. The freshwater head is defined as

$$h_f = \frac{p}{\rho_f g} + z,$$

where p is the pressure of the groundwater, ρ_f the density of freshwater, g the acceleration due to gravity and z the vertical coordinate of the location of measure. The freshwater head can be explained as the elevation above an arbitrary datum of the water surface in a piezometer tube filled over its full height with freshwater. In Figure 2.1 the difference between the freshwater head and hydraulic head is explained. Fluids flow down a hydraulic gradient, from points of higher to lower hydraulic head. The quantity of head is expressed in terms of a length of water. Formulation of the flow equation in terms of freshwater head causes no increase in complexity and allows the use of existing software with relatively little modification.

2.1.1 Continuity of mass flow

The control volume in Figure 2.2 is defined in order to derive the conservation of mass flow. The mass flow $\dot{m}$ is defined as the amount of mass flowing through the control volume per unit time. For directions x, y and z the mass flow is respectively $\dot{m}_x$, $\dot{m}_y$ and $\dot{m}_z$ and hence the total change of mass flow in the control volume is

$$\dot{m}_{out} - \dot{m}_{in} = \dot{m}_{x_{out}} + \dot{m}_{y_{out}} + \dot{m}_{z_{out}} - \dot{m}_{x_{in}} - \dot{m}_{y_{in}} - \dot{m}_{z_{in}},$$

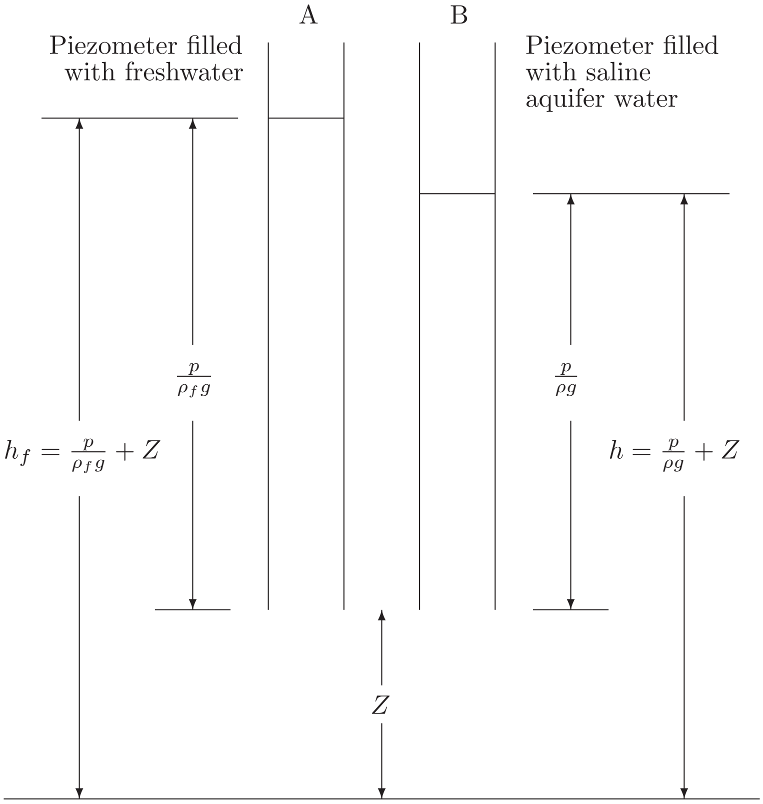

A
B
Piezometer filled with freshwater
Piezometer filled with saline aquifer water
$\frac{p}{\rho_f g}$
$h_f = \frac{p}{\rho_f g} + Z$
$\frac{p}{\rho g}$
$h = \frac{p}{\rho g} + Z$
Z

Figure 2.1: Two piezometers, one filled with freshwater and the other with saline water, open to the same point in the aquifer. With h_f the freshwater head, h the hydraulic head, ρ_f the freshwater density, ρ the density of the saline aquifer water and Z the elevation.

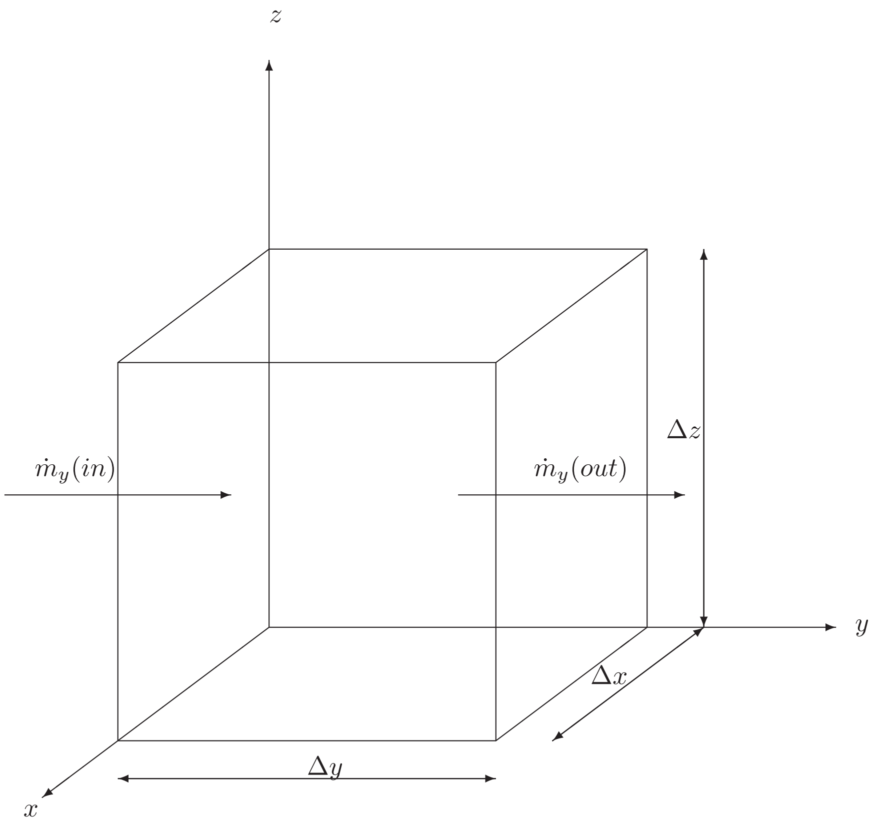


Figure 2.2: Control volume for the mass flow in y-direction.

Continuity of the mass flow for one of the directions then can be defined by the outgoing minus incoming flux q_i in the ith direction multiplied by the density ρ of the water and the surface of the control volume in the ith direction. Consider the mass flow in the x-direction, the continuity equation is

$$\dot{m}_{x_{out}} - \dot{m}_{x_{in}} = (\rho_{out} q_{x_{out}} - \rho_{in} q_{x_{in}})\,\Delta y \Delta z =: \Delta(\rho q_x) \Delta y \Delta z,$$

or

$$\Delta \dot{m}_x = \frac{\Delta(\rho q_x)}{\Delta x} \Delta x \Delta y \Delta z.$$

The mass flow in the y-direction and z-direction can be derived in an equivalent way. Now the total change of mass flow through the control volume can be written as:

$$\Delta \dot{m} = \left(\frac{\Delta(\rho q_x)}{\Delta x} + \frac{\Delta(\rho q_y)}{\Delta y} + \frac{\Delta(\rho q_z)}{\Delta z} \right) \Delta x \Delta y \Delta z. \tag{2.1}$$

The specific storage S_s is the change in storage and is defined as the amount of water which a given volume of aquifer will produce, provided a unit change in hydraulic head is applied to it. It has units of inverse length. Flow in a porous medium is considered, hence the volume of a control volume of an aquifer ($\Delta x \Delta y \Delta z$) is not necessarily the same as the volume of water (V_w) in the same control volume. There is porosity to relate the aquifer volume to the water volume. The specific storage is by definition expressed in terms of V_w, h, x, y and z:

$$S_s = -\frac{\Delta V_w}{\Delta h \Delta x \Delta y \Delta z}. \tag{2.2}$$

The total change in mass flow can be defined by

$$\Delta \dot{m} = \frac{\Delta V_w \Delta \rho}{\tau}. \tag{2.3}$$

Substitution of the Equations (2.1) and (2.2) in Equation (2.3) gives

$$\Delta \dot{m} = -S_s \frac{\Delta\left(\rho h\right)}{\tau} \Delta x \Delta y \Delta z. \tag{2.4}$$

Collecting both expressions for the change in mass flow (equations (2.1) and (2.4)) and dividing by $\Delta x \Delta y \Delta z$ results into the continuity equation for the mass flow in a control volume:

$$\left(\frac{\Delta(\rho q_x)}{\Delta x} + \frac{\Delta(\rho q_y)}{\Delta y} + \frac{\Delta(\rho q_z)}{\Delta z} \right) = -S_s \frac{\Delta\left(\rho h\right)}{\tau}. \tag{2.5}$$

If a source or sink is present Equation (2.5) becomes:

$$\frac{\Delta(\rho q_x)}{\Delta x} + \frac{\Delta(\rho q_y)}{\Delta y} + \frac{\Delta(\rho q_z)}{\Delta z} + \rho q' = -S_s \frac{\Delta\left(\rho h\right)}{\tau}, \tag{2.6}$$

with q' the volumetric flow rate per unit volume of aquifer representing sources and sinks. It has units of inverse time. Taking limits results into:

$$\lim_{\tau \to 0, \Delta x \to 0, \Delta y \to 0, \Delta z \to 0} \left(\frac{\Delta(\rho q_x)}{\Delta x} + \frac{\Delta(\rho q_y)}{\Delta y} + \frac{\Delta(\rho q_z)}{\Delta z} + \rho q' \right) = \frac{\partial(\rho q_x)}{\partial x} + \frac{\partial(\rho q_y)}{\partial y} + \frac{\partial(\rho q_z)}{\partial z} + \rho q',$$

$$\lim_{\tau\to 0,\Delta x\to 0,\Delta y\to 0,\Delta z\to 0}\left(-S_s\frac{\Delta(\rho h)}{\tau}\right)=-S_s\frac{\partial(\rho h)}{\partial t}.$$

The differential equation for the continuity of mass flow then becomes:

$$\frac{\partial(\rho q_x)}{\partial x}+\frac{\partial(\rho q_y)}{\partial y}+\frac{\partial(\rho q_z)}{\partial z}+\rho q'=-S_s\frac{\partial(\rho h)}{\partial t}. \qquad (2.7)$$

In order to rewrite Equation (2.7) in terms of freshwater head, define the freshwater head

$$h_f=\frac{p}{\rho_f g}+z,$$

and the (hydraulic)water head

$$h=\frac{p}{\rho g}+z,$$

and eliminate the pressure in the above equations to obtain the relation

$$h=\frac{\rho_f}{\rho}h_f+\frac{\rho-\rho_f}{\rho}z.$$

The right-hand side of Equation (2.7) can now be written as:

$$-S_s\frac{\partial(\rho h)}{\partial t}=-S_s\left(\rho_f\frac{\partial h_f}{\partial t}+z\frac{\partial\rho}{\partial t}\right).$$

Note that the density is written as a function of the concentration (C) of a solute (for example salt) because the equation for Solute Transport in Chapter 2.2 is expressed in terms of concentration. The relation between those two parameters is also explained in Chapter 2.2. Note that ρ is differentiable to C and C is differentiable to the time t. Under isothermal conditions and use of the Chain Rule for differentiating on $\rho=\rho(C)$, the groundwater flow equation expressed in terms of the freshwater head is:

$$-\frac{\partial(\rho q_x)}{\partial x}-\frac{\partial(\rho q_y)}{\partial y}-\frac{\partial(\rho q_z)}{\partial z}+\rho q'=S_s\left(\rho_f\frac{\partial h_f}{\partial t}+z\frac{\partial\rho}{\partial C}\frac{\partial C}{\partial t}\right). \qquad (2.8)$$

The left-hand side of Equation (2.8) is the net flux of mass through the faces of the control volume plus the rate at which mass enters from sources or leaves through sinks located in the control volume. The right-hand side is the time rate of change in the mass stored in the control volume over a given period. The recharge term q' has dimension $[1/s]$ and is the sum of four distinctive components, depending on the origin of the water:

$$q'=q_a+q_l+q_r+q_s,$$

with

q_l	recharge due to leakage,
q_r	recharge from rivers canals and drains,
q_s	recharge from sources or sinks,
q_a	recharge from the top-system (precipitations, shallow drainage system etc.),

[8, 13].

2.1.2 Darcy's law

Darcy's law describes the flow of a fluid through a porous medium. For variable density it is given by:

$$\begin{pmatrix} q_x \\ q_y \\ q_z \end{pmatrix} = -\frac{1}{\mu} \begin{pmatrix} \kappa_{xx} & \kappa_{xy} & \kappa_{xz} \\ \kappa_{yx} & \kappa_{yy} & \kappa_{yz} \\ \kappa_{zx} & \kappa_{zy} & \kappa_{zz} \end{pmatrix} \begin{pmatrix} \frac{\partial p}{\partial x} \\ \frac{\partial p}{\partial y} \\ \frac{\partial p}{\partial z} + \rho g \end{pmatrix}, \tag{2.9}$$

with κ, the intrinsic permeability. From the definition of the freshwater head, the pressure is given by:

$$p = \rho_f g(h_f - z), \tag{2.10}$$

with z upward positive.

For the same reasons as for the continuity of mass flow, Darcy's law is rewritten in terms of freshwater head and freshwater hydraulic conductivity. Define the freshwater hydraulic conductivity as

$$k_{f_{ij}} = \frac{\kappa_{ij}\rho_f g}{\mu_f},$$

with μ_f the freshwater dynamic viscosity, ρ_f the freshwater density and g the acceleration due to gravity. The derivatives of the pressure can be calculated by

$$\frac{\partial p}{\partial x} = \rho_f g \frac{\partial h_f}{\partial x},$$

$$\frac{\partial p}{\partial y} = \rho_f g \frac{\partial h_f}{\partial y},$$

$$\frac{\partial p}{\partial z} = \rho_f g \left(\frac{\partial h_f}{\partial z} - 1 \right),$$

hence Equation (2.9) becomes

$$\begin{pmatrix} q_x \\ q_y \\ q_z \end{pmatrix} = - \begin{pmatrix} k_{f_{xx}} & k_{f_{xy}} & k_{f_{xz}} \\ k_{f_{yx}} & k_{f_{yy}} & k_{f_{yz}} \\ k_{f_{zx}} & k_{f_{zy}} & k_{f_{zz}} \end{pmatrix} \begin{pmatrix} \frac{\partial h_f}{\partial x} \\ \frac{\partial h_f}{\partial y} \\ \frac{\partial h_f}{\partial z} + \frac{\rho - \rho_f}{\rho_f} \end{pmatrix}. \tag{2.11}$$

This is Darcy's law for variable density expressed in freshwater head [7, 14].

2.1.3 Groundwater flow equation

Substitution of Darcy's law (Equation (2.11)) in the equation for conservation of mass (Equation (2.8)) results in the general Groundwater flow equation in terms of fresh groundwater head and density:

$$
\begin{aligned}
&\frac{\partial}{\partial x}\left(\rho\left(k_{f_{xx}}\frac{\partial h_f}{\partial x}+k_{f_{xy}}\frac{\partial h_f}{\partial y}+k_{f_{xz}}\left(\frac{\partial h_f}{\partial z}+\frac{\rho-\rho_f}{\rho_f}\right)\right)\right)+\\
&\quad+\frac{\partial}{\partial y}\left(\rho\left(k_{f_{yx}}\frac{\partial h_f}{\partial x}+k_{f_{yy}}\frac{\partial h_f}{\partial y}+k_{f_{yz}}\left(\frac{\partial h_f}{\partial z}+\frac{\rho-\rho_f}{\rho_f}\right)\right)\right)+\\
&\quad+\frac{\partial}{\partial z}\left(\rho\left(k_{f_{zx}}\frac{\partial h_f}{\partial x}+k_{f_{zy}}\frac{\partial h_f}{\partial y}+k_{f_{zz}}\left(\frac{\partial h_f}{\partial z}+\frac{\rho-\rho_f}{\rho_f}\right)\right)\right)+\rho q'=\\
&\quad=S_s\left(\rho_f\frac{\partial h_f}{\partial t}+z\frac{\partial \rho}{\partial C}\frac{\partial C}{\partial t}\right). \qquad (2.12)
\end{aligned}
$$

The boundary and initial conditions for the groundwater head can be found Section 2.4.

Parameters of the flow equation

According to [9], the ranges of values of the specific storage S_s are independent of time but depend on location. For several materials they can be found in Table 2.1.

Table 2.1: Ranges of values of S_s, adapted from Domenico 1972 [9].

Material	Specific storage S_s (m^{-1})
Loose sand	$1.0 * 10^{-3} - 4.9 * 10^{-4}$
Dense sand	$2.0 * 10^{-4} - 1.3 * 10^{-4}$
Dense sandy gravel	$1.0 * 10^{-4} - 4.9 * 10^{-5}$
Rock, fissured, jointed	$6.9 * 10^{-5} - 3.3 * 10^{-6}$

The temperature of the groundwater is often the mean temperature of the air during the year and hence constant. The freshwater density ρ_f is constant under isothermal conditions. In Table 2.2 can be seen that the chosen groundwater temperature does not highly influence the freshwater density.

Table 2.2: Ranges of values of ρ_f.

Temperature (℃)	Freshwater density ρ_f $(kg/liter)$
4	1.000
20	0.9982
40	0.9922
80	0.9718

The values of the freshwater hydraulic conductivity tensor k_f are all known and assumed to be continuous and differentiable. The control volume $\Delta x \Delta y \Delta z$ is aligned with coordinate directions that are neither parallel nor normal to the aquifer. Often the aquifers are horizontal and in that case the non-diagonal elements are zero. But in order to be able to use this model in all cases, for example in the case of a lateral moraine where the groundlayers are not in the same direction as the water flow, the complete tensor is used.

The freshwater head h_f is the output variable which is determined by this differential equation.

2.2 Solute transport

In general situations, the direction of flow is variable. Again consider the control volume as defined in Figure 2.2 but now consider the change of mass in time ($\dot{m}$) expressed in the Darcy velocity $\mathbf{q}$ and the concentration C of a solute (salt) in the water. The control volume $\Delta x \Delta y \Delta z$ is aligned with coordinate directions that are neither parallel nor normal to the aquifer. Thus the Darcy velocity has components in all three dimensions. Presence of sources or sinks within the control volume is possible.

First the change of mass in time due to advection and sources or sinks is developed. For simplicity, it is assumed that storage effects involve only changes in fluid density within a rigid porous framework. The net inflow of solute mass in the x-direction for the control volume is:

$$\dot{m}_x = -\frac{\Delta(q_x C)}{\Delta x} \Delta x \Delta y \Delta z.$$

The y-direction and z-direction are derived analogously. And the total change of mass in time due to advection and sources or sinks is:

$$\dot{m} = -\left(\frac{\Delta(q_x C)}{\Delta x} + \frac{\Delta(q_y C)}{\Delta y} + \frac{\Delta(q_z C)}{\Delta z} \right) \Delta x \Delta y \Delta z + Q_{so} C_s. \tag{2.13}$$

C_s represents the concentration of the solute in the water that is added or withdrawn and Q_{so} denotes the volumetric rate at which water is added or removed, where a positive sign indicates a source and a negative sign a sink. The term $Q_{so}C_s$ thus represents the net rate at which solute mass is added to or removed from the control volume by the source or sink, expressed in units of mass per unit time.

It is assumed that the solute carried in advective transport remains completely within the moving water. In particular there is no diffusion of solute into and from sections of the pore space that may contain (nearly) static water. Static water is the term used for non-moving water that does not contribute to the continuity of mass. Then the volume of water containing solute in the control volume $\Delta x \Delta y \Delta z$ is $\theta \Delta x \Delta y \Delta z$. With θ the dimensionless effective porosity independent on time but dependent on spatial coordinates. The mass of solute in the control volume at any time is $\theta \Delta x \Delta y \Delta z C$ with C the average concentration in the control volume. Thus the rate at which the mass changes with time can also be written as:

$$\dot{m} = \theta \frac{\Delta C}{\tau} \Delta x \Delta y \Delta z. \tag{2.14}$$

Combination of the equations (2.13) and (2.14), dividing both sides by $\Delta x \Delta y \Delta z$ and taking the limits of τ, Δx, Δy and Δz to 0 results in:

$$-\nabla \cdot (\mathbf{q}C) + q_{so}C_s = \theta \frac{\partial C}{\partial t}|_{\text{due to advection and sources/sinks}}, \tag{2.15}$$

with q_{so} the volumetric flow rate per unit volume of the aquifer due to the fluid source or sink.

Second, the change of mass in time due to dispersion is developed. For the three-dimensional case, the dispersion coefficient tensor contains nine terms. The dispersive transport in terms of mass per unit time in the control volume is derived in [9] and given by:

$$\dot{m}_i = -\left(D_{ix}\frac{\Delta C}{\Delta x} + D_{iy}\frac{\Delta C}{\Delta y} + D_{iz}\frac{\Delta C}{\Delta z} \right) \theta \Delta x_j \Delta x_k .$$

The difference between inflow and outflow of mass due to dispersion can be derived by multiplying above equation by $\frac{\Delta x_i}{\Delta x_i}$, again using $\dot{m} = \dot{m}_1 + \dot{m}_2 + \dot{m}_3$ and taking the limits for τ, Δx, Δy and Δz to 0. This results in:

$$\nabla \cdot (\theta D \nabla C) = \theta \frac{\partial C}{\partial t}|_{\text{due to dispersion}}, \tag{2.16}$$

Combination of the equations (2.15) and (2.16) results in the transport equation of solute mass in groundwater:

$$\theta \frac{\partial C}{\partial t} = \nabla \cdot (\theta D \nabla C) - \nabla \cdot (\mathbf{q} C) + q_{so} C_s . \tag{2.17}$$

When the assumption of divergence free (solenoidal) groundwater is used, Equation (2.17) becomes

$$\boxed{\theta \frac{\partial C}{\partial t} = \nabla \cdot (\theta D \nabla C) - \nabla \cdot (\mathbf{q} C) + q_{so} C_s , \qquad (2.18)}$$

with

$$D \nabla C = \begin{pmatrix} D_{xx} & D_{xy} & D_{xz} \\ D_{yx} & D_{yy} & D_{yz} \\ D_{zx} & D_{zy} & D_{zz} \end{pmatrix} \begin{pmatrix} \frac{\partial C}{\partial x} \\ \frac{\partial C}{\partial y} \\ \frac{\partial C}{\partial z} \end{pmatrix} .$$

2.2.1 Parameters of the solute transport equation

Solute mass is transported in porous media by the flow of groundwater (advection), molecular diffusion and mechanical dispersion. Convection is the internal movement of mass within fluids (i.e. liquids and gases). It cannot occur in solids due to the atoms not being able to flow freely. Convection may cause a related phenomenon called *advection*, in which mass or heat is transported by the motion of the fluid. In hydrology, advection refers to the horizontal or vertical flow of water in a stream.

Dispersive transport describes the dilution or mixing of a solute due to different velocities of groundwater, which is moving at rates that are both greater and smaller than the average advective pore velocity. Dispersion is observed on both the microscopic and the macroscopic scale. The three main reasons for the different velocities at the microscopic scale are friction in pores, varying travel path lengths and pore sizes. Macroscopic dispersion is caused by variable permeability's of single layers inducing different velocities. On a microscopic scale also a process independently from the water movement becomes important: diffusion. Diffusion describes the movement of a solute from an area of greater concentration to one of less concentration. *Molecular diffusion* is defined as the transport of matter solely by the random motions of individual molecules not moving together in coherent groups. This process is a consequence of concentration gradients and described by Fick's law.

In [14] and [15] the coefficients of the dispersive matrix D are derived and presented in Cartesian coordinates:

$$\begin{aligned}
D_{xx} &= \frac{a_T\left(v_y^2+v_z^2\right)+a_L v_x^2}{v}, \\
D_{xy} &= \frac{\left(a_L-a_T\right)v_x v_y}{v} = D_{yx}, \\
D_{xz} &= \frac{\left(a_L-a_T\right)v_x v_z}{v} = D_{zx}, \\
D_{yy} &= \frac{a_T\left(v_x^2+v_z^2\right)+a_L v_y^2}{q}, \\
D_{yz} &= \frac{\left(a_L-a_T\right)v_y v_z}{v} = D_{zy}, \\
D_{zz} &= \frac{a_T\left(v_x^2+v_y^2\right)+a_L v_z^2}{q},
\end{aligned} \tag{2.19}$$

with $v = \sqrt{v_x^2+v_y^2+v_z^2}$ the magnitude of the seepage velocity. Note that the relation between the Darcy velocity $\mathbf{q}$ and the seepage velocity $\mathbf{v}$ is given by

$$\mathbf{q} = \mathbf{v}/\theta$$

. Seepage velocity is defined as the percolation of water through the soil from unlined canals, ditches, laterals, watercourses, or water storage facilities. It is assumed that $q \neq 0$. The matrix D is a full matrix (anisotropic), which means that dispersive transport in each coordinate direction depends on components of the velocity and concentration gradient in all directions.

The dispersivity is expressed in a_L and a_T. a_L is defined as the longitudinal dispersivity of the porous medium, a property of the porous medium describing dispersive transport in the direction of flow. The constant a_T is the transversal dispersivity of the porous medium, that is, normal to the direction of flow. Both have the dimension of length. When the water completely fills the void space in a porous medium, a_L should be of the order of magnitude of a pore size. Experiments have shown that a_T is 8 to 24 times as small as a_L.

The first term of the right-hand side of Equation (2.18) is a combined term of molecular diffusion and dispersion. The order of magnitude of diffusion is $10^{-9}\ m^2/day$, the order of magnitude of dispersion is $3 * 10^{-3}\ m^2/day$. The second term of the right-hand side is the advection term and is of the order $3 * 10^{-2}\ m/day$. The diffusion becomes only important when the velocity field is zero because the order of the dispersion and advection effect is much bigger than the order of the diffusion effect. When the groundwater does not move the dispersion coefficient or matrix D reduces to the diffusion coefficient or matrix. Note that there is no reaction term in the differential equation because only salt is considered.

The Darcy velocity vector $\mathbf{q}$ is determined by the groundwater flow equation (2.12). The porosity θ is a subsurface property and hence depends on the spatial coordinates. The porosity θ and the source/sink term $q_{so}C_s$ are known for each part of the ground. The concentration C as a function of time and location has to be solved from this differential equation [14, 15].

2.3 Conversion of concentration into density

The model for solute transport will deliver the values for the concentration C, and hence the density ρ, each time step for each location. The advection-dispersion equation calculates the concentration of the salt in the groundwater. The flow equation uses the density ρ in $[kg/l]$ for the calculation of the groundwater velocity, so a translation has to be done from C into ρ. In [16] the following formula is experimentally derived:

$$\rho = 1 + 8.05 * 10^{-7} * RE - 6.5 * 10^{-6} \left(T - 4 + 2.2 * 10^{-4} * RE\right)^2 , \qquad (2.20)$$

where T is the temperature in °C and RE is the residue on evaporation at 180°C in mg/kg which can be calculated from chlorinity ($mg\ Cl^-$/kg water):

$$RE = 1.805Cl^- + 30.$$

The temperature of groundwater remains relatively constant throughout the year. The ground temperature for an area is approximately equal to an area's annual average air temperature, which is around 10°C in the Netherlands. The earth and groundwater temperatures are much more stable than the highly variable seasonal air temperature. Note that in freshwater $C = 0\ kg/m^3$ so $Cl^- = 0\ mg/kg$, which results in $\rho = 0.9998\ kg/l$ or $\rho = 999.8\ kg/m^3$. The concentration C is used instead of Cl^- in mg/kg water, the difference is approximately 2.5%. Note that for C can be said: $mg/kg \approx mg/l = \frac{1}{1000} kg/m^3$ and for ρ: $kg/l = \frac{1}{1000} kg/m^3$.

2.4 Boundary and initial conditions

In order to make the solution of Equations (2.12) and (2.18) unique and well-posed, a number of conditions (boundary and initial) should be prescribed. For a unique solution of the advection-dispersion equation or the diffusion equation (elliptic equations), it is necessary to prescribe exactly one boundary condition at each part of the boundary. Note that the advection term in Equation (2.18) might strongly dominate the dispersive term. For a pure advection equation (hyperbolic equation), boundary conditions should only be given at inflow and not at outflow boundaries. Since for the advection-dispersion equation boundary conditions must be given at the outflow, those boundary conditions are recommended that influence the solution as little as possible. In general this means that at the outflow boundary one usually applies natural boundary conditions. Dirichlet boundary conditions may result in unwanted wiggles.

The following boundary conditions are considered:

Dirichlet boundary

A Dirichlet boundary Γ_0 is such that the value of the head or the concentration is specified at all points along the boundary Γ_0. For example, for the concentration the Dirichlet boundary conditon is:

$$C(\mathbf{x}) = g_0(\mathbf{x}), \qquad \mathbf{x} \in \Gamma_0. \tag{2.21}$$

A physical example of a Dirichlet boundary might be a continuous and fast source of salt.

Neumann boundary
On the Neumann boundary Γ_1 a condition in which the gradient of the dependent variable normal to the boundary is specified. For example, for the concentration the Neumann boundary condition for the boundary Γ_1 is:

$$(D \cdot \nabla C) \cdot \mathbf{n} = g_1(\mathbf{x}), \qquad \mathbf{x} \in \Gamma_1. \tag{2.22}$$

For groundwater flow this boundary condition results into a specified flux of water into or out of the modeled area. For solute transport the concentration gradient is specified normal to the boundary. A physical example is an impermeable boundary where the gradients of head and concentration are zero at the boundary. An example of a nonzero Neumann boundary in flow simulation might be a surface-water body from which seepage occurs at a prescribed rate.

Robbins boundary
On the Robbins boundary Γ_3 a mixed condition is specified:

$$(D \cdot \nabla C) \cdot \mathbf{n} + \sigma C = g_2(\mathbf{x}), \qquad \mathbf{x} \in \Gamma_2. \tag{2.23}$$

Here a flow might be prescribed in which both the dispersive and advective contributions are taken into account.

According to Leijnse [17] four types of boundaries can be distinguished: noflow boundary, inflow boundary, outflow boundary and dissolving salt boundary. A dissolving salt boundary is present where rock salt formations are in contact with flowing groundwater. This boundary will not be used here, but more information can be found in [17].

The boundary condition for the *no-flow boundary* is defined as the dispersive salt mass flux across the boundary being zero, so an homogeneous Neumann condition is used.

For the *inflow boundary* the case is considered where the boundary is within the porous medium and separates one part of the porous medium from another part. The assumption of continuity of salt mass flux across the boundary is

$$(\rho C\mathbf{q} + D\nabla C) \cdot \mathbf{n}|_+ = (\rho C\mathbf{q} + D\nabla C) \cdot \mathbf{n}|_-, \tag{2.24}$$

where $-$ indicates a position just outside the domain and $+$ indicates a position just inside the domain. This assumption is combined with the assumption that the dispersive and diffusive fluxes in the inflow reservoir can be neglected. Or in other words, the influx of mass takes place from a well mixed reservoir. A combination of both assumptions results in

$$(\rho C\mathbf{q} + D\nabla C) \cdot \mathbf{n} = C_0\rho\mathbf{q} \cdot \mathbf{n}, \tag{2.25}$$

with C_0 the prescribed salt mass fraction in the inflowing liquid.

It is usually assumed that the salt mass fraction is continuous across the *outflow boundary.* If the dispersive and diffusive mass fluxes just outside the domain are neglected the boundary condition for the outflow boundary is

$$D\nabla C \cdot \mathbf{n} = 0. \tag{2.26}$$

For the time dependent problem, initial conditions for the transport equation (concentration) must be specified. The flow equation is a steady equation in the coupled model (Section 2.5) [7, 13, 17, 18].

2.5 The coupled model

The movement of groundwater (Equation (2.12)) and the transport of solutes (Equation(2.18)) in the aquifer are coupled processes and the two equations must be solved as a coupled problem. A schematic representation of this coupled process is shown in Figure 2.4 and starts with the steady flow equation. The flow equation is taken time independent because the differences in pressure in the groundwater are small, hence the storage of water is negligible. The velocity of the flow is calculated instantly when the density of the water is given. Each loop of the coupled process between the flow equation and the transport equation is called an cycle. The density for the first cycle is a given density. The flow equation calculates the freshwater head with this initial density of the water at time t_i. The freshwater head is converted into the flow velocity $\mathbf{q}$ with Darcy's law. Then the time dependent solute transport equation is solved with the calculated velocity at t_i, the solution of this equation is the concentration at time t_{i+1}. The initial concentration for the transport equation of the first cycle of the coupled process is determined by a conversion of the initial density ρ of the flow equation into concentration by the inverse formula of Equation (2.20). The concentration at t_{i+1} is converted into the density at $t + 1$ and returned to the flow equation which calculates the new freshwater head at t_{i+1}. The initial condition of the transport equation at t_{i+1} is now the final concentration calculated at t_i by the transport equation. The course of time of this process is shown in Figure 2.3.

In order to increase the accuracy or stability of the coupled system, inner iterations can be used. An inner iteration is an coupling between the steady flow equation and the time dependent transport equation in the same way as shown in Figure 2.4, but in this case without the time passing by. Inner iterations are made until stable solutions appear for both equations. After this stable solution is reached, a new cycle is made for the next time step. If necessary, again inner iterations are done for this new time step. This process increases the computational time significantly.

A better accuracy can also be obtained by decreasing the time step of one cycle, and increasing the number of cycles. In Chapter 4 research is done on the impact of the time step of the Transport equation, the time step of the coupled process and the number of cycles.

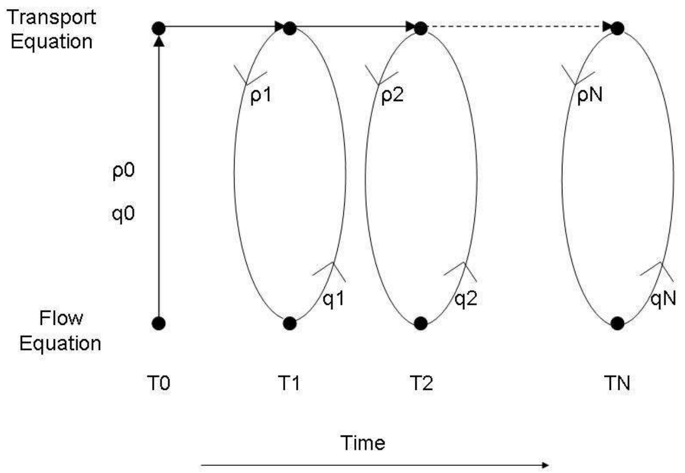


Figure 2.3: The timesteps and interaction within the coupled model.

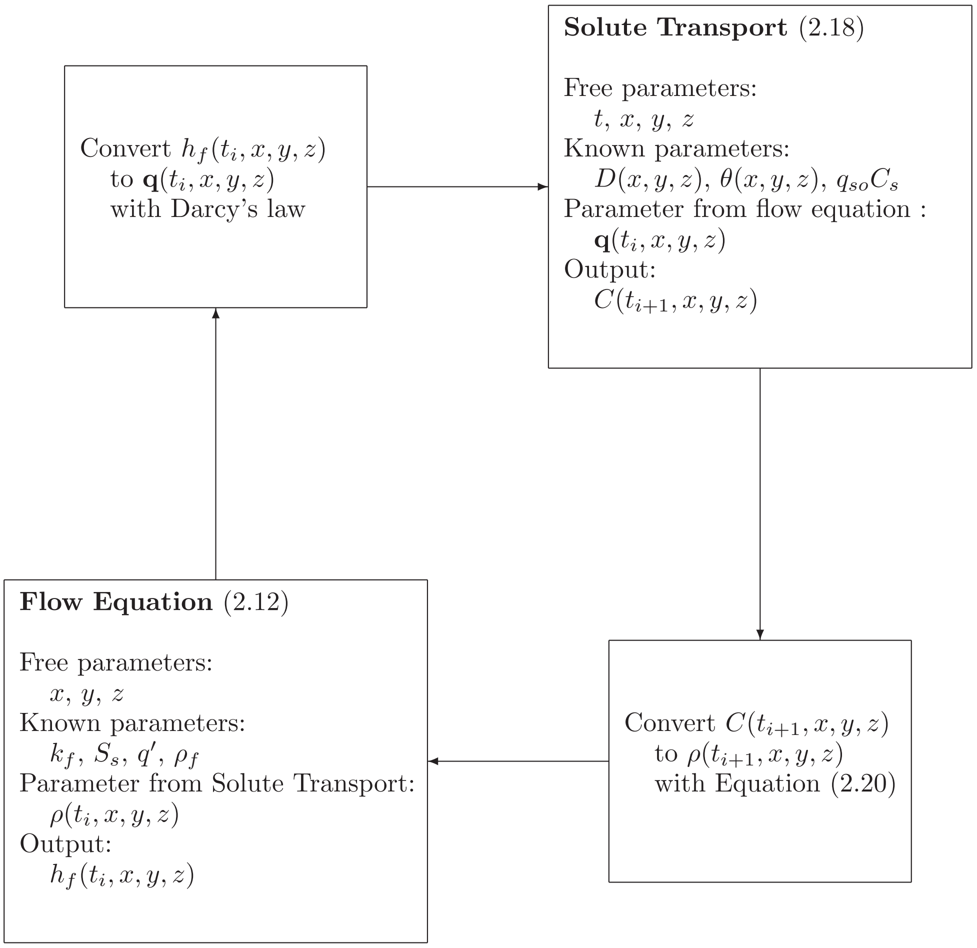


Figure 2.4: The coupled process for the solving of the equations for solute transport (2.18) and groundwater flow (2.12).

Chapter 3

Numerical solution methods

The Solute transport equation is solved with a numerical method. In Section 3.1 the model and grid for the transport equation (Equation (2.18)) is explained. The spatial discretization of the transport equation is done by a combination of different numerical methods. The Finite Element Methods (FEM) that solve this equation in the two dimensional horizontal direction can be found in Section 3.2. For the third dimension, in the vertical direction, the used Finite Difference and Finite Volume Methods can be found in Section 3.3. In Section 3.4 the temporal discretization scheme for the transport equation can be found. In Section 3.5 some notes are made on behalf of the stability and accuracy of the used methods. Section 3.6 gives the method that solves the system of equations formed by the spatial and temporal discretization schemes.

The flow equation is solved by Triwaco, an introduction into Triwaco can be found in Section 1.3. The transport equation and the flow equation are coupled in order to calculate the salt transport with variable velocities or the groundwater flow with variable density. In Section 2.5 information is shown about the coupled model that solves the density dependent groundwater flow or the salt migrations in time.

3.1 Grid transport equation

The transport equation

$$\begin{cases} -\nabla \cdot (\theta D \nabla C) + \mathbf{q} \cdot \nabla C + \theta \frac{\partial C}{\partial t} = q_{so} C_s, & \\ C|_{\Gamma_1} = g_1(\mathbf{x}), & \\ ((\theta D \nabla C) \cdot \mathbf{n})\,|_{\Gamma_2} = g_2(\mathbf{x}), & \\ (\sigma C + (\theta D \nabla C \cdot \mathbf{n}))\,|_{\Gamma_3} = g_3(\mathbf{x}), & \sigma \geq 0, \\ C(\mathbf{x}, t_0) = C_0(\mathbf{x}), & \end{cases} \tag{3.1}$$

with D the symmetric dispersivity matrix

$$D = \begin{pmatrix} D_{xx} & D_{xy} & D_{xz} \\ D_{yx} & D_{yy} & D_{yz} \\ D_{zx} & D_{zy} & D_{zz} \end{pmatrix},$$

is numerically solved in three dimensions. Γ_1 is empty, Γ_2 is defined at the no-flow and outflow boundaries and Γ_3 is defined at the inflow boundary. The quantity σ is taken 0 or 0.5 and $g_2(\mathbf{x})$=$g_3(\mathbf{x})$=0. (In the experiments there is no clear dependence of the concentration on the values of σ, $g_2(\mathbf{x})$ and $g_3(\mathbf{x})$.) Note that the differential equation is linear, because the coefficients are independent of the solution.

From a geological point of view (see Figure 1.4) the subsurface is divided into aquifers and (geological) aquitards. The three dimensional problem of Equation (3.1) holds for the aquifers. But in the aquitards, there is no horizontal flow because the permeability in the horizontal direction is zero. Since the intrinsic permeabilities in the horizontal direction in Darcy's law (Equation (2.9)) are

$$\kappa_{xx} = \kappa_{xy} = \kappa_{xz} = \kappa_{yx} = \kappa_{yy} = \kappa_{yz} = 0,$$

it can be seen from Darcy's law that indeed

$$q_x = q_y = 0.$$

The dispersion coefficients in the aquitards can be calculated with the Equations (2.19). The zero horizontal velocity, $q_x = q_y = 0$, results in

$$D_{xy} = D_{xz} = D_{yx} = D_{yz} = D_{zx} = D_{zy} = 0.$$

Hence the transport equation for the geological aquitards becomes

$$\theta \frac{\partial C}{\partial t} = \nabla \cdot (\theta D \nabla C) - \frac{\partial q_z C}{\partial z} + q_{so} C_s, \qquad (3.2)$$

with

$$D = \begin{bmatrix} D_{xx} & 0 & 0 \\ 0 & D_{yy} & 0 \\ 0 & 0 & D_{zz} \end{bmatrix},$$

with

$$D_{xx} = a_T q_z, \qquad D_{yy} = a_T q_z, \qquad D_{zz} = a_L q_z.$$

A difference has to be made between model aquitards and real (geological) aquitards. A model aquitard is a very small horizontal layer within an aquifer used to solve the flow equation with a finer grid in the vertical direction. In geological aquitards (called aquitards in this thesis) Equation (3.2) holds. Model aquitards are only used to solve the flow equation, they are skipped in the numerical solution of the transport equation.

When no model aquitards are used, the cell size of the grid in the vertical direction equals half the height of an aquifer plus half the height of the neighbour aquitard, as can be seen in Figure 3.1. Often aquitards are much smaller than aquifers, the magnitude of aquitards in the Netherlands is $5 - 10$ meters and the magnitude of aquifers is $20 - 50$ meters. This increases the numerical error in the vertical direction because of the different cell sizes. It may be better to use model aquitards in this case in order to split the aquifer into aquifers with the same height as the aquitards and infinitely small model aquitards. The grid of the model aquitards is not used in the numerical method for the transport equation. An example can be found in Figure 3.2.

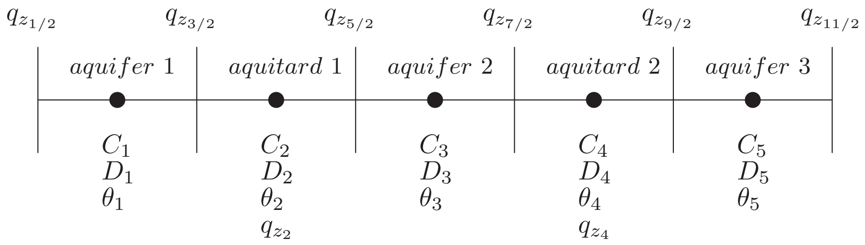


Figure 3.1: The one dimensional grid in the vertical direction with the known parameters.

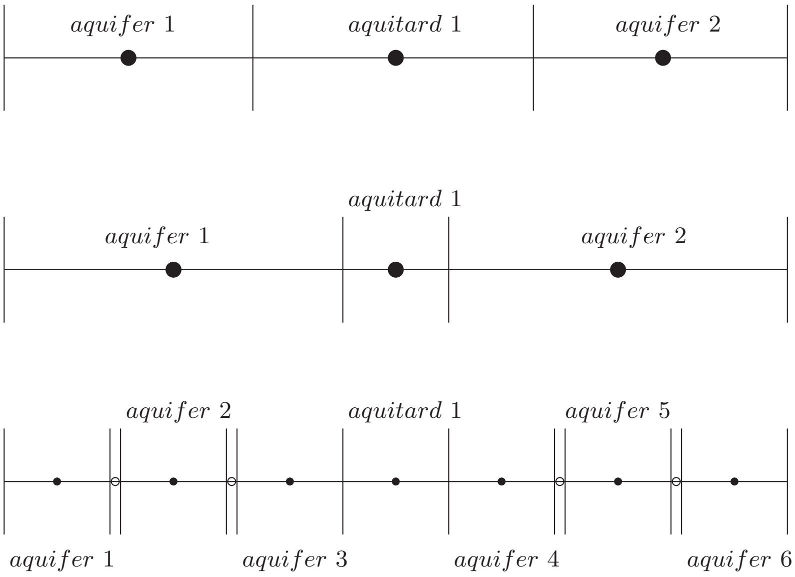


Figure 3.2: The one dimensional grid in the vertical direction. a) All grid cells have the same size. b) The aquitard is smaller than the aquifer, but the grid cells are just a bit smaller than in situation a. c) Modified grid such that the aquifers have the same size as the aquitards. The small layers are model aquitards, which are only used in the flow equation, not in the transport equation.

In order to simplify the calculations and to adapt the numerical method to the used method for the flow equation, the numerical method is split. In the middle of the aquifers the solute transport is calculated into the horizontal directions x and y with the Finite Element Method (FEM). For this method triangular shaped elements are taken. The two dimensional version of Equation (3.1) has to be solved with

$$D = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{yx} & D_{yy} \end{bmatrix}.$$

The numerical method to solve this equation can be found in Section 3.2.

In the middle of the aquitards also the Finite Element Method with triangular shaped elements is used to solve the two dimensional transport equation. Because there is no flow in the horizontal direction in the aquitards the equation

$$\theta \frac{\partial C}{\partial t} = \nabla \cdot (\theta D \nabla C) + q_{so} C_{s_{xy}}, \tag{3.3}$$

with

$$D = \begin{bmatrix} D_{xx} & 0 \\ 0 & D_{yy} \end{bmatrix},$$

is solved with the FEM. The shapes of the triangles are determined by the method that solves the flow equation. These triangles can have angles of more than 90 degrees.

The different layers of the Finite Element grids are in the vertical direction coupled by the Finite Difference Method (FDM) or the Finite Volume Method (FVM). So in the vertical direction an one dimensional grid is used. These methods can be found in Section 3.3. All grid points in the vertical direction are exactly above or beneath each other. The grid and known parameters are shown in Figure 3.3.

3.2 Spatial discretization transport equation: 2D FEM

In the Interim Master's thesis [19] the numerical spatial discretization methods Finite Differences (FDM), Finite Volumes (FVM) and Finite Elements (FEM) and the semi-analytical IFALT method are compared for the one dimensional transport equation. Numerical experiments of these numerical methods can be found in [19]. The advantage of the FDM is that it is easy to implement, a disadvantage is the numerical dispersion which smears out the solution. The advantage of the FVM is the existence of higher order methods which are relatively easy to implement for 1D problems. For higher dimensions this advantage may disappear. The semi-analytical IFALT method has no Courant or Peclet conditions, no time-stepping, is computationally efficient and has low numerical dispersion but is probably less accurate when applied to advective dominated flow.

For advective dominated flow problems in complex domains with sharp fresh-salt fronts in the initial condition the FEM gave the best results. In addition to that the transport equation will be coupled to the flow equation in Triwaco which already uses a grid with triangular cells. Aquifers often have complex shapes and Triwaco already uses the FEM for the flow equation in the x- and

y-direction, hence the conclusion of the Interim Master's thesis [19] was that the FEM is the best method to use for the transport equation.

First, in Section 3.2.1 the Standard Galerkin Approach (SGA) is derived for the two dimensional grid with triangular shaped elements and linear basis functions. In Section 3.2.2 the Streamline Upwind Petrov Galerkin (SUPG) method is discussed and in Section 3.2.3 more information can be found about the Mizukami Hughes algorithm.

3.2.1 SGA

Basis functions

In the ideal situation basis functions are chosen according to the eigenfunctions of the continuous eigenvalue problem corresponding to the advection-dispersion equation. Then an inner product of the eigenfunctions can be defined such that all non-diagonal elements of the coefficient matrix will be zero. So the coefficient matrix will be diagonal, and therefore the solution of the system of equations is trivial. Unfortunately in practice it is almost impossible to find an analytical expression for the eigenfunctions. Numerical computation of the eigenfunctions is in general a harder task than solving the matrix-vector system obtained by the numerical method. Choosing an arbitrary set of basis functions leads to a full coefficient matrix, which costs a lot of computational time.

Basis functions have to be found that satisfy some requirements. First, the basis functions must be linearly independent. Secondly, the basis functions must

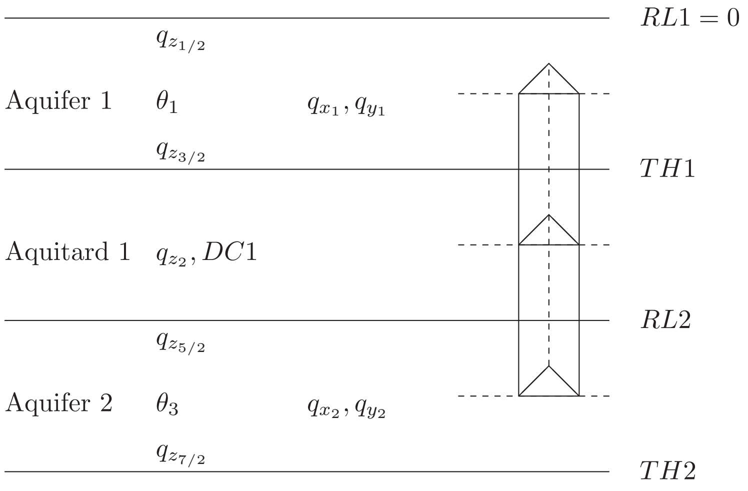


Figure 3.3: The 3D grid: the vertices of the numerical grid lie on the dotted lines within the aquifers and aquitards. q_{i_k} is a known velocity in vertex k in the direction of i ($i \in \{x, y, z\}$), θ_k the porosity in vertex k and RLj and THj are the heights of the top and bottom of aquifer j.

approximate the complete space Σ, which is defined as

$$\Sigma = \{C : C \text{ sufficiently smooth and } C|_{\Gamma_1} = 0\},$$

with Γ_1 the boundary where $C|_{\Gamma_1} = g_1(\mathbf{x})$. The third requirement is that the basis functions should be ”nearly orthogonal”, so most of the integrals $\int_\Omega \nabla\phi_i \cdot \nabla\phi_j d\mathbf{x}$ should be 0. The last requirement is that arbitrary functions in Σ must be approximated by a limited number of basis functions ($\sum_n \to \sum$ as $n \to \infty$).

Note that a set of vectors which is linearly independent and spans some vector space, forms a basis for that vector space. This is equivalent to saying that this basis is a minimal generating set of the vector space or to saying that this basis is a maximal set of linearly independent vectors.

Basis functions are constructed that satisfy the described requirements. In order to construct these basis functions, first a linear polynomial is constructed. In order to construct this linear polynomial, on each triangle three parameters are needed. A natural choice is to use the function values of the concentration C in the three vertices of the triangle. This has added the benefit of making the approximation continuous across element boundaries. Call the three vertices of a triangle $\mathbf{x}^1 = (x_1, y_1)$, $\mathbf{x}^2 = (x_2, y_2)$ and $\mathbf{x}^3 = (x_3, y_3)$.

Each point $\mathbf{x}$ on the triangle can be written as

$$\begin{aligned} \mathbf{x} &= \mathbf{x}^1 + l_2\left(\mathbf{x}^2 - \mathbf{x}^1\right) + l_3\left(\mathbf{x}^3 - \mathbf{x}^1\right) && (3.4)\\ &= l_1\mathbf{x}^1 + l_2\mathbf{x}^2 + l_3\mathbf{x}^3, && (3.5)\end{aligned}$$

with

$$l_1 = 1 - (l_2 + l_3),$$

or

$$l_1 + l_2 + l_3 = 1.$$

To constrain $\mathbf{x}$ inside the triangular shaped element assume $0 \leq l_1, l_2, l_3 \leq 1$.

In order to determine l_1, l_2 and l_3 the system of 3 linear equations given by

$$\begin{aligned} x &= l_1x_1 + l_2x_2 + l_3x_3, && (3.6)\\ y &= l_1y_1 + l_2y_2 + l_3y_3, && (3.7)\\ l_1 + l_2 + l_3 &= 1 && (3.8)\end{aligned}$$

has to be solved. This results in

$$\begin{aligned} l_1(\mathbf{x}) &= 1 - (l_2 + l_3), && (3.9)\\ l_2(\mathbf{x}) &= \frac{y_1 - y_3}{\Delta}x + \frac{x_3 - x_1}{\Delta}y + \frac{x_1y_3 - x_3y_1}{\Delta} && (3.10)\\ l_3(\mathbf{x}) &= \frac{y_2 - y_1}{\Delta}x + \frac{x_1 - x_2}{\Delta}y + \frac{x_2y_1 - x_1y_2}{\Delta} && (3.11)\end{aligned}$$

with

$$\Delta = -x_2y_3 + x_2y_1 + x_1y_3 + x_3y_2 - x_3y_1 - x_1y_2.$$

Note that Δ equals two times the area of the triangle.

It is clear that l_1, l_2 and l_3 are linear functions per element and are defined by the relations

$$l_i(\mathbf{x}^j) = \delta_{ij}. \qquad (3.12)$$

The linear interpolation polynomial for the k^{th} element in 2D can now be constructed by

$$C_k(\mathbf{x}) = l_1 C(\mathbf{x}^1) + l_2 C(\mathbf{x}^2) + l_3 C(\mathbf{x}^3), \tag{3.13}$$

From the Equations (3.12) and (3.13) it is clear that C^n is a linear function of $C_0, C_1, ..., C_n$ so

$$C^n(\mathbf{x}, t) = \sum_{j=0}^{n} C_j(t)\phi_j(\mathbf{x}). \tag{3.14}$$

In order to determine the basis functions, take $C_k = 0$ for $i \neq j$ and $C_i = 1$ in Equation (3.14). This results in

$$\phi_i(\mathbf{x}^j) = l_i(\mathbf{x}^j) = \delta_{ij}. \tag{3.15}$$

So the basis function ϕ_i is only non-zero in the elements that contain the node x_i. The conclusion is that the basis functions ϕ_i are defined by:

Definition 1 *The linearly independent basis functions ϕ_j are defined as:*

1. *$\phi_i(\mathbf{x})$ linear per triangle,*
2. *$\phi_i(\mathbf{x^j}) = \delta_{ij}$.*

Under the conditions of Definition 1, $\{\phi_i\}$ are linearly independent, $\{\phi_i\}$ span a complete function space, and hence $lim_{n\to\infty} C^n(\mathbf{x}, t) = C(\mathbf{x}, t)\ \forall \mathbf{x} \in \mathbb{R}^2$, provided C is continuous. The number n is the number of nodes, which equals $nv \times nl$ with nv the number of nodes for each aquifer and nl the number of aquifers in the model.

The gradients of the basis functions can be determined with the aid of Equations (3.9), (3.10) and (3.11).

$$\nabla\phi_1(\mathbf{x}) = \begin{bmatrix} \frac{\partial\phi_1(\mathbf{x})}{\partial x} \\ \frac{\partial\phi_1(\mathbf{x})}{\partial y} \end{bmatrix} = \begin{bmatrix} \frac{y_3 - y_2}{\Delta} \\ \frac{x_2 - x_3}{\Delta} \end{bmatrix}, \tag{3.16}$$

$$\nabla\phi_2(\mathbf{x}) = \begin{bmatrix} \frac{\partial\phi_2(\mathbf{x})}{\partial x} \\ \frac{\partial\phi_2(\mathbf{x})}{\partial y} \end{bmatrix} = \begin{bmatrix} \frac{y_1 - y_3}{\Delta} \\ \frac{x_3 - x_1}{\Delta} \end{bmatrix}, \tag{3.17}$$

$$\nabla\phi_3(\mathbf{x}) = \begin{bmatrix} \frac{\partial\phi_3(\mathbf{x})}{\partial x} \\ \frac{\partial\phi_3(\mathbf{x})}{\partial y} \end{bmatrix} = \begin{bmatrix} \frac{y_2 - y_1}{\Delta} \\ \frac{x_1 - x_2}{\Delta} \end{bmatrix}. \tag{3.18}$$

Element matrices

The spatial discretization is based on the Finite Element Method. Consider a bounded domain Ω in $\mathbb{R}^2$ and subdivide it into triangles. Define the approximation C^n of the unknown solution C by a finite linear combination of basis functions:

$$C^n(\mathbf{x}, t) = \sum_{j=1}^{n} C_j(t)\phi_j(\mathbf{x}), \tag{3.19}$$

with n the number of nodes, which equals $n_v \times n_l$ with n_v the number of nodes for each aquifer and n_l the number of aquifers in the model.

Galerkin's method for the spatial discretization is derived and results into a system of n linear ordinary differential equations which can be represented by:

$$M\frac{dC}{dt} = SC + f,$$

with the mass matrix M and the stiffness matrix S $n \times n$-matrices with n the number of nodes and $\frac{dC}{dt}$, C and f a $n \times 1$-vectors.

In order to derive the system of ordinary differential equations, start with multiplying the differential equation in (3.1) by a time-independent test function η. This function η satisfies the homogeneous essential boundary condition $\eta|_{\Gamma_1} = 0$ and has to be an element of the Sobolev space H_1. The $\mathcal{L}^2$ space is defined as

$$\mathcal{L}^2(\Omega) = \{f : \Omega \to \mathbb{R} : \int_\Omega f^2 d\mu < \infty\}.$$

This means

$$\int_\Omega \|\nabla\phi\|^2 d\Omega < \infty,$$
$$\phi \text{ measurable},$$
$$\nabla\phi \in \mathcal{L}^2(\Omega) \Rightarrow \phi \in H^1(\Omega).$$

More information about this space can be found in [20].

Then, integrate over the domain Ω:

$$\int_\Omega \left\{-\nabla \cdot (\theta D \nabla C) + \mathbf{q} \cdot \nabla C + \theta\frac{\partial C}{\partial t} - q_{so}C_s\right\} \eta d\Omega = 0. \tag{3.20}$$

Apply Green's theorem only to the second derivative. Application to the first order term would not result in lower order derivatives, since the first derivative of the concentration would be replaced by a first derivative of the test function.

Green's theorem:
Let Ω be the bounded domain with piecewise smooth boundary Γ. Let c, $\mathbf{u}$ be sufficiently smooth, and n the outward normal. Then

$$\int_\Omega c\nabla \cdot \mathbf{u} d\Omega = -\int_\Omega (\nabla c) \cdot \mathbf{u} d\Omega + \int_\Gamma c\mathbf{u} \cdot \mathbf{n} d\Gamma. \tag{3.21}$$

With Green's theorem the second order derivative in Equation (3.20) becomes:

$$-\int_\Omega \{\nabla \cdot (\theta D\nabla C)\} \eta d\Omega = \int_\Omega (\nabla\eta) \cdot (\theta D \nabla C) d\Omega - \int_\Gamma \eta\theta D\nabla C \cdot \mathbf{n} d\Gamma.$$

So Equation (3.20) becomes:

$$\int_\Omega \left((\nabla\eta) \cdot (\theta D\nabla C) + \left(\mathbf{q} \cdot \nabla C + \theta\frac{\partial C}{\partial t} - q_{so}C_s\right)\eta\right) d\Omega - \int_\Gamma \eta\theta D\nabla C \cdot \mathbf{n}\, d\Gamma = 0. \tag{3.22}$$

Substituting the boundary conditions on Γ_2 and Γ_3 (see (3.1)) as well as the essential boundary condition for the test function $\eta|_{\Gamma_1} = 0$ leads to:

$$\int_\Omega \left((\nabla\eta) \cdot (\theta D\nabla C) + (\mathbf{q} \cdot \nabla C)\eta + \theta\frac{\partial C}{\partial t}\eta\right) d\Omega + \int_{\Gamma_3} \sigma C\eta d\Gamma =$$
$$= \int_\Omega q_{so}C_s\eta d\Omega + \int_{\Gamma_2} g_2\eta d\Gamma + \int_{\Gamma_3} g_3\eta d\Gamma. \tag{3.23}$$

Equation (3.23) together the initial condition ($C(\mathbf{x}, t_0) = C_0(\mathbf{x})$), forms the *weak formulation* of (3.1). Approximate C by C^n and substitute (3.19) into (3.23) and substitute $\eta = \phi_i(\mathbf{x})$ for i from 1 to n. This yields the following system of ordinary differential equations (*the Galerkin formulation*):

$$\begin{aligned}
&\frac{d}{dt}\sum_{j=1}^{n} C_j \int_{\Omega} \theta(\mathbf{x})\phi_j\phi_i d\Omega = \\
&\quad = -\sum_{j=1}^{n} C_j \left(\int_{\Omega} \left(\nabla\phi_i \cdot (\theta(\mathbf{x})D(\mathbf{x})\nabla\phi_j) + (\mathbf{q}(\mathbf{x}) \cdot \nabla\phi_j)\,\phi_i\right) d\Omega + \right. \\
&\quad \left. \int_{\Gamma_3} \sigma(\mathbf{x})\phi_j\phi_i d\Gamma \right) + \int_{\Omega} (q_{so}C_s)(\mathbf{x})\phi_i d\Omega + \int_{\Gamma_2} g_2(\mathbf{x})\phi_i d\Gamma + \int_{\Gamma_3} g_3(\mathbf{x})\phi_i d\Gamma, \\
&\quad \text{for } i \in 1, ..., n. \qquad (3.24)
\end{aligned}$$

This system of n linear ordinary differential Equations with n unknowns can be written in the form

$$M\frac{dC}{dt} = SC + f,$$

with M and S $n \times n$-matrices and $\frac{dC}{dt}$, C and f $n \times 1$-vectors. The elements of M, S and f are:

$$M(i,j) = \int_{\Omega} \theta(\mathbf{x})\phi_j\phi_i d\Omega, \qquad (3.25)$$

$$S(i,j) = -\int_{\Omega} \left(\nabla\phi_i \cdot (\theta(\mathbf{x})D(\mathbf{x})\nabla\phi_j) + (\mathbf{q}(\mathbf{x}) \cdot \nabla\phi_j)\phi_i\right) d\Omega - \int_{\Gamma_3} \sigma\phi_j\phi_i d\Gamma, \qquad (3.26)$$

$$f(i) = \int_{\Omega} (q_{so}C_s)(\mathbf{x})\phi_i d\Omega + \int_{\Gamma_2} g_2(\mathbf{x})\phi_i d\Gamma + \int_{\Gamma_3} g_3(\mathbf{x})\phi_i d\Gamma. \qquad (3.27)$$

The above integrals over the domain are split into integrals over the elements in order to make the computations less complicated. With n_e the number of elements, e_k a typical element, n_{be} the number of boundary elements and Ω^{e_k} the area of element e_k the Equations (3.25), (3.26) and (3.27) become:

$$M(i,j) = \sum_{k=1}^{n_e} \int_{\Omega^{e_k}} \theta(\mathbf{x})\phi_j\phi_i d\Omega, \qquad (3.28)$$

$$S(i,j) = -\sum_{k=1}^{n_e} \int_{\Omega^{e_k}} \left(\nabla\phi_i \cdot (\theta(\mathbf{x})D(\mathbf{x})\nabla\phi_j) + (\mathbf{q}(\mathbf{x}) \cdot \nabla\phi_j)\phi_i\right) d\Omega - \sum_{k=1}^{n_{be_3}} \int_{\Gamma_3^{e_k}} \sigma\phi_j\phi_i d\Gamma, \qquad (3.29)$$

$$f(i) = \sum_{k=1}^{n_e} \int_{\Omega^{e_k}} (q_{so}C_s)(\mathbf{x})\phi_i d\Omega + \sum_{k=1}^{n_{be_2}} \int_{\Gamma_2^{e_k}} g_2(\mathbf{x})\phi_i d\Gamma + \sum_{k=1}^{n_{be_3}} \int_{\Gamma_3^{e_k}} g_3(\mathbf{x})\phi_i d\Gamma. \qquad (3.30)$$

It is assumed that the boundary of the domain equals the outer boundary of the elements.

Only those basis functions corresponding to nodal points in the element e_k have a non-zero contribution to the integrals for this element. So for a triangular shaped element e_k only a small number of the integrals over the element is unequal to zero. These integrals are computed and stored in an element matrix. For a linear triangle such an element matrix is a 3×3 matrix. The element vector corresponding to q_i reduces to a 3×1 vector.

The elements of the element matrix and element vector are computed with a numerical integration rule. The Newton-Cotes rule is based upon exact integration of the basis functions

$$\text{func}(\mathbf{x}) \approx \sum_{k=1}^{d+1} \text{func}(\mathbf{x_k})\phi_k(\mathbf{x}), \tag{3.31}$$

with $d+1$ the number of basis functions in the element, and application of the general rule:

Theorem 1

$$\int_{simplex} \phi_1^{m_1}\phi_2^{m_2}...\phi_{d+1}^{m_{d+1}} d\Omega = \frac{m_1!m_2!...m_{d+1}!}{(m_1+m_2+...+m_{d+1}+d)!}|\Delta|, \tag{3.32}$$

where d denotes the dimension of space [13].

$$|\Delta| = |-x_2y_3 + x_2y_1 + x_1y_3 + x_3y_2 - x_3y_1 - x_1y_2|, \tag{3.33}$$

represents two times the area of a triangle (for $d = 2$).

From (3.31) and (3.32) it follows that the Newton-Cotes rule for the triangle is defined by:

$$\int_{\Omega_{e_k}} f(\mathbf{x})d\Omega = \frac{|\Delta|}{6}\sum_{l=1}^{3} f(\mathbf{x}_l), \tag{3.34}$$

where $\mathbf{x}_l$ is the l^{th} vertex of the triangle. The Newton-Cotes rule for the boundary element is defined by:

$$\int_{\Gamma_{e_k}} f(\mathbf{x})d\Gamma = \frac{|\tilde{\Delta}|}{2}\sum_{l=1}^{2} f(\mathbf{x}_l), \tag{3.35}$$

with $|\tilde{\Delta}|$ the length of the boundary side of the triangle. For example, if x_2 and x_1 are the vertices on the boundary, then the boundary element is $[\mathbf{x}^1, \mathbf{x}^2]$ and

$$|\tilde{\Delta}| = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}. \tag{3.36}$$

A graphical representation of an element and a boundary element can be found in Figure 3.4.

Application of the Newton-Cotes rule results in the element matrices and vector (with 1,2,3 the vertices of the triangle). The mass-matrix M is

$$M^{e_k} = \begin{bmatrix} M^{e_k}(1,1) & M^{e_k}(1,2) & M^{e_k}(1,3) \\ M^{e_k}(2,1) & M^{e_k}(2,2) & M^{e_k}(2,3) \\ M^{e_k}(3,1) & M^{e_k}(3,2) & M^{e_k}(3,3) \end{bmatrix}, \tag{3.37}$$

with

$$M^{e_k}(i,j) = \frac{|\Delta|}{6}\theta(\mathbf{x}_i)\delta_{ij}. \tag{3.38}$$

For the stiffness-matrix S internal elements and boundary elements are considered separately. The element matrix for the internal element is a 3×3-matrix

$$S^{e_k} = \begin{bmatrix} S^{e_k}(1,1) & S^{e_k}(1,2) & S^{e_k}(1,3) \\ S^{e_k}(2,1) & S^{e_k}(2,2) & S^{e_k}(2,3) \\ S^{e_k}(3,1) & S^{e_k}(3,2) & S^{e_k}(3,3) \end{bmatrix}, \tag{3.39}$$

with

$$S^{e_k}(i,j) = -\frac{|\Delta|}{6}\left((\nabla\phi_i \cdot \nabla\phi_j)\sum_{l=1}^{3}(\theta(\mathbf{x}_l)D(\mathbf{x}_l)) + \nabla\phi_j \cdot \mathbf{q}(\mathbf{x}_i)\right), \tag{3.40}$$

when D is a scalar. When D is the matrix

$$D = \begin{pmatrix} D_{xx} & D_{xy} \\ D_{yx} & D_{yy} \end{pmatrix}, \tag{3.41}$$

the elements of the element matrices of S are

$$S^{e_k}(i,j) = -\frac{|\Delta|}{6}\left(\sum_{l=1}^{3}\left(\theta(\mathbf{x}_l)(\nabla\phi_j)^T D(\mathbf{x}_l)\nabla\phi_i\right) + \nabla\phi_j \cdot \mathbf{q}(\mathbf{x}_i)\right), \tag{3.42}$$

or

$$\begin{aligned} S^{e_k}(i,j) \quad = \quad & -\tfrac{|\Delta|}{6}\left(\textstyle\sum_{l=1}^{3}\left(\theta(\mathbf{x}_l)\left\{\left(D_{xx}(\mathbf{x}_l)\tfrac{\partial\phi_j}{\partial x} + D_{yx}(\mathbf{x}_l)\tfrac{\partial\phi_j}{\partial y}\right)\tfrac{\partial\phi_i}{\partial x} + \right.\right.\right. \\ & \left.\left.\left.\left(D_{xy}(\mathbf{x}_l)\tfrac{\partial\phi_j}{\partial x} + D_{yy}(\mathbf{x}_l)\tfrac{\partial\phi_j}{\partial y}\right)\tfrac{\partial\phi_i}{\partial y}\right\}\right) + \nabla\phi_j \cdot \mathbf{q}(\mathbf{x}_i)\right). \end{aligned} \tag{3.43}$$

Since linear triangles are used, the boundary is approximated by straight lines. A boundary element is the line (a side) of an element that corresponds to the boundary. Both elements can be found in Figure 3.4. The element matrices for the boundary elements on the Robbins boundary, Γ_3, are the 2×2-matrix

$$S^{e_l} = \begin{bmatrix} S^{e_l}(1,1) & S^{e_l}(1,2) \\ S^{e_l}(2,1) & S^{e_l}(2,2) \end{bmatrix}, \tag{3.44}$$

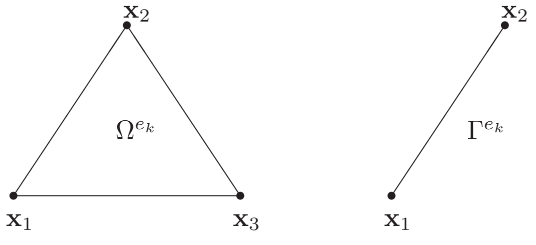


Figure 3.4: Internal element Ω^{e_k} and boundary element Γ^{e_k}.

with

$$S^{e_l}(i,j) = -\frac{|\tilde{\Delta}|}{2}\sigma(\mathbf{x}_i)\delta_{ij}, \tag{3.45}$$

For the vector f again internal and boundary elements are considered. For the internal elements the vector f is the 3×1-vector

$$f^{e_k} \;\; = \begin{bmatrix} f^{e_k}(1) \\ f^{e_k}(2) \\ f^{e_k}(3) \end{bmatrix}, \tag{3.46}$$

with

$$f^{e_k}(i) = \frac{|\Delta|}{6} q_{so}(\mathbf{x}_i) C_s(\mathbf{x}_i). \tag{3.47}$$

For the boundary elements, the vector f is the 2×1-vector

$$f^{e_l} \;\; = \begin{bmatrix} f^{e_l}(1) \\ f^{e_l}(2) \end{bmatrix}, \tag{3.48}$$

with

$$f^{e_l}(i) = \frac{|\tilde{\Delta}|}{2} g_2(\mathbf{x}_i), \tag{3.49}$$

for the natural boundary condition at Γ_2. And

$$f^{e_l}(i) = \frac{|\tilde{\Delta}|}{2} g_3(\mathbf{x}_i), \tag{3.50}$$

for the natural boundary condition at Γ_3. Note that there is no essential boundary condition because Γ_1, the boundary with the Dirichlet boundary condition, is empty.

SGA for the aquitards

In Section 3.1 the transport equation for the aquitards is derived. The transport equation for the geological aquitards becomes

$$\theta \frac{\partial C}{\partial t} = \nabla \cdot (\theta D \nabla C) - \frac{\partial q_z C}{\partial z} + q_{so} C_s, \tag{3.51}$$

with

$$D = \begin{bmatrix} D_{xx} & 0 & 0 \\ 0 & D_{yy} & 0 \\ 0 & 0 & D_{zz} \end{bmatrix},$$

and

$$D_{xx} = a_T q_z, \qquad D_{yy} = a_T q_z, \qquad D_{zz} = a_L q_z.$$

The element matrices of the Standard Galerkin Approach for the aquitards are derived equivalent to the element matrices for the aquifers. The mass-matrix M for the aquitards is equal to the mass matrix for the aquifers

$$M^{e_k} \;\; = \begin{bmatrix} M^{e_k}(1,1) & M^{e_k}(1,2) & M^{e_k}(1,3) \\ M^{e_k}(2,1) & M^{e_k}(2,2) & M^{e_k}(2,3) \\ M^{e_k}(3,1) & M^{e_k}(3,2) & M^{e_k}(3,3) \end{bmatrix}, \tag{3.52}$$

with

$$M^{e_k}(i,j) = \frac{|\Delta|}{6}\theta(\mathbf{x}_i)\delta_{ij}. \tag{3.53}$$

For the stiffness-matrix S internal elements and boundary elements are considered separately. The element matrix for the internal element is a 3 × 3-matrix

$$S^{e_k} = \begin{bmatrix} S^{e_k}(1,1) & S^{e_k}(1,2) & S^{e_k}(1,3) \\ S^{e_k}(2,1) & S^{e_k}(2,2) & S^{e_k}(2,3) \\ S^{e_k}(3,1) & S^{e_k}(3,2) & S^{e_k}(3,3) \end{bmatrix}, \tag{3.54}$$

the elements of the element matrices of S are

$$S^{e_k}(i,j) = -\frac{|\Delta|}{6}\sum_{l=1}^{3}\theta(\mathbf{x}_l)\left\{\left(D_{xx}(\mathbf{x}_l)\frac{\partial\phi_j}{\partial x}\right)\frac{\partial\phi_i}{\partial x} + \left(D_{yy}(\mathbf{x}_l)\frac{\partial\phi_j}{\partial y}\right)\frac{\partial\phi_i}{\partial y}\right\}. \tag{3.55}$$

The element matrices for the boundary elements on the Robbins boundary, Γ_3, are the 2 × 2-matrix

$$S^{e_l} = \begin{bmatrix} S^{e_l}(1,1) & S^{e_l}(1,2) \\ S^{e_l}(2,1) & S^{e_l}(2,2) \end{bmatrix}, \tag{3.56}$$

with

$$S^{e_l}(i,j) = -\frac{|\tilde{\Delta}|}{2}\sigma(\mathbf{x}_i)\delta_{ij}, \tag{3.57}$$

For the vector f again internal and boundary elements are considered. For the internal elements the vector f is the 3 × 1-vector

$$f^{e_k} = \begin{bmatrix} f^{e_k}(1) \\ f^{e_k}(2) \\ f^{e_k}(3) \end{bmatrix}, \tag{3.58}$$

with

$$f^{e_k}(i) = \frac{|\Delta|}{6}q_{so}(\mathbf{x}_i)C_s(\mathbf{x}_i). \tag{3.59}$$

For the boundary elements, the vector f is the 2 × 1-vector

$$f^{e_l} = \begin{bmatrix} f^{e_l}(1) \\ f^{e_l}(2) \end{bmatrix}, \tag{3.60}$$

with

$$f^{e_l}(i) = \frac{|\tilde{\Delta}|}{2}g_2(\mathbf{x}_i), \tag{3.61}$$

for the natural boundary condition at Γ_2. And

$$f^{e_l}(i) = \frac{|\tilde{\Delta}|}{2}g_3(\mathbf{x}_i), \tag{3.62}$$

for the natural boundary condition at Γ_3.

3.2.2 SUPG

According to [18] it can be shown that the Standard Galerkin Approach in combination with the FEM yields an accuracy of $O(h^{k+1})$, where h is some representative diameter of the triangles and k is the degree of the polynomials used in the aproximation per element (for linear shaped elements $k = 1$). However, this is only true for problems where advection does not dominate dispersion. As soon as the advection dominates, the accuracy strongly deteriorates. Inspired by upwind finite differences, upwind finite elements have been developed to preclude wiggles. These upwind methods can represent a significant improvement over the Standard Galerkin Approach, but problems have been observed with the treatment of source terms, time dependent behavior and with the generalization to multidimensions. In these cases, pronounced dispersion corrupts the true solution. For more information about these techniques and their problems, see [21].

An example of a class of upwind methods is the class of Petrov-Galerkin methods (PG), that can be used in order to obtain a better accuracy and fewer wiggles for advection dominated flows. The results of the accuracy of both the methods SGA and SUPG can be found in Chapter 4. PG methods are methods in which the test functions and the basis functions for the solution have different shapes. Split the testfunction $\eta(\mathbf{x})$ into two parts:

$$\eta(\mathbf{x}) = w(\mathbf{x}) + b(\mathbf{x}), \tag{3.63}$$

where $w(\mathbf{x})$ is the classical test function from the same function space as the solution and $b(\mathbf{x})$ is used to take care of the upwind behavior. The $w(\mathbf{x})$ part ensures the consistency of the scheme. This function must be sufficiently smooth to allow integration by parts. The function $b(\mathbf{x})$ on the other hand will be defined elementwise, which means that it may be discontinuous over the element boundaries. Rewrite the weak formulation before the application of Green's theorem (Equation (3.20)) by substitution of (3.63):

$$\int_\Omega \left\{ -\nabla \cdot (\theta D \nabla C) + \mathbf{q} \cdot \nabla C + \theta \frac{\partial C}{\partial t} - q_{so} C_s \right\} (w + b) d\Omega = 0. \tag{3.64}$$

The function $b(\mathbf{x})$ can be discontinuous over the elements, hence Green's theorem (see (3.21)) can only be applied to the $w(\mathbf{x})$ part of (3.64). After the application of this theorem, Equation (3.64) becomes:

$$\begin{aligned} \int_\Omega \left((\nabla w) \cdot (\theta D \nabla C) + (\mathbf{q} \cdot \nabla C) w + \theta \frac{\partial C}{\partial t} w \right) d\Omega + \int_{\Gamma_3} \sigma C w d\Gamma + \\ + \int_\Omega \left\{ -\nabla \cdot (\theta D \nabla C) + \mathbf{q} \cdot \nabla C + \theta \frac{\partial C}{\partial t} - q_{so} C_s \right\} b d\Omega = \\ = \int_\Omega q_{so} C_s w d\Omega + \int_{\Gamma_2} g_2 w d\Gamma + \int_{\Gamma_3} g_3 w d\Gamma. \end{aligned} \tag{3.65}$$

It is possible that $\nabla \cdot (\theta D \nabla C)$ does not exist over the element boundaries and that the integral containing the b term can only be computed by a summation over the elements. In order to solve this problem the integral containing b is split into a sum of integrals over the elements, and the inter-element contributions are neglected. Reformulation of Equation (3.65) results in:

$$\begin{aligned}
&\int_\Omega \left((\nabla w) \cdot (\theta D \nabla C) + (\mathbf{q} \cdot \nabla C) w + \theta \frac{\partial C}{\partial t} w \right) d\Omega + \int_{\Gamma_3} \sigma C w d\Gamma + \\
&+ \sum_{k=1}^{n_e} \int_{\Omega^{e_k}} \left\{ -\nabla \cdot (\theta D \nabla C) + \mathbf{q} \cdot \nabla C + \theta \frac{\partial C}{\partial t} \right\} b d\Omega = \\
&= \int_\Omega q_{so} C_s w d\Omega + \int_{\Gamma_2} g_2 w d\Gamma + \int_{\Gamma_3} g_3 w d\Gamma + \\
&+ \sum_{k=1}^{n_e} \int_{\Omega^{e_k}} q_{so} C_s b d\Omega. \qquad (3.66)
\end{aligned}$$

Note that

$$\begin{aligned}
-\nabla \cdot (\theta D \nabla C) &= \tfrac{\partial}{\partial x} \left\{ \theta D_{xx} \tfrac{\partial C}{\partial x} + \theta D_{xy} \tfrac{\partial C}{\partial y} \right\} + \tfrac{\partial}{\partial y} \left\{ \theta D_{yx} \tfrac{\partial C}{\partial x} + \theta D_{yy} \tfrac{\partial C}{\partial y} \right\} \\
&= \tfrac{\partial(\theta D_{xx})}{\partial x} \tfrac{\partial C}{\partial x} + \tfrac{\partial(\theta D_{xy})}{\partial x} \tfrac{\partial C}{\partial y} + \tfrac{\partial(\theta D_{yx})}{\partial y} \tfrac{\partial C}{\partial x} + \tfrac{\partial(\theta D_{yy})}{\partial y} \tfrac{\partial C}{\partial y} \\
&= -\nabla^T (\theta D) \nabla C,
\end{aligned} \qquad (3.67)$$

where the second derivative of the concentration C disappears because C is a linear combination of the (linear) basis functions.

After substituting the approximation for the concentration and the basis-functions for the testfunction w, the Galerkin formulation for the PG method can be derived. This results in a system of equations, $M\frac{dC}{dt} = SC + f$, with

$$M(i,j) = \sum_{k=1}^{n_e} \int_{\Omega^{e_k}} \theta(\mathbf{x}) \phi_j \left(\phi_i + b(\mathbf{x}) \right) d\Omega, \qquad (3.68)$$

For $D \in \mathbb{R}^{2\times 2}$:

$$\begin{aligned}
S(i,j) = -\sum_{k=1}^{n_e} \int_{\Omega^{e_k}} &\left[\nabla \phi_i \cdot (\theta(\mathbf{x}) D(\mathbf{x}) \nabla \phi_j) - (\nabla^T (\theta(\mathbf{x}) D(\mathbf{x})) \nabla \phi_j) b(\mathbf{x}) + \right. \\
&\left. (\mathbf{q}(\mathbf{x}) \cdot \nabla \phi_j) (\phi_i + b(\mathbf{x})) \right] d\Omega - \sum_{k=1}^{n_{be_3}} \int_{\Gamma_3^{e_k}} \sigma(\mathbf{x}) \phi_j \phi_i d\Gamma, \qquad (3.69)
\end{aligned}$$

$$f(i) = \sum_{k=1}^{n_e} \int_{\Omega^{e_k}} (q_{so} C_s)(\mathbf{x}) (\phi_i + b(\mathbf{x})) d\Omega + \sum_{k=1}^{n_{be_2}} \int_{\Gamma_2^{e_k}} g_2(\mathbf{x}) \phi_i d\Gamma + \sum_{k=1}^{n_{be_3}} \int_{\Gamma_3^{e_k}} g_3(\mathbf{x}) \phi_i d\Gamma. \qquad (3.70)$$

The choice of the function $b(\mathbf{x})$ is completely free but actually defines the type of the PG method.

Brooks and Hughes [21] tried to apply upwind only in the direction of the velocity of the flow of a more dimensional problem. They achieved this by giving the perturbation parameter b a tensor character:

$$b(\mathbf{x}) = \frac{h\xi}{2} \frac{\nabla \phi_i \cdot \mathbf{q}}{||\mathbf{q}||},$$

with $\frac{\nabla\phi_i\cdot\mathbf{q}}{||\mathbf{q}||}$ the inner product of the gradient of the basis function and the direction of the velocity and $h=\sqrt{\Delta x^2+\Delta y^2}$ some representative distance in the element, preferably in the direction of $\mathbf{q}$.

This choice of $b(\mathbf{x})$ is called the Streamline Upwind Petrov Galerkin method (SUPG), since streamlines (lines which are everywhere tangent to the velocity of the flow) are always in the direction of the velocity. The explanation in two dimensions is given. Call $\Psi = constant$ a streamline, ϕ the potential and q_x the x-component of the velocity vector $\mathbf{q}$. By definition, $q_x=\frac{\partial\phi}{\partial x}$ and $q_y=\frac{\partial\phi}{\partial y}$. Also by definition, $q_x=\frac{\partial\Psi}{\partial y}$ and $q_y=-\frac{\partial\Psi}{\partial x}$. Hence

$$\nabla\Psi=\left[\begin{array}{c}\frac{\partial\Psi}{\partial x}\\\frac{\partial\Psi}{\partial y}\end{array}\right]=\left[\begin{array}{c}-q_y\\q_x\end{array}\right].$$

The inner product $(\nabla\Psi,\nabla\phi)=0$. So if $\Psi=constant$, the level curves of ϕ and Ψ are orthogonal. The conclusion is that $\Psi=constant$ is the direction of the velocity for two dimensional cases.

The following values of ξ are commonly proposed;

Classical upwind scheme

$$\xi = sign(\alpha), \tag{3.71}$$

Il'in scheme

$$\xi = coth(\alpha)-1/\alpha, \tag{3.72}$$

Double asymptotic approximation

$$\xi=\begin{cases}\alpha/3, & -3\le\alpha\le 3,\\ sign(\alpha), & |\alpha|>3,\end{cases} \tag{3.73}$$

Critical approximation

$$\xi=\begin{cases}-1-1/\alpha, & \alpha\le -1,\\ 0, & -1\le\alpha\le 1,\\ 1-1/\alpha, & \alpha\ge 1,\end{cases} \tag{3.74}$$

α is the element Peclet number and is defined as

$$\alpha=\frac{\mathbf{q}\cdot\Delta\mathbf{x}}{2D\theta}. \tag{3.75}$$

Provided that the dispersion matrix D is symmetric ($D^T=D$) and positive definite,

$$||D||_p=\sup_{\mathbf{x}\in\mathbb{R}^2}\frac{||D\mathbf{x}||_p}{||\mathbf{x}||_p},$$

can be used in Equation (3.75). Note that if $p=2$

$$||D||_2=\max_{\lambda}(|\lambda(D)|),$$

for symmetric matrices. Note that

$$\max_{\lambda}(|\lambda(D)|)=\text{Spec}(D),$$

with $Spec(D)$ the spectral radius of the matrix D, which is defined as the radius of the smallest circle in the complex plane that contains all eigenvalues of D. Hence $||D||_2 = Spec(D)$ for symmetric matrices [22, 23].

Call $M_u^{e_k}$ the element matrix of the mass matrix corresponding to the upwind part with parameter b such that $M^{e_k} = M_g^{e_k} + M_u^{e_k}$, with $M_g^{e_k}$ the element matrix of the mass matrix corresponding to the SGA (see Equation (3.52)). Then

$$M_u^{e_k}(i,j) = \frac{|\Delta|}{6}\sum_{l=1}^{3}\theta(\mathbf{x}_l)\phi_j(\mathbf{x}_l)b(\mathbf{x}_l) = \frac{|\Delta|}{6}\theta(\mathbf{x}_j)\frac{h_j\xi(\mathbf{x}_j)}{2}\frac{\nabla\phi_i\cdot\mathbf{q}(\mathbf{x}_j)}{||\mathbf{q}(\mathbf{x}_j)||}, \quad (3.76)$$

with $||\mathbf{q}|| = \sqrt{q_x^2 + q_y^2}$. The representative distance of an element in the direction of the velocity $\mathbf{q}$, h_j is determined by the following algorithm developed by A. Segal:

```
Algorithm to determine $h_j$, the representative distance of an element in the direction of $\mathbf{q}$ :

For each vertex $i$ within the element
        $\phi_{max} = \max_k\left(|\mathbf{q}_i|\cdot\nabla\phi_k|\right)$
        if $\phi_{max} > \epsilon$
                $h(i) = \frac{||\mathbf{q}_i||}{\phi_{max}}$
        else
                $h(i) = 0.$
```

Some examples of this algorithm are given in Appendix D.

If the Classical upwind scheme is used, $\xi(\mathbf{x}_j) = sign(\alpha) = sign\left(\frac{\mathbf{q}_j\cdot\Delta\mathbf{x}}{2D\theta}\right)$, $\xi(\mathbf{x}_j)$ can be written as

$$\xi(\mathbf{x}_j) = sign(\mathbf{q}_j\cdot\Delta\mathbf{x}),$$

because the dispersion coefficient D and the porosity θ are always positive. The distances $\Delta\mathbf{x}$ in the element can be determined by

$$\Delta x = max(|x_3 - x_2|, |x_3 - x_1|, |x_2 - x_1|),$$

$$\Delta y = max(|y_3 - y_2|, |y_3 - y_1|, |y_2 - y_1|).$$

Numerical experiments have shown that for a regular grid, taking $sign(\alpha) = 1$ for all nodes gives the best results, independent on the direction of $\mathbf{q}$. Hence, in the experiments the Classical Upwind scheme is chosen with $\xi = 1$.

Call $S1_u^{e_k}$ the element matrix of the dispersion part of the stiffness matrix corresponding to the upwind part with parameter b such that $S1^{e_k} = S1_g^{e_k} + S1_u^{e_k}$, with $S1_g^{e_k}$ the element matrix of the stiffness matrix corresponding to the SGA (see Equation (3.40)). In the same way the matrices $S2_u^{e_k}$ and $S2_g^{e_k}$ (see Equation (3.40)) are defined for the advective part of the stiffness matrix.

$$S1_u^{e_k}(i,j) = \int_{\Omega^{e_k}} \nabla^T\left(\theta(\mathbf{x})D(\mathbf{x})\right)\nabla\phi_j b(\mathbf{x})d\Omega =$$

$$= \frac{|\Delta|}{6}\sum_{l=1}^{3}(\nabla^T(\theta D))(\mathbf{x}_l)\nabla\phi_j\frac{h_l\xi(\mathbf{x}_l)}{2}\frac{\nabla\phi_i\cdot\mathbf{q}(\mathbf{x}_l)}{||\mathbf{q}(\mathbf{x}_l)||}. \quad (3.77)$$

The derivatives $\nabla^T(\theta D)$ are determined by the information available in the element:

$$\nabla^T(\theta D) \approx \sum_{i=1}^{3} \theta(\mathbf{x}_i) D(\mathbf{x}_i) \nabla\phi_i(\mathbf{x}). \tag{3.78}$$

$$\begin{aligned} S2_u^{e_k}(i,j) &= -\int_{\Omega^{e_k}} (\mathbf{q}(\mathbf{x}) \cdot \nabla\phi_j) b(\mathbf{x}) d\Omega = \\ &= -\frac{|\Delta|}{6} \sum_{l=1}^{3} (\mathbf{q}(\mathbf{x}_l) \cdot \nabla\phi_j) \frac{h_l \xi(\mathbf{x}_l)}{2} \frac{\nabla\phi_i \cdot \mathbf{q}(\mathbf{x}_l)}{||\mathbf{q}(\mathbf{x}_l)||}. \end{aligned} \tag{3.79}$$

And the elements of the element vector corresponding to the upwind part are

$$f_u^{e_k}(i) = \frac{|\Delta|}{6} \sum_{l=1}^{3} q_{so}(\mathbf{x}_l) C_s(\mathbf{x}_l) b(\mathbf{x}_l), \tag{3.80}$$

The representative area $|\Delta|$ equals two times the area of a triangle and is defined as

$$|\Delta| = |-x_2y_3 + x_2y_1 + x_1y_3 + x_3y_2 - x_3y_1 - x_1y_2|, \tag{3.81}$$

as presented in Equation (3.33).

SUPG for pure advection

Mizukami [24] derived the SUPG method for linear triangular elements for the pure advection equation by taking

$$\xi = 1 - \frac{1}{\alpha + 1}, \tag{3.82}$$

in the perturbation parameter

$$b(\mathbf{x}) = \frac{h\xi}{2} \frac{\nabla\phi_i \cdot \mathbf{q}}{||\mathbf{q}||},$$

where α is the element Peclet number which is defined as

$$\alpha = \frac{\mathbf{q} \cdot \Delta\mathbf{x}}{2D\theta}. \tag{3.83}$$

The parameter Δx is not defined in [24], the advantage of this method is that it is not needed to determine this representative distance in the element. So for pure advection this results in $\xi = 1$ and for pure dispersion (i.e. diffusion) in $\xi = 0$. In [24] another methode to determine h_j, which represents the maximum element dimension in the direction of $\mathbf{q}$, is presented. The coefficient $\tilde{\tau}$ has the dimension of time and is a function of element parameters (e.g. element dimension, element Peclet number). The formula for $\tilde{\tau}$ is

$$\tilde{\tau}_i = \frac{1}{2} \left(\frac{M(i,i)}{\max_j |S2(i,j)| + S1(i,i)} \right), \tag{3.84}$$

where $M(i,j)$ is the Galerkin-type element lumped (row sum) mass matrix. $S2(i,j)$ is the Galerkin-type element advection matrix and $S1(i,i)$ is the Galerkin-type element dispersion matrix. The indices i and j denote element node numbers (1, 2 or 3). The index i of $\tilde{\tau}_i$ means that the values of $\tilde{\tau}$ generally vary with element nodes. The validity of Formula (3.84) is shown in [24].

Let,

$$b(\mathbf{x}_l) = \frac{h\xi}{2}\frac{\nabla\phi_i \cdot \mathbf{q}}{||\mathbf{q}||} = \tilde{\tau}_l \nabla\phi_i \cdot \mathbf{q}_l.$$

Hence

$$\tilde{\tau}_l = \frac{\xi}{2}\frac{h_l}{||\mathbf{q}_l||}. \tag{3.85}$$

First, determine the element matrices of the SGA in order to calculate $\tilde{\tau}$ in Equation (3.84). Then use this value for $\tilde{\tau}$ in order to determine the representative distance h of the element with the aid of Equation (3.85) and the information that $\xi = 1$ for the pure advection equation [24].

3.2.3 Mizukami-Hughes algorithm

The SUPG method does not preclude small nonphysical oscillations localized in narrow regions along sharp layers. Shock capturing methods were developed in order to obtain a method which is monotone or which at least reduces the oscillations. A basic problem of most of these methods is the design of appropriate stabilization parameters which lead to sufficiently small nonphysical oscillations without deterioration of accuracy. Mizukami and Hughes [25] introduced an interesting method for solving the steady advection-dispersion equation.

One property of this algorithm is that the solutions always satisfy the discrete maximum principle when the magnitude of the angles of the triangles of the grid are less than or equal to $\pi/2$. This is called a triangulation of the weakly acute type. When the discrete maximum principle is satisfied no spurious oscillations appear, not even in the vicinity of sharp layers. Another property is that the scheme is conservative and since it is a Petrov-Galerkin method, it is consistent. The third important property is the nonlinearity of the method, because it depends on the unknown discrete solution.

Mizukami and Hughes showed that the streamline is not always the appropriate upwind direction. First define $\mathbf{v}$ by

$$\mathbf{v} = \begin{cases} \frac{(\mathbf{q}\cdot\nabla C)\nabla C}{\|\nabla C\|^2}, & \text{if } \nabla C \neq 0, \\ \mathbf{q}, & \text{if } \nabla C = 0, \end{cases} \tag{3.86}$$

then the following equation can be obtained:

$$\mathbf{v} \cdot \nabla C = \mathbf{q} \cdot \nabla C. \tag{3.87}$$

More generally, if $\tilde{\mathbf{q}}$ is defined by

$$\tilde{\mathbf{q}} = \mathbf{q} + \mathbf{k}, \tag{3.88}$$

where $\mathbf{k}$ is perpendicular to ∇C, but otherwise arbitrary, then Equation (3.87) may be generalized to

$$\tilde{\mathbf{q}} \cdot \nabla C = \mathbf{q} \cdot \nabla C. \tag{3.89}$$

This suggests that the streamline may not always be the appropriate two-dimensional generalization of the one-dimensional upwind direction.

The Mizukami-Hughes method is a Petrov-Galerkin method (see Section 3.2.2) with weighting functions

$$\eta(\mathbf{x}) = w(\mathbf{x}) + \sum_{e_k \in T_h} b_i^{e_k} \chi_{e_k}, \qquad i = 1, \cdots, M_h. \tag{3.90}$$

with T_h the triangulation consisting of a finite number of triangular elements e_k.

$$\chi_{e_k}(\mathbf{x}) = \begin{cases} 1 & \mathbf{x} \in e_k, \\ 0 & \mathbf{x} \notin e_k. \end{cases}$$

$w(\mathbf{x})$ is again the classical testfunction.

The weighting function in an element is defined as

$$\eta_i = w_i + b_i, \tag{3.91}$$

where w_i are the linear basis functions and b_i are constants which satisfy the following conditions:

$$\begin{array}{l} b_i \geq -\frac{1}{3}, \\ b_1 + b_2 + b_3 = 0. \end{array} \tag{3.92}$$

These conditions result in:

$$\begin{array}{l} \int_{\Omega_{e_k}} \eta_i d\Omega \geq 0, \\ \eta_1 + \eta_2 + \eta_3 = 1, \end{array} \tag{3.93}$$

where Ω_{e_k} is again the domain of an element.

Call $M_{mh}^{e_k}$ the element matrix of the mass matrix corresponding to the Mizukami Hughes part such that $M^{e_k} = M_g^{e_k} + M_{mh}^{e_k}$, with $M_g^{e_k}$ the element matrix of the mass matrix corresponding to the SGA (see Equation (3.52)). Then, by substitution of $\eta_i = \phi_i(\mathbf{x})$,

$$M_{mh}^{e_k}(i,j) = \frac{|\Delta|}{6} \sum_{l=1}^{3} \theta(\mathbf{x}_l)\phi_j(\mathbf{x}_l) b_i = \frac{|\Delta|}{6} \theta(\mathbf{x}_j) b_i. \tag{3.94}$$

$S2_{mh}^{e_k}$ is the element matrix of the advective part of the stiffness matrix corresponding to the Mizukami Hughes algorithm such that $S2_{mh}^{e_k} = S2_g^{e_k} + S2_{mh}^{e_k}$, with $S2_g^{e_k}$ the element matrix corresponding to the SGA.

$$\begin{array}{rcl} S2_{mh}^{e_k}(i,j) & = & \int_{\Omega_{e_k}} \eta_i \mathbf{q} \cdot \nabla w_j d\Omega \\ & = & \int_{\Omega_{e_k}} \eta_i d\Omega \, (\mathbf{q} \cdot \nabla w_j) \\ & = & -\frac{|\Delta|}{2} (\mathbf{q} \cdot \nabla \phi_j) b_i. \end{array} \tag{3.95}$$

with $\mathbf{q}$ the average (centroid) velocity in an element

$$\mathbf{q} = \frac{\mathbf{q(x_1) + q(x_2) + q(x_3)}}{3}.$$

As $\int_{\Omega_{e_k}} \eta_i d\Omega \leq 0$, the sign of each $S2_{mh}^{e_k}(i,j)$ is determined by the sign of $\mathbf{q} \cdot \nabla \eta_j$.

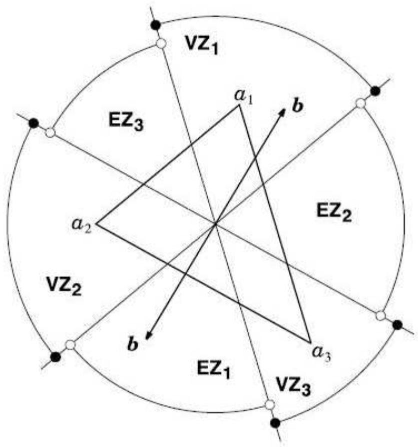


Figure 3.5: Definition of edge zones (EZ_i) and vertex zones (VZ_i) [25].

Now, define the *vertex zones* and the *edge zones* of an element, see Figure 3.5 for the definition of the vertex and element zones and [25] for more information. The boundary of two adjacent zones is included in the vertex zone. It is assumed that **q** points into the vertex zone or the edge zone of node 1 without loss of generality. If **q** lies in the vertex zone of node 1, then

$$\mathbf{q}\cdot\nabla\phi_1 > 0, \qquad \mathbf{q}\cdot\nabla\phi_2 \leq 0, \qquad \mathbf{q}\cdot\nabla\phi_3 \leq 0. \tag{3.96}$$

If **q** lies in the edge zone of node 1, then

$$\mathbf{q}\cdot\nabla\phi_1 < 0, \qquad \mathbf{q}\cdot\nabla\phi_2 > 0, \qquad \mathbf{q}\cdot\nabla\phi_3 > 0. \tag{3.97}$$

This vertex is called number 1. Vertex number 2 is the first vertex anticlockwise and vertex number 3 the second vertex anticlockwise in the element. In the case of (3.96), by setting the coefficients b_i as

$$b_1 = \frac{2}{3}, \qquad b_1 = \frac{-1}{3}, \qquad b_1 = \frac{-1}{3},$$

the signs of the element matrix $S2^{e_k}_{mh}(i,j)$ become

$$sign(S2^{e_k}_{mh}(i,j)) = \begin{bmatrix} + & - & - \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix},$$

where $sign(a)$ is defined as

$$sign(a) \quad = \begin{cases} +, & \text{if } a > 0, \\ -, & \text{if } a < 0, \\ 0, & \text{if } a = 0. \end{cases}$$

This matrix is of nonnegative type (i.e. off-diagonal entries of the matrix are nonpositive and the sum of the entries in each row is nonnegative).

On the other hand, in the case of (3.97), it is impossible to obtain any matrix of the nonnegative type because of the conditions (3.92). This is also the reason why the SUPG method does not satisfy the discrete maximum principle. Thus, define another convection matrix $S2^{e_k}_{mh}\tilde{}(i,j)$ by

$$S2^{e_k\tilde{}}_{mh}(i,j) = \int_{\Omega_{e_k}} \eta_i d\Omega(\tilde{\mathbf{q}}_i \cdot \nabla w_j), \tag{3.98}$$

where $\tilde{\mathbf{q}}_\mathbf{1}$ is defined by

$$\tilde{\mathbf{q}}_\mathbf{1} = \mathbf{q}, \tag{3.99}$$

$\tilde{\mathbf{q}}_\mathbf{2}$ by

$$\tilde{\mathbf{q}}_\mathbf{2} = \mathbf{q} + \mathbf{k}_2, \qquad \mathbf{k}_2 \perp \nabla C, \tag{3.100}$$

and $\tilde{\mathbf{q}}_\mathbf{3}$ by

$$\tilde{\mathbf{q}}_\mathbf{3} = \mathbf{q} + \mathbf{k}_3, \qquad \mathbf{k}_3 \perp \nabla C. \tag{3.101}$$

$\mathbf{k}_2 \perp \nabla C$ is equivalent to $\mathbf{k}_2 \cdot \nabla C = 0$ where

$$\mathbf{k}_2 \ = \left[\begin{array}{c} k_{2_x} \\ k_{2_y} \end{array} \right].$$

Apart from $\mathbf{k}_2 \perp \nabla C$, $\mathbf{k}_2$ can be chosen arbitrary. For example take $k_{2_x} = 1$, which results in $k_{2_y} = \frac{-\partial C/dx}{\partial C/dy}$. In general,

$$k_{2_y} = -k_{2_x} \frac{\partial C/dx}{\partial C/dy}.$$

Note that $C = \sum_{j=1}^{3} C_j \phi_j$, hence

$$\nabla C = \sum_{j=1}^{3} C_j \nabla \phi_j.$$

$\nabla \phi_j$ can be found in the Equations (3.16), (3.17) and (3.18).

If there exists $\tilde{q}_2$ which lies in the vertex zone of node 2, i.e., which satisfies

$$\tilde{\mathbf{q}}_\mathbf{2} \cdot \nabla \phi_1 > 0, \qquad \tilde{\mathbf{q}}_\mathbf{2} \cdot \nabla \phi_2 \leq 0, \qquad \tilde{\mathbf{q}}_\mathbf{2} \cdot \nabla \phi_3 \leq 0, \tag{3.102}$$

then by substituting such $\tilde{\mathbf{q}}_\mathbf{2}$ into (3.98) the element matrix of the advective part is again of nonnegative type. The values for b can be found in Figure 3.6. Information about all other possibilities can be found in [25].

Substitution of $\eta_j = \phi_j(\mathbf{x})$ results in $\nabla w_j = \nabla \phi_j$ because b_j are constant: $\nabla \eta_j = \nabla(w_j + b_j) = \nabla w_j$. The element matrix of the stiffness matrix of the advective part in this case becomes

$$S2^{e_k}_{mh}(i,j) = -\frac{|\Delta|}{6} \sum_{l=1}^{3} (\tilde{\mathbf{q}}_i \cdot \nabla \phi_j) b_i = -\frac{|\Delta|}{2} (\tilde{\mathbf{q}}_i \cdot \nabla \phi_j) b_i, \tag{3.103}$$

Note that the element matrix of the dispersive part of the stiffness matrix corresponding to the Mizukami Hughes algorithm $S1^{e_k}_{mh} = 0$ when θD is constant within an element.

Equivalently, the right hand side can be defined as

$$f_{mh}^{e_k}(i) = \frac{|\Delta|}{2} q_{so}C_s b_i, \tag{3.104}$$

with

$$q_{so}C_s = q_{so}C_s(\mathbf{x}_1) + q_{so}C_s(\mathbf{x}_2) + q_{so}C_s(\mathbf{x}_3).$$

The full algorithm for the determination of b_i can be found in Figure 3.6. In this figure

$$\begin{array}{llll}
(A) & \mathbf{q}\cdot\nabla\phi_1 > 0, & \mathbf{q}\cdot\nabla\phi_2 \le 0, & \mathbf{q}\cdot\nabla\phi_3 \le 0, \\
(B) & \mathbf{q}\cdot\nabla\phi_1 < 0, & \mathbf{q}\cdot\nabla\phi_2 > 0, & \mathbf{q}\cdot\nabla\phi_3 > 0, \\
(C) & \tilde{\mathbf{q}}_\mathbf{2}\cdot\nabla\phi_1 < 0, & \tilde{\mathbf{q}}_\mathbf{2}\cdot\nabla\phi_2 > 0, & \tilde{\mathbf{q}}_\mathbf{2}\cdot\nabla\phi_3 < 0, \\
(D) & \tilde{\mathbf{q}}_\mathbf{3}\cdot\nabla\phi_1 < 0, & \tilde{\mathbf{q}}_\mathbf{3}\cdot\nabla\phi_2 < 0, & \tilde{\mathbf{q}}_\mathbf{3}\cdot\nabla\phi_3 > 0.
\end{array} \tag{3.105}$$

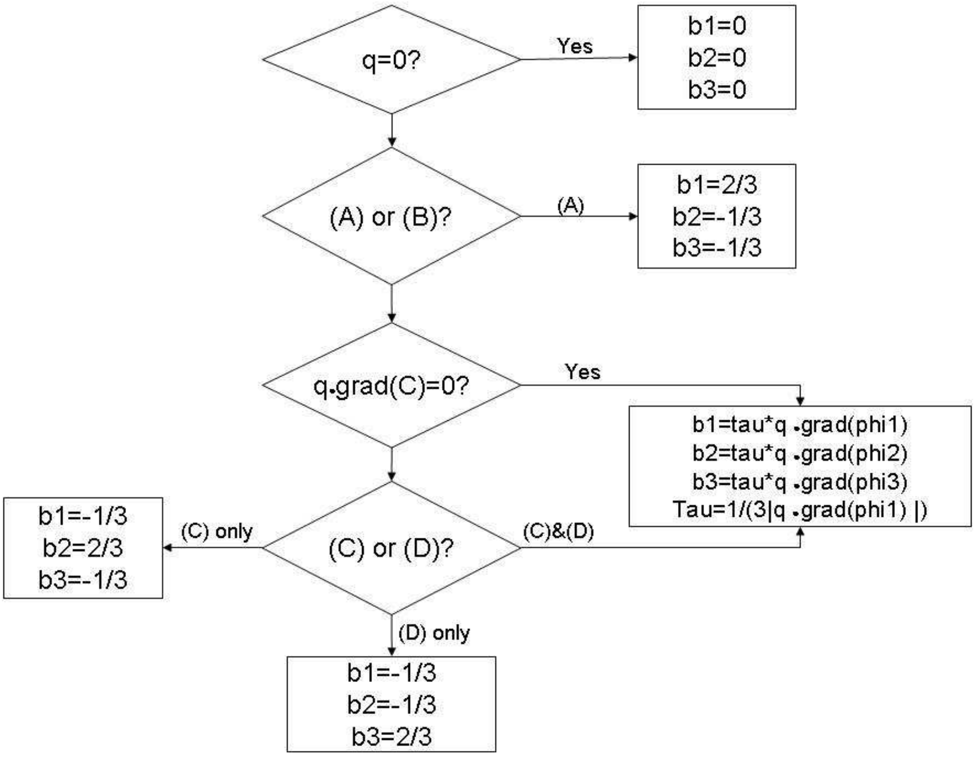


Figure 3.6: Flow chart - an algorithm for the determination of b_i [25]. See the Equations (3.105) for the definitions of (A), (B), (C) and (D).

3.3 Spatial discretization transport equation: vertical direction

3.3.1 FVM

Consider the solute transport equation in one dimension:

$$\theta \frac{\partial C}{\partial t} + \frac{\partial q_z C}{\partial z} - \frac{\partial}{\partial z}\left(\theta D \frac{\partial C}{\partial z}\right) = q_{so} C_s, \qquad z \in \Omega = [0, H] \text{ and } t \in [0, T]. \tag{3.106}$$

The domain Ω equals the hight of the subsurface and is subdivided into segments Ω_j, $j = 1, ..., J$ corresponding to the aquifers and aquitards, as shown in the Figures 3.1 and 3.2. The segments are called cells and the cell-length, denoted by Δz_j for the j^{th} cell, is called the mesh width. Integrate equation (3.106) over Ω_j and approximate this integral by

$$\Delta z_j \theta_j \frac{\partial C}{\partial t} - F|_{j-1/2}^{j+1/2} = \Delta z_j \left(q_{so} C_s\right)_j, \qquad j = 1, ..., J, \tag{3.107}$$

with

$$\begin{aligned} F|_{j-1/2}^{j+1/2} &= F_{j+1/2} - F_{j-1/2}, \\ F_{j+1/2} &= F(z_{j+1/2}), \\ F(z) &= \theta D \frac{\partial C}{\partial z} - qC. \end{aligned} \tag{3.108}$$

Here $F(z)$ is called the flux, $Fa(z) = -qC$ the advective flux and $Fd(z) = \theta D \frac{\partial C}{\partial z}$ the dispersive flux. Equation (3.107) can be rewritten as

$$\theta_j \frac{\partial C}{\partial t} + \frac{-F_{j+1/2} + F_{j-1/2}}{\Delta z_j} = \left(q_{so} C_s\right)_j. \tag{3.109}$$

The flux $F_{j+1/2}$ has to be approximated in terms of neighboring grid functions. The dispersive flux Fd is discretized in space with central differences

$$\begin{aligned} Fd_{j+1/2} &= \left(\theta D \tfrac{dC}{dx}\right)_{j+1/2} \approx (\theta D)_{j+1/2} \left(\tfrac{C_{j+1} - C_j}{\Delta z_{j+1/2}}\right), \\ \Delta z_{j+1/2} &\approx \tfrac{1}{2}(\Delta z_j + \Delta z_{j+1}). \end{aligned} \tag{3.110}$$

As can be seen in Figure 3.1, the porosity θ and the dispersivity D are known at j and $j + 1$, not at $j + 1/2$. The quantities D and θ are assumed constant per element, so use

$$Fd_{j+1/2} = \left(\theta D \frac{dC}{dz}\right)_{j+1/2} \approx (\theta D)_j \left(\frac{C_{j+1} - C_j}{\Delta z_{j+1/2}}\right), \tag{3.111}$$

$$Fd_{j-1/2} = \left(\theta D \frac{dC}{dz}\right)_{j-1/2} \approx (\theta D)_j \left(\frac{C_j - C_{j-1}}{\Delta z_{j-1/2}}\right) \tag{3.112}$$

Note that it can be considered to take the average of D_j and D_{j+1} in above equations.

For the advective flux Fa first order upwind discretization is used:

$$Fa_{j+1/2} = -\left(qC\right)_{j+1/2} \approx -\frac{1}{2}\left(q_{z_{j+1/2}} + |q_{z_{j+1/2}}|\right) C_j - \frac{1}{2}\left(q_{z_{j+1/2}} - |q_{z_{j+1/2}}|\right) C_{j+1}. \tag{3.113}$$

For the aquifers, indeed $q_{j+1/2}$ and $q_{j-1/2}$ are known. The velocity q_{z_j} is assumed to be constant within element j, so for the aquitards it is used

$$Fa_{j+1/2} = -\left(q_z C\right)_{j+1/2} \approx -\frac{1}{2}\left(q_{z_j} + |q_{z_j}|\right) C_j - \frac{1}{2}\left(q_{z_j} - |q_{z_j}|\right) C_{j+1}, \tag{3.114}$$

$$Fa_{j-1/2} = -\left(q_z C\right)_{j-1/2} \approx -\frac{1}{2}\left(q_{z_j} + |q_{z_j}|\right) C_{j-1} - \frac{1}{2}\left(q_{z_j} - |q_{z_j}|\right) C_j. \tag{3.115}$$

For advective dominated problems it is expected that information about the concentration can be found backwards in space. For a system of equations there might be several waves propagating at different speeds and perhaps in different directions. It makes sense to use the knowledge of the structure of the solution to determine better numerical flux functions. This idea gives rise to *upwind* methods in which the information for the concentration is obtained by looking in the direction from which this information should be coming.

For a scalar advection equation (q constant), there is only one speed, which is either positive or negative. So an upwind method is typically an one-sided method with first order accuracy in space. For the one dimensional advection-dispersion equation the inequality $|\frac{q\tau}{\Delta x}| \leq 1$ must be satisfied in order for this method to be stable. This condition is known as the CFL-condition.

The CFL condition is a necessary condition that must be satisfied by any finite volume method if stability and convergence to the solution of the differential equation as the grid is refined is expected. Its formal definition is

Definition 1 *The CFL condition is defined as: a numerical method is convergent if and only if its numerical domain of dependence tends to the true domain of dependence of the PDE, in the limit as τ and Δx go to zero.*

In Section 4.4 of Levèque [26] the CFL condition is derived for the one dimensional advection equation with a three-point stencil

$$\mu \equiv |\frac{q\tau}{\Delta x}| \leq 1. \tag{3.116}$$

This condition holds also for the advection-dispersion equation.

TVD method

Solutions produced by standard discretization techniques are typically corrupted by nonphysical oscillations and/or excessive numerical dispersion. Traditionally, these problems have been dealt with by means of a nonlinear shock-capturing viscosity, like high resolution methods with limiter. Modern high-resolution schemes are based on flux/slope limiters which switch between linear high- and low-order discretizations adaptively depending on the smoothness of the solution.

Definition 1 *For one dimension, a method is called* Total Variation Diminishing (TVD) *if, for any set of data Q^n, the values Q^{n+1} computed by the method satisfy*

$$TV(Q^{n+1}) \leq TV(Q^n), \tag{3.117}$$

with

$$TV(Q^n) = \sum_{i=-\infty}^{\infty} |Q_i^n - Q_{i-1}^n|. \tag{3.118}$$

If a method is TVD, then in particular for data that are initially monotone, say

$$Q_i^n \geq Q_{i+1}^n \qquad \text{for all } i,$$

the data will remain monotone in all future time steps. Hence if a single propagating discontinuity is discretized, the discontinuity may become smeared in future time steps but cannot become oscillatory. A TVD method is monotonicity-preserving. This is proved in [27] for the hyperbolic conservation law.

Definition 2 *A method is called* monotonicity-preserving *if*

$$Q_i^n \geq Q_{i+1}^n \qquad \textit{for all } i,$$

implies that

$$Q_i^{n+1} \geq Q_{i+1}^{n+1} \qquad \textit{for all } i.$$

This implies that a TVD method is stable. Note that stability plus consistency implies convergence. This is known as Lax's equivalence theorem. The definitions of *consistency*, *stability*, *convergence*, *local truncation error* and *global truncation error* can be found in Appendix B [26].

To obtain a second-order accurate discretization in space for the advective part, a *high-resolution method with nonzero slope* is used as derived in [26]. A nonzero slope is chosen in such a way that the slope approximates the derivative over the ith cell.

Assume the velocity $q > 0$ and $|q\tau/\Delta x| \leq 1$ as is required by the CFL condition, then the advective flux in Equation (3.115) can be written as

$$Fa_{j+1/2}^n = -\left(qC\right)_{j+1/2}^n \approx -q_{j+1/2}C_j^n.$$

With a nonzero slope the advective flux becomes

$$Fa_{j+1/2}^n = -q_{j+1/2}C_j^n - \frac{1}{2}q_{j+1/2}(\Delta x - q_{j+1/2}\tau)\sigma_j^n. \tag{3.119}$$

Three possibilities for the nonzero slope are:

$$\text{Centered slope:} \qquad \sigma_j^n = \frac{C_{j+1}^n - C_{j-1}^n}{2\Delta x_j} \qquad \text{(Fromm)}, \tag{3.120}$$

$$\text{Upwind slope:} \qquad \sigma_j^n = \frac{C_j^n - C_{j-1}^n}{\Delta x_j} \qquad \text{(Beam-Warming)}, \tag{3.121}$$

$$\text{Downwind slope:} \qquad \sigma_j^n = \frac{C_{j+1}^n - C_j^n}{\Delta x_j} \qquad \text{(Lax-Wendroff)}, \tag{3.122}$$

Second-order accurate methods such as Lax-Wendroff or Beam-Warming give much better accuracy on smooth solutions than the upwind method, but fail near discontinuities, where oscillations are generated. In fact for the advective equation, according to [26], even when the solution is smooth, oscillations may appear due to the dispersive nature of these methods. Upwind methods have the advantage that they cannot introduce oscillations, so they keep the solution monotonically varying in regions where the solution should be monotone. The disadvantage on the other hand is that they smear the solution.

High-resolution methods combine the best features of both the upwind and the second-order accurate methods. Second-order accuracy is obtained where possible, but it is not insisted in regions where the solution is not behaving smoothly. The idea is to apply some form of limiter that changes the magnitude of the correction actually used, depending on how the solution is behaving. This leads to the so-called slope-limiter methods.

The first order upwind method is TVD for the advection equation and has the advantage that it cannot introduce oscillations but the disadvantage that it smears the solution. The Fromm, Beam-Warming and Lax-Wendroff methods are not unconditionally TVD methods.

Take again the advective flux as described in Equation (3.119). One choice of slope that gives second-order accuracy for smooth solutions while still satisfying the TVD property is the *minmod slope*, which is a slope-limiter method

$$\sigma_j^n = minmod\left(\frac{C_j^n - C_{j-1}^n}{\Delta x}, \frac{C_{j+1}^n - C_j^n}{\Delta x}\right), \tag{3.123}$$

where the minmod function of two arguments is defined by

$$minmod(a,b) = \begin{cases} a & \text{if } |a| < |b| \text{ and } ab > 0, \\ b & \text{if } |b| < |a| \text{ and } ab > 0, \\ 0 & \text{if } ab \leq 0. \end{cases} \tag{3.124}$$

Another popular choice is the *monotonized central-difference limiter* (MC limiter)

$$\sigma_j^n = minmod\left(\left(\frac{C_{j+1}^n - C_{j-1}^n}{2\Delta x_j}\right), 2\left(\frac{C_j^n - C_{j-1}^n}{\Delta x_j}\right), 2\left(\frac{C_{j+1}^n - C_j^n}{\Delta x_j}\right)\right), \tag{3.125}$$

where the midmod function of three arguments is defined by

$$minmod(a,b,c) = minmod(a, minmod(b,c)). \tag{3.126}$$

This compares the central difference of Fromm's method with twice the one-sided slope to either side. In smooth regions this reduces to the centered slope of Fromm's method [19].

In Appendix G 1D numerical experiments can be found for the MC limiter.

3.3.2 FDM

In the vertical direction the Finite Difference Method can be used instead of the Finite Volume Method. Divergence free flow is considered ($\nabla \mathbf{q} = 0$), hence the 1D transport equation in the vertical direction is

$$\theta \frac{\partial C}{\partial t} + q_z \frac{\partial C}{\partial z} - \frac{\partial}{\partial z}\left(\theta D \frac{\partial C}{\partial z}\right) = q_{so} C_s, \qquad z \in \Omega = [0,H] \text{ and } t \in [0,T]. \tag{3.127}$$

With central differences for the spatial discretization, Equation (3.106) becomes

$$\begin{aligned}&\theta_i \tfrac{\partial C}{\partial t} + q_{z_i} \tfrac{C_{i+1}-C_{i-1}}{2\Delta z} - \tfrac{C_{i+1}-C_{i-1}}{2\Delta z} \tfrac{\theta_{i+1}D_{i+1}-\theta_{i-1}D_{i-1}}{2\Delta z} \\ &-\theta_i D_i \tfrac{C_{i+1}-2C_i+C_{i-1}}{\Delta z^2} = q_{so_i} C_{s_i},\end{aligned} \tag{3.128}$$

for the aquitards. Where $i+1$ denotes the vertex above vertex i and $i-1$ the vertex beneath vertex i.

For the aquifers

$$\begin{aligned}&-\tfrac{C_{i+1}-C_{i-1}}{2\Delta z} \tfrac{\theta_{i+1}D_{i+1}-\theta_{i-1}D_{i-1}}{2\Delta z} - \theta_i D_i \tfrac{C_{i+1}-2C_i+C_{i-1}}{\Delta z^2} + \\ &+q_{z_{i+1/2}} \tfrac{C_{i+1}}{2\Delta z} - q_{z_{i-1/2}} \tfrac{C_{i-1}}{2\Delta z} + \theta_i \tfrac{\partial C}{\partial t} = q_{so_i} C_{s_i},\end{aligned} \tag{3.129}$$

is used because the velocity in the upward direction is unknown in the vertex.

The upwind schemes for the aquitards and aquifers are

$$\begin{aligned}&-\tfrac{C_{i+1}-C_{i-1}}{2\Delta z} \tfrac{\theta_{i+1}D_{i+1}-\theta_{i-1}D_{i-1}}{2\Delta z} - \theta_i D_i \tfrac{C_{i+1}-2C_i+C_{i-1}}{\Delta z^2} + \\ &+\tfrac{1}{\Delta z}\left(\tfrac{1}{2}(q_{z_i} + |q_{z_i}|)C_i + \tfrac{1}{2}(q_{z_i} - |q_{z_i}|)C_{i+1} - \tfrac{1}{2}(q_{z_i} + |q_{z_i}|)C_{i-1} - \tfrac{1}{2}(q_{z_i} - |q_{z_i}|)C_i\right) + \\ &+\theta_i \tfrac{\partial C}{\partial t} = q_{so_i} C_{s_i},\end{aligned} \tag{3.130}$$

for the aquitards and

$$\begin{aligned}&-\tfrac{C_{i+1}-C_{i-1}}{2\Delta z} \tfrac{\theta_{i+1}D_{i+1}-\theta_{i-1}D_{i-1}}{2\Delta z} - \theta_i D_i \tfrac{C_{i+1}-2C_i+C_{i-1}}{\Delta z^2} + \\ &+\tfrac{1}{\Delta z}\left(\tfrac{1}{2}(q_{z_{i+/2}} + |q_{z_{i+1/2}}|)C_i + \tfrac{1}{2}(q_{z_{i+1/2}} - |q_{z_{i+1/2}}|)C_{i+1} - \right. \\ &\left.\tfrac{1}{2}(q_{z_{i-1/2}} + |q_{z_{i-1/2}}|)C_{i-1} - \tfrac{1}{2}(q_{z_{i-1/2}} - |q_{z_{i-1/2}}|)C_i\right) + \theta_i \tfrac{\partial C}{\partial t} = q_{so_i} C_{s_i},\end{aligned} \tag{3.131}$$

for the aquifers. A system of equations is derived of the form

$$M\frac{dC}{dt} = SC + f.$$

This system forms together with the system for the 2D horizontal FEM the system of equations for the 3D model.

Note that the matrix M of the one dimensional FDM does not equal the mass matrix M of the FEM. Equations (3.130) and (3.131) have to be rewritten in order to substitude the third dimension in the system of equations. This can be done by multiplication of (3.130) and (3.131) by $\frac{M_{FEM}(i,i)}{\theta}$, where $M_{FEM}(i,i)$ equals the i^{th} diagonal element of the mass matrix M formed by the FEM.

3.4 Temporal discretization

The system of ordinary differential equations has to be discretized in time. A choice has to be made between the one-step and multi-step methods. Here one-step methods are considered, so only information of the preceding time-step is used and not of previous time-steps.

The ω-method is given by:

$$\left(\frac{M}{\tau} - \omega S\right) C^{n+1} = \left(\frac{M}{\tau} + (1-\omega)S\right) C^n + \left((1-\omega)f^n + \omega f^{n+1}\right). \tag{3.132}$$

M is the mass-matrix as defined in equation (3.25) and S the stiffness-matrix as defined in equation (3.26). The most common values for ω are:

$\omega = 0$	Forward Euler method;
$\omega = \frac{1}{2}$	Crank-Nicolson method;
$\omega = 1$	Backward Euler method.

In the literature it is common to split the matrix S into an advective and dispersive part. Say $S = S1 + S2$, with $S1$ the stiffness matrix for the dispersive part and $S2$ the stiffness matrix for the advective part.

A common used option is Backward Euler for the dispersive part and Forward Euler for the advective part. For this method better conditions for the stepsize can be derived in order to avoid wiggles. With $S1$ the matrix for the dispersive part and $S2$ the matrix for the advective part, this scheme results in

$$\left(\frac{M}{\tau} - S1\right) C^{n+1} = \left(\frac{M}{\tau} + S2\right) C^n + f^n. \tag{3.133}$$

In the Interim Master's thesis [19] research was done on other methods for the temporal discretization, like the Crank-Nicholson method and the Runge-Kutta-2 method. But the combination Backward Euler for the dispersive part and Forward Euler for the advective part of the stiffnessmatrix gave the best results. In Table 3.4 the conclusions of those methods can be found for the one dimensional problem.

	$T1$	$T2$	$T3$
stability condition	$\lvert\frac{q\tau}{\Delta x}\rvert \leq 1$	unconditionally stable	$\lvert\frac{q\tau}{\Delta x}\rvert \leq 1$ and $\frac{D\tau}{\Delta x^2} \leq \frac{1}{2}$
Accuracy	$O(\tau)$	$O(\tau^2)$	$O(\tau^2)$
Work	implicit scheme	implicit scheme	explicit but a one-step method with two stages
Numerical dispersion	Less	More	More

Table 3.4: Characteristics of the temporal discretization schemes Backward Euler for the dispersion part and Forward Euler for the advection part ($T1$), Crank-Nicholson ($T2$) and Runge-Kutta-2 ($T3$) for the 1D advection dispersion equation [19].

3.5 Stability and accuracy

Stability

In the Interim Master's thesis [19] the stability conditions for all in Section 3.4 mentioned temporal discretization methods are given. Some details can be found in Appendix F. The used temporal discretization method uses the Forward Euler method (explicit) for the advective part and Backward Euler (implicit) for the dispersive part. The Von Neumann stability in [23] results in the stability condition

$$|\frac{q\tau}{\theta\Delta x}| \leq 1, \tag{3.134}$$

for one dimensional problems. This condition is called the CFL condition, named after Courant, Friedrich and Levy. The dispersive part is discretized with Euler Backward which results in unconditional stability. Note that the seepage velocity $\mathbf{v}$ is defined by

$$\mathbf{v} = \frac{\mathbf{q}}{\theta}, \tag{3.135}$$

with θ the porosity of the subsurface. The seepage velocity, or Darcy velocity divided by the porosity is used for the CFL condition.

For two dimensional problems with the same temporal discretization the CLF condition can be defined by

$$|\frac{q_x\tau}{\theta\Delta x}| + |\frac{q_y\tau}{\theta\Delta y}| \leq 1, \tag{3.136}$$

or by

$$\max\left(|\frac{q_x\tau}{\theta\Delta x}|, |\frac{q_x\tau}{\theta\Delta x}|\right) \leq 1. \tag{3.137}$$

Note that Equation (3.136) leads to a condition with the smallest time step, hence the most safe condition.

For the three dimensional problem the condition

$$|\frac{q_x\tau}{\theta\Delta x}| + |\frac{q_y\tau}{\theta\Delta y}| + |\frac{q_y\tau}{\theta\Delta y}| \leq 1, \tag{3.138}$$

leads to the most stern demands for the time step τ.

Accuracy

The used temporal discretization scheme which uses Backward Euler for the dispersion part and Forward Euler for the advection part and the source term is first order accurate ($O(\tau)$).

The spatial discretization method Standard Galerkin Approach is second order accurate if all angles of the triangles are smaller than 135^0 (see [13] or [19] for more information). The SUPG classicial upwind method is first order accurate. The upwind version of the finite difference method is also first order accurate. The accuracy of the finite volume methods is analysed in [26]. The MC-limiter is second-order accurate, where the solution is smooth.

Hence the three dimensional method which uses SUPG classical upwind or the Mizukami Hughes algorithm for the x- and y-direction and the FDM upwind for the z-direction is first order accurate.

3.6 Method to solve the system of equations

The system of Equations

$$\left(\frac{M}{\tau} - S1\right) C^{n+1} = \left(\frac{M}{\tau} + S2\right) C^n + f^n, \tag{3.139}$$

has to be solved in order to determine the concentration C at the next time step. Let

$$A = \frac{M}{\tau} - S1, \tag{3.140}$$

$$b = \left(\frac{M}{\tau} + S2\right) C^n + f^n. \tag{3.141}$$

Now, the system $AC^{n+1} = b$ has to be solved.

This system of equations can be solved with a *direct method*. A popular method is the Gaussian elimination method, also known as the LU decomposition. This method is the method of choice when the matrix A is square nonsingular, dense and relatively small. for small two dimensional problems, this method will often satisfy.

For bigger problems, a *basic iterative method* may be more useful. The matrix A is sparse, so iterative methods for the solution of the linear system of equations are useful. A good guess for the concentration C^{n+1} is available, namely the calculated concentration of the previous loop of the coupled problem. Also because of this characteristic of the problem, iterative methods may be more advantageous.

Use the following general iteration

$$C^{i+1} = QC^i + s, \qquad (i = 0, 1, 2, ...), \tag{3.142}$$

such that the system $C = QC + s$ is equivalent to the original problem. Q is called the iteration matrix. The simplest iteration scheme is the Richardson iteration (p. 39 [22]). Two other well-known methods are the Jacobi and the Gauss-Seidel methods. In these methods a splitting of the matrix A takes place in order to construct the matrix Q.

When A is symmetric ($A = A^T$) and positive definite ($x^T Ax > 0$ for $x \neq 0$) the Conjugate Gradient (CG) method can be used. In order to obtain faster convergence, the Preconditioned Conjugate Gradient (PCG) method may be used. A preconditioner is a matrix that transforms the linear system such that the transformed system has the same solution but the transformed coefficient matrix has a more favorable spectrum (p. 66 [22]).

When the matrix Q is only symmetric and not positive definite, methods as discussed in Chapter 5.3 of [22] may be used. One of those methods is a BiCG Type Method. When advection is involved in the problem, the matrix A used in the CGM is no longer symmetric and therefor the CGM cannot be used. Another method is used, a BiCG Type Method. In this type of method there are short recurrences but there is no optimality property. In the numerical experiments, the Bi-CGSTAB method is used, more information can be found in [22].

According to [28] Multigrid methods are faster than Conjugate Gradient methods but require more complicated and individual programming.

3.7 Flow equation

The flow equation (2.12) derived in Section 2.1, time independent and two dimensional in the x-direction and z-direction.The freshwater hydraulic conductivity is assumed to be constant (k). This flow equation is given by

$$\frac{\partial}{\partial x}\left(\rho k_f \frac{\partial h_f}{\partial x}\right) + \frac{\partial}{\partial z}\left(\rho k_f \frac{\partial h_f}{\partial z} + \frac{\rho - \rho_f}{\rho_f}\right) = -\rho q', \tag{3.143}$$

with k the freshwater hydraulic conductivity

$$k_{f_{ij}} = \frac{\kappa_{ij}\rho_f g}{\mu_f}, \tag{3.144}$$

and h_f the freshwater head. This flow equation is numerically solved with the Standard Galerkin Approach.

Equation (3.143) is multiplied by a test function η satisfying the homogeneous essential boundary condition $\eta|_{\Gamma_1} = 0$ and integrated over the domain Ω.

$$\int_\Omega \frac{\partial}{\partial x}\left(\rho k_f \frac{\partial h_f}{\partial x}\right) + \frac{\partial}{\partial z}\left(\rho k_f \frac{\partial h_f}{\partial z} + \frac{\rho - \rho_f}{\rho_f}\right) d\Omega = -\int_\Omega \eta \rho q' d\Omega. \tag{3.145}$$

Applying Green's theorem (Equation (3.21)) to the second derivative and substituting ϕ for η results in

$$-\int_\Omega \nabla\phi_i \cdot (\rho k \nabla h_f) d\Omega + \int_\Omega \phi_i \frac{\partial}{\partial z}\left(\rho k \frac{\rho - \rho_f}{\rho_f}\right) + \int_\Gamma \phi_i \rho k \nabla \cdot h_f = -\int_\Omega \phi_i \rho q' d\Omega. \tag{3.146}$$

The integral over the boundary Γ disappears when the homogeneous Neumann boundary condition $\nabla h_f \cdot \mathbf{n} = 0$ is used. Substitution of

$$h_f = \sum_{j=1}^{n} h_{f_j} \phi_j(\mathbf{x}), \tag{3.147}$$

with

$$\mathbf{x} = \begin{bmatrix} x \\ z \end{bmatrix}, \tag{3.148}$$

results in

$$\begin{aligned} -\sum_{j=1}^{n} h_{f_j} \int_\Omega \rho k \nabla\phi_i \cdot \phi_j d\Omega = \\ = -\int_\Omega \phi_i \frac{\partial}{\partial z}\left(\rho k \frac{\rho - \rho_f}{\rho_f}\right) d\Omega - \sum_{j=1}^{n_b} h_{f_j} \int_\Gamma \phi_i g_2 - \int_\Omega \phi_i \rho q' d\Omega. \end{aligned} \tag{3.149}$$

Equation (3.149) is a system of equations of the form

$$T h_f = f,$$

with T the $nv*nl \times nv*nl$ stiffness matrix, h_f the vector with the freshwater heads and f the right hand side. The element matrix of the stiffness matrix is

$$T^{e_k} = \begin{bmatrix} T^{e_k}(1,1) & T^{e_k}(1,2) & T^{e_k}(1,3) \\ T^{e_k}(2,1) & T^{e_k}(2,2) & T^{e_k}(2,3) \\ T^{e_k}(3,1) & T^{e_k}(3,2) & T^{e_k}(3,3) \end{bmatrix}, \tag{3.150}$$

with

$$T^{e_k}(i,j) = -\frac{|\Delta|}{6} k(\nabla\phi_i \cdot \nabla\phi_j) \sum_{l=1}^{3} \rho(\mathbf{x}_l). \tag{3.151}$$

The element vector of f is

$$f^{e_k} = \begin{bmatrix} f^{e_k}(1) \\ f^{e_k}(2) \\ f^{e_k}(3) \end{bmatrix}, \tag{3.152}$$

with

$$f(i)^{e_k} = -\frac{|\Delta|}{6} \left(\frac{\partial}{\partial z} \left(\rho(\mathbf{x}_i) k \frac{\rho(\mathbf{x}_i) - \rho_f}{\rho_f} \right) + \rho(\mathbf{x}_i) q'(\mathbf{x}_i) \right). \tag{3.153}$$

The element vector for the boundary elements is

$$f^{e_l}(i) = -\frac{|\tilde{\Delta}|}{2} g_2(\mathbf{x}_i), \tag{3.154}$$

with $|\tilde{\Delta}|$ the length of the boundary element.

The term $\frac{\partial}{\partial z}\left(\rho(\mathbf{x}_i) k \frac{\rho(\mathbf{x}_i)-\rho_f}{\rho_f}\right)$ can be written as

$$\frac{\partial}{\partial z} \left(\rho(\mathbf{x}_i) k \frac{\rho(\mathbf{x}_i) - \rho_f}{\rho_f} \right) = \frac{\partial}{\partial z} \left(\frac{k}{\rho_f} \rho(\mathbf{x}_i)^2 - k\rho(\mathbf{x}_i) \right),$$

and is determined by

$$\frac{\partial}{\partial z} \left(\rho(\mathbf{x}_i) k \frac{\rho(\mathbf{x}_i) - \rho_f}{\rho_f} \right) \approx \left(\frac{2k}{\rho_f} - k \right) \sum_{l=1}^{3} \rho(\mathbf{x}_l) \frac{\partial \phi_l}{\partial z}. \tag{3.155}$$

The gradients of the basis functions can be found in the Equations (3.9), (3.10) and (3.11).

In order to calculate the velocities q_x and q_y Darcy's law (Equation (2.11)) is used. With k constant this results in

$$q_x = -k \frac{\partial h_f}{\partial x}, \tag{3.156}$$

$$q_z = -k \left(\frac{\partial h_f}{\partial z} + \frac{\rho - \rho_f}{\rho_f} \right). \tag{3.157}$$

The partial derivatives of h_f for a regular grid can be determined by central differences.

Chapter 4

Numerical experiments

4.1 2D Transport Equation

error(1): wrong syntax

4.1.1 2D advection equation

The transport equation

$$\begin{cases} -\nabla \cdot (\theta D \nabla C) + \mathbf{q} \cdot \nabla C + \theta \frac{\partial C}{\partial t} = q_{so} C_s, \\ \left((\theta D \nabla C) \cdot \mathbf{n}\right)|_{\Gamma_2} = g_2(\mathbf{x}), \\ \left(\sigma C + (\theta D \nabla C \cdot \mathbf{n})\right)|_{\Gamma_3} = g_3(\mathbf{x}), \qquad \sigma \geq 0, \\ C(\mathbf{x}, t_0) = C_0(\mathbf{x}), \end{cases} \tag{4.1}$$

is derived in Chapter 2 and numerically solved in two dimensions with the Finite Element Method as shown in Chapter 3. At the no-flow and outflow boundary a homogeneous Neumann boundary condition is used. At the inflow boundary a homogeneous Robbins condition is used with $\sigma = 0$.

For the two dimensional advection-dispersion equation numerical experiments are done for the Standard Galerkin Approach (SGA) (see Section 3.2.1), SUPG Classicial upwind method (see Section 3.2.2), SUPG pure advection algorithm (see Section 3.2.2) and Mizukami-Hughes algorithm (see Section 3.2.3).

The size of the domain Ω is one meter in the x-direction and one meter in the y-direction. This area is divided into regular triangles with 50 nodes in both directions, which results in a total of 2500 nodes. The elements and vertices in the grid of Figure 4.1 are numbered from left to right and from the bottom to the top. All angles of the 4802 elements of this structured grid are smaller than or equal to $\pi/2$ or 90 degrees.

The initial condition of Figure 4.2 can be seen as an injection of salt in some grid points at the initial state. A density ρ of 1025 kg/m^3 corresponds to salt water, $\rho = 1000$ kg/m^3 corresponds to freshwater. A density between those values is called brackish water.

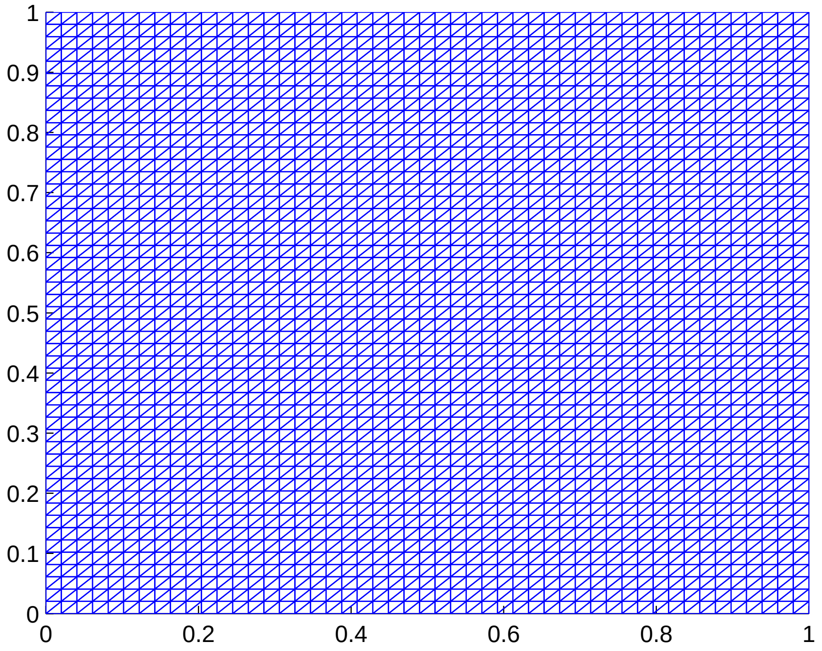


Figure 4.1: Structured grid, with 4802 elements and 2500 vertices, numbered from te left to the right and from the bottom to the top.

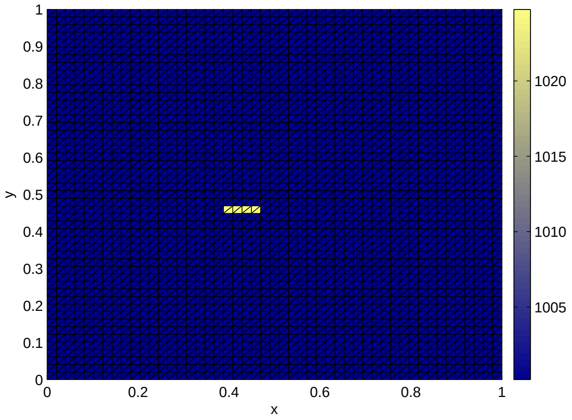


Figure 4.2: Initial condition of the density in the area $\Omega = 1 \times 1$ meter. The density $\rho = 1000\ kg/m^3$ corresponds to freshwater, $\rho = 1025\ kg/m^3$ corresponds to salt water.

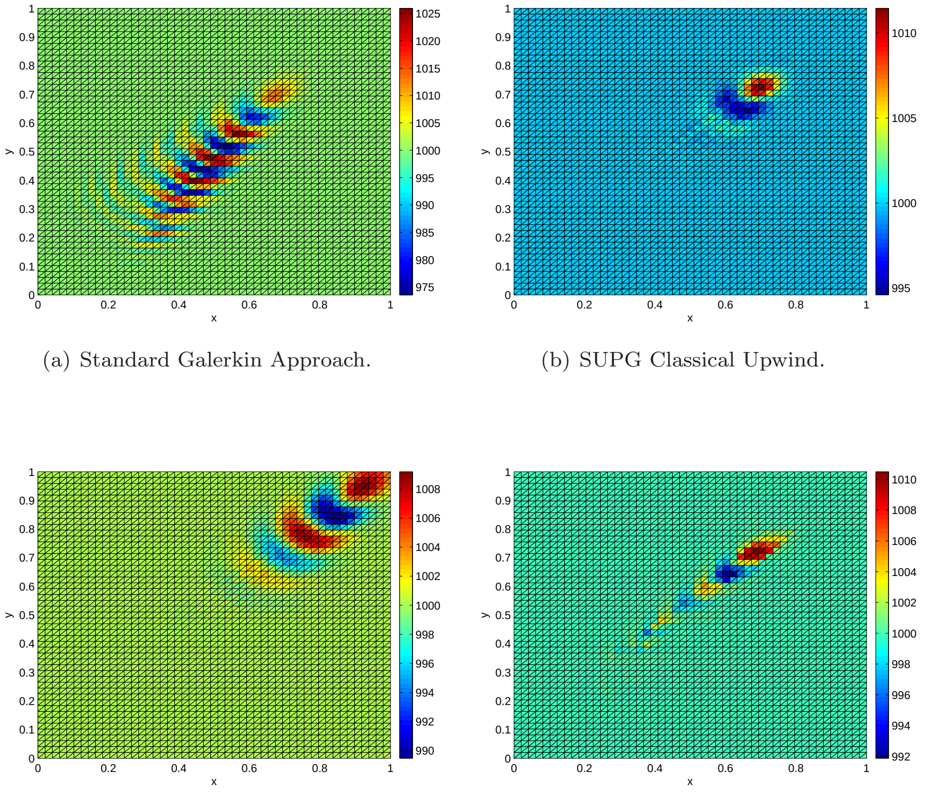
Different methods

For the parameters the following values are taken: dispersion coefficient $D = 0$, velocity in the x-direction $q_x = 0.1$, velocity in the y-direction $q_y = 0.1$, porosity $\theta = 1$, time step $\tau = 0.05$ and the number of time steps $T = 60$. So the salt concentration should move 0.3 meters into the $x-$ as well as the $y-$direction.

In Figure 4.3(a) the results for the Standard Galerkin Approach are shown. It can be seen that unwanted wiggles appear. The SUPG method Classical Upwind is shown in Figure 4.3(b), less wiggles appear with this method but this method has numerical dispersion. It is noted that the SUPG Classical Upwind method gives a solution with the same quality for negative velocities ($q_x = q_y = -0.1$).

In Figure 4.3(c) the algorithm for the pure advection equation is shown. This method shows more wiggles and more numerical dispersion than the SUPG Classical Upwind method. The Mizukami Hughes algorithm is shown in Figure 4.3(d). This Figure shows a maximum density ρ of 1010 kg/m^3, due to numerical dispersion. In the case of Figure 4.3(d), the flow chart of the Mizukami Hughes algorithm (see Figure 3.6 in Section 3.2.3) shows that the coefficients b_i are set $b_1 = 2/3$, $b_i = -1/3$ and $b_i = -1/3$ because the velocity $\mathbf{q}$ is in the direction of the vertex zone of node 1. In this case the choice of w_2 and w_3 does not influence the solution.

The maximum density ρ should remain 1025 kg/m^3, but increases with 12 kg/m^3 for the SUPG Classical Upwind method, with 16 kg/m^3 for the pure advection algorithm and with 14 kg/m^3 for the Mizukami Hughes algorithm. The covered distance after 60 time steps, $\tau = 0.05$ and with a velocity of $q_x = q_y = 0.1$ m/day should be 0.3 meter. Both the SUPG Classical Upwind and Mizukami Hughes algorithm have their maximum at the right location. For this example, the SUPG Classical Upwind method (Figure 4.3(a)) gives the best results.



(a) Standard Galerkin Approach.

(b) SUPG Classical Upwind.

(c) SUPG Advection only algorithm.

(d) Mizukami Hughes algorithm.

Figure 4.3: Advection equation with $q_x = q_y = 0.1$, $D = 0$, $\theta = 1$, $\tau = 0.05$, $T = 60$.

Timestep

The CFL (Courant-Friedrich-Levy) condition for the advection and advection-dispersion equation is

$$|\frac{q_x\tau}{\theta\Delta x}| + |\frac{q_y\tau}{\theta\Delta y}| \leq 1, \tag{4.2}$$

for two dimensional problems. The parameter for the porosity $\theta = 1$ in these examples.

Different element distances are shown in Figure 4.4. Note that $\Delta x_{max} = 0.02$ and $\Delta y_{max} = 0.02$ in the grid of Figure 4.1, so when $q_x = 0.1$ and $q_y = 0.1$, the CFL condition results in $\tau \leq 0.1$. But Δx and Δy are the distances in a triangle and are actually $0 \leq \Delta x \leq 0.02$ and $0 \leq \Delta y \leq 0.02$ (see Figure 4.4). When $\Delta x = \Delta y = 0.02$ is taken, the CFL condition results in a time step of $\tau \leq 0.1$.

Figure 4.5(a) shows that the time step $\tau = 0.1$ is too large for the advection equation, this can be seen by the unwanted wiggles in the density.

In Figure 4.5(b) the representative distances are taken $\Delta x = 0.002$ and $\Delta y = 0.002$, so a tenth of the maximum distances of the element, which results in $\tau \leq 0.01$. These results are better, the solution has less wiggles.

Figure 4.5(c) shows that taking $\Delta x = 0.01$ and $\Delta y = 0.01$, so the half of the maximum distances of the element, satisfies. This corresponds to a time step $\tau = 0.05$. In all two dimensional numerical experiments with $q_x = q_y = 0.1$, $\tau = 0.05$ is used in this section.

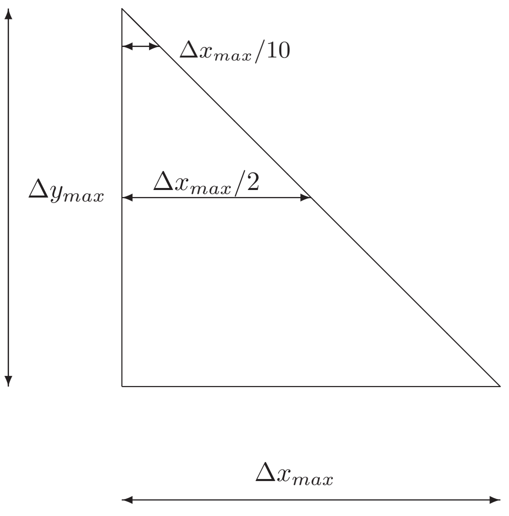
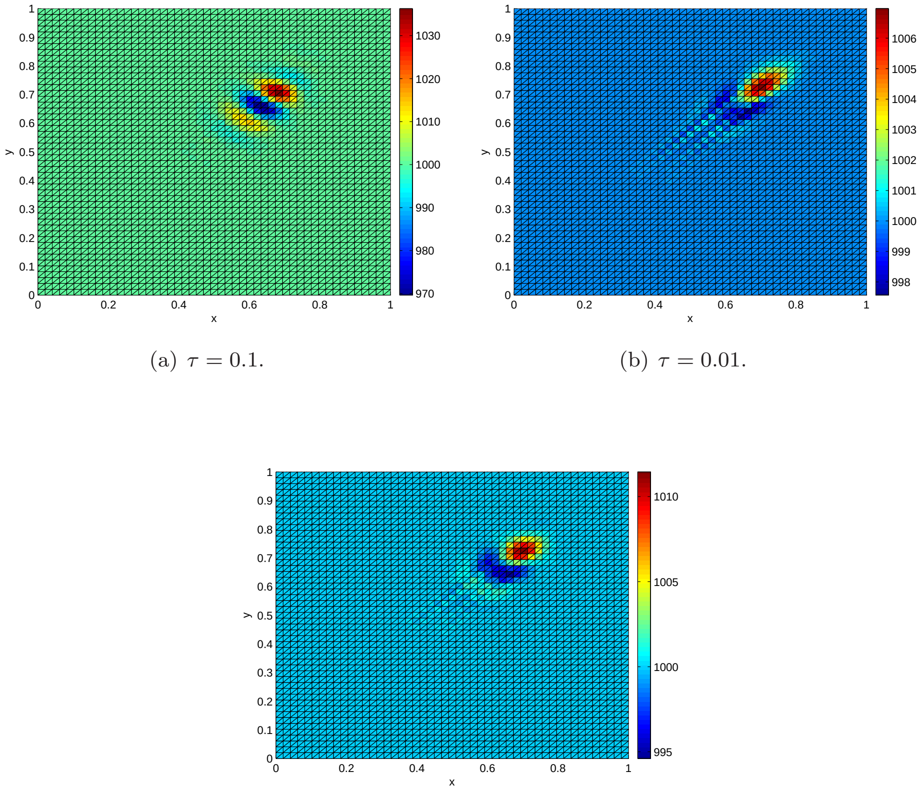


Figure 4.4: Element of the Finite Element grid, with the element distances $\Delta x_{max} = \Delta y_{max} = 0.02$, $\Delta x_{max}/2 = \Delta y_{max}/2 = 0.01$ and $\Delta x_{max}/10 = \Delta y_{max}/10 = 0.002$.



(a) $\tau = 0.1$.

(b) $\tau = 0.01$.

(c) $\tau = 0.05$.

Figure 4.5: Advection equation solved with SUPG Classical Upwind. $q_x = q_y = 0.1$, $D = 0$, $\theta = 1$, $T = 30$. In Figure 4.5(a) the time step is $\tau = \mathbf{0.1}$, in Figure 4.5(b) $\tau = \mathbf{0.01}$ and in Figure 4.5(c) $\tau = \mathbf{0.05}$.

Porosity

The porosity θ has an influences the seepage velocity. The seepage velocity $\mathbf{v}$ is defined as

$$\mathbf{v} = \frac{\mathbf{q}}{\theta},$$

hence for $\theta = 1$ the seepage velocity equals the Darcy velocity $\mathbf{q}$. When the porosity $\theta = 0.2$, the seepage velocity increases five times.

A porosity $\theta = 1$ means that the total volume is available for fluid transmission. The value $\theta = 0$ results in the differential equation

$$\mathbf{q} \cdot \nabla C = q_{so} C_s. \tag{4.3}$$

Hence when there is no source, the velocity $\mathbf{q} = 0$. This can physically be explained as no possibility for a fluid to flow in a solid material. The source term $q_{so}C_s$ should be dependent on the porosity, because no salt can be injected in a solid with no volume available for fluid transmission.

When $\theta = 0.5$, the seepage velocity increases two times, as can be seen in Figure 4.6, because there is less space for the fluid to flow. The covered distance has now increased from 0.3 meter (see Figure 4.3(b)) to about 0.6 meter.

Note that when the porosity changes, the stability condition for the time step changes. When $\theta = 0.5$ and $\Delta x_{max}/2 = \Delta x_{max}/2 = 0.01$, the stability condition for the time step becomes $\tau \leq 0.025$. The example of Figure 4.6 does not satisfy this condition, which explains the wiggles. Unless the wiggles, still can be seen that the covered distance doubles when the porosity $\theta = 0.5$ instead of $\theta = 1$.

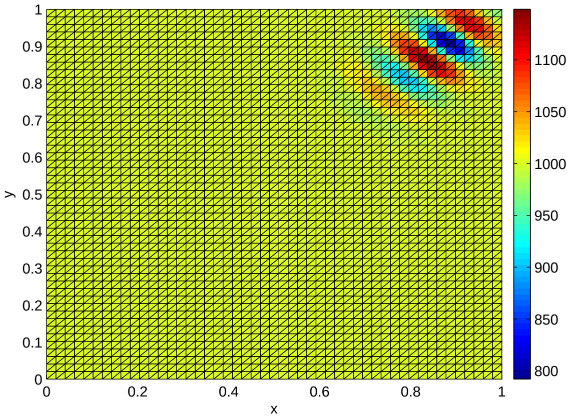


Figure 4.6: SUPG Classical Upwind. $q_x = q_y = 0.1$, $D = 0$, $\theta = \mathbf{0.5}$, $\tau = 0.05$, $T = 60$.

Sharp fresh-salt front

In Figure 4.7 another initial condition is given for the advection equation. This Initial Condition can be seen as a sharp fresh-salt-fresh front. The values of the parameters are $D = 0$, $q_x = 0$, $q_y = 0.1$, $\theta = 1$, $\tau = 0.05$.

Results are shown after 60 time steps ($T = 60$) for the SUPG Classical Upwind method (Figure 4.8(a)) and the Mizukami Hughes algorithm (Figure 4.8(b)) and after 120 time steps (Figure 4.9). At the boundaries $x = 0$ and $x = 1$ the Mizukami Hughes algorithm shows a density with less wiggles than the SUPG Classical Upwind method. The minimum and maximum densities of the SUPG Classical Upwind method in Figure 4.9(a) are $\rho = 970$ and $\rho = 1030$ kg/m^3. For the Mizukami Hughes algorithm, these values are $\rho = 975$ and $\rho = 1025$ kg/m^3. It can be concluded that in this example the Mizukami Hughes algorithm is better than the SUPG Classicial Upwind method.

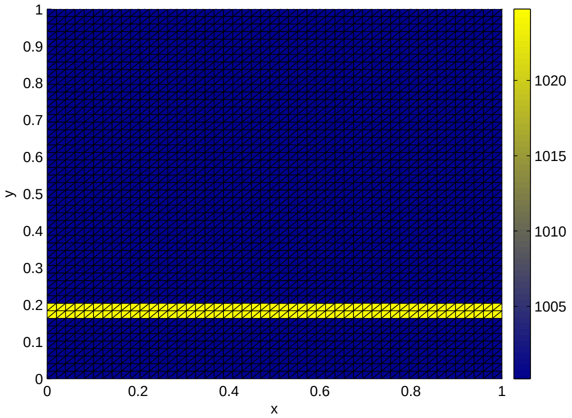
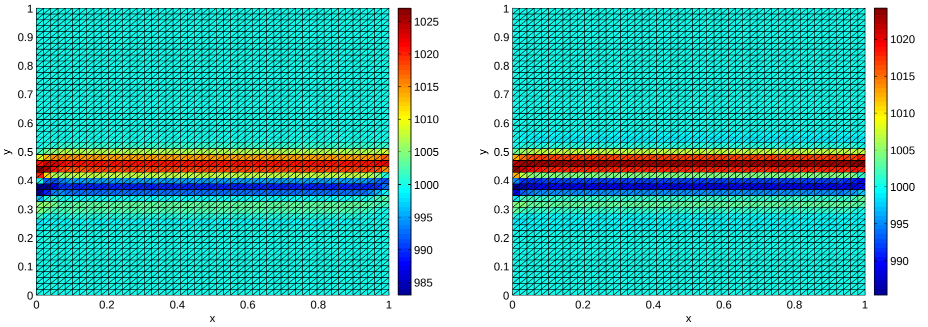


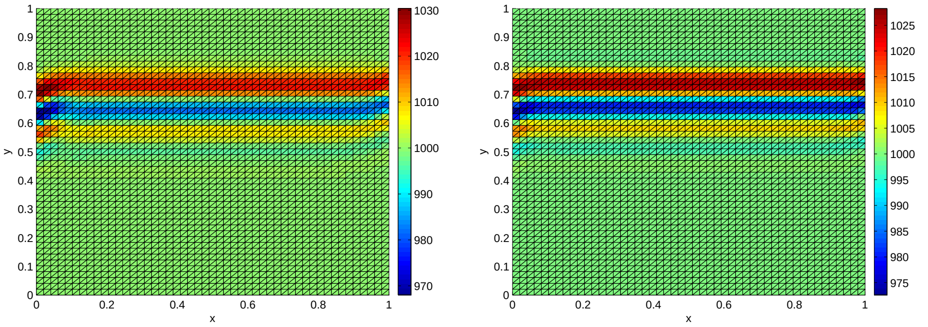
Figure 4.7: Initial condition of a sharp fresh-salt-fresh front. The density $\rho = 1000$ kg/m^3 corresponds to freshwater, $\rho = 1025$ kg/m^3 to salt water.



(a) SUPG Classical Upwind.

(b) Mizukami Hughes algorithm.

Figure 4.8: Advection equation, contourplot of the density ρ of the water with $q_x = 0$, $q_y = 0.1$, $D = 0$, $\theta = 1$, $\tau = 0.05$, $\mathbf{T = 60}$. In Figure 4.8(a) the SUPG Classical Upwind is shown and in Figure 4.8(b) the Mizukami Hughes algorithm.



(a) SUPG Classical Upwind.

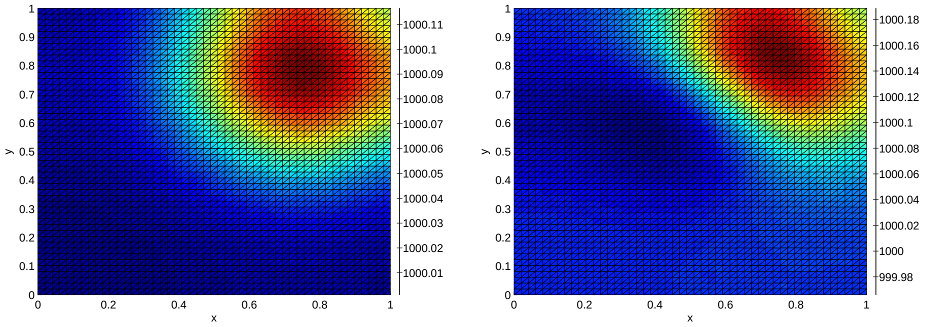
(b) Mizukami Hughes algorithm.

Figure 4.9: Advection equation, contourplot of the density ρ of the water with $q_x = 0$, $q_y = 0.1$, $D = 0$, $\theta = 1$, $\tau = 0.05$, $\mathbf{T = 120}$. In Figure 4.9(a) the SUPG Classical Upwind is shown and in Figure 4.9(b) the Mizukami Hughes algorithm.

4.1.2 2D advection-dispersion equation

Different dispersion coefficients

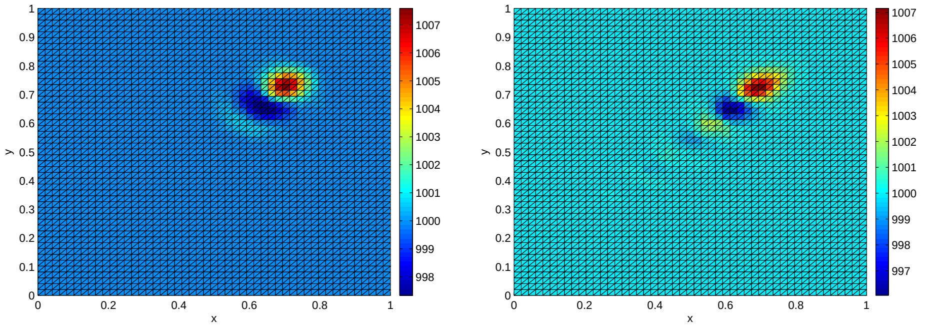
In general, in groundwater flow the order of dispersion is $3 * 10^{-3}$ m^2/day and the order of advection is $3 * 10^{-2}$ m/day. Hence the dispersion is ten times smaller than the advection. In the Figures 4.10(a) and 4.10(b) these values for the dispersion coefficient D and the velocities q_x and q_y are taken. In the Figures 4.11(a) and 4.11(b) the SUPG Classical Upwind and the Mizukami Hughes algorithm for $q_x = q_y = 0.1$ and $D = 10^{-4}$ are shown. In the Figures 4.12(a) and 4.12(b) these contour lines of the density ρ are shown for $D = 10^{-8}$. The SUPG Classical Upwind method shows the best results for small dispersion coefficients in this example.



(a) SUPG Classical Upwind.

(b) Mizukami Hughes algorithm.

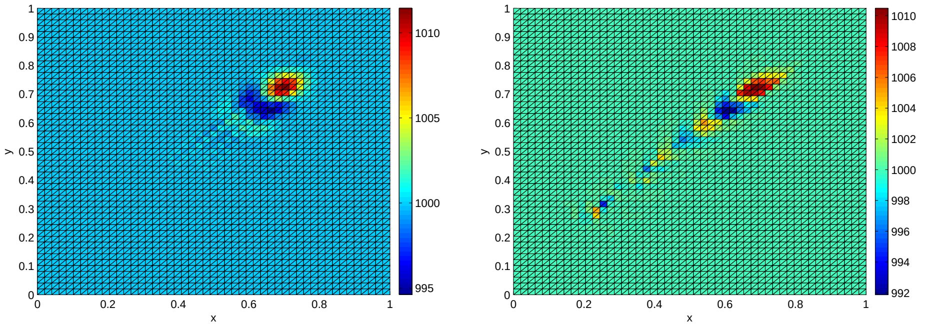
Figure 4.10: Advection-dispersion equation with $q_x = q_y = 0.1$, $\mathbf{D = 10^{-2}}$, $\theta = 1$, $\tau = 0.05$, $T = 60$. In Figure 4.10(a) the SUPG Classical Upwind is shown and in Figure 4.10(b) the Mizukami Hughes algorithm.



(a) SUPG Classical Upwind.

(b) Mizukami Hughes algorithm.

Figure 4.11: Advection-dispersion equation with $q_x = q_y = 0.1$, $\mathbf{D = 10^{-4}}$, $\theta = 1$, $\tau = 0.05$, $T = 60$. In Figure 4.11(a) the SUPG Classical Upwind is shown and in Figure 4.11(b) the Mizukami Hughes algorithm.



(a) SUPG Classical Upwind.

(b) Mizukami Hughes algorithm.

Figure 4.12: Advection-dispersion equation with $q_x = q_y = 0.1$, $\mathbf{D = 10^{-8}}$, $\theta = 1$, $\tau = 0.05$, $T = 60$. In Figure 4.12(a) the SUPG Classical Upwind is shown and in Figure 4.12(b) the Mizukami Hughes algorithm.

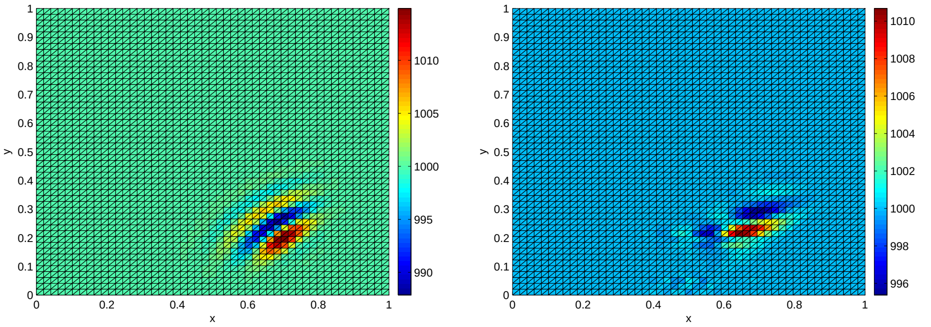
Direction of the velocity

In Figure 4.13(b) an example is given of the Mizukami Hughes algorithm with $q_x = 0.1$, $q_y = -0.1$ and $D = 10^{-4}$. For most of the elements, the velocity is now in the direction of the edge zone of node 1. See Section 3.2.3 for more information and Figure 4.11 for an example where the velocity is in the direction of the vertex zone of node 1. In Figure 4.13(a) the SUPG Classical Upwind method can be found for the same parameters. The Mizukami Hughes algorithm is better in this case. A possible cause for the wiggles in the SUPG classical upwind method is the chosen representative element distance in the direction of $\mathbf{q}$, see Appendix D for more details.

In Figure 4.14 the velocity $q_x = 0.1$ and $q_y = 0$, for half the elements this results in a velocity in the direction of the edge zone of node 1 in the Mizukami Hughes algorithm. For the other half of the elements the velocity is in the direction of the vertex zone of node 1. Figure 4.15 shows the SUPG Classical Upwind method and the Mizukami Hughes algorithm for $q_x = -0.1$ and $q_y = -0.1$, which results in a velocity in the direction of the vertex zone of node 1 for all elements.

It seems that the Mizukami Hughes algorithm is better than the SUPG Classical Upwind method only when at least for half of the elements the velocity will be in the direction of the edge zone of node 1 of the element. It is noted that computations are much more time consuming for the Mizukami Hughes algorithm than for the SUPG Classical Upwind method, because the element matrices have to be recalculated every time step. This is necessary because the coefficients of the upwind part of the Mizukami Hughes algorithm depend on the gradients of the concentration C.

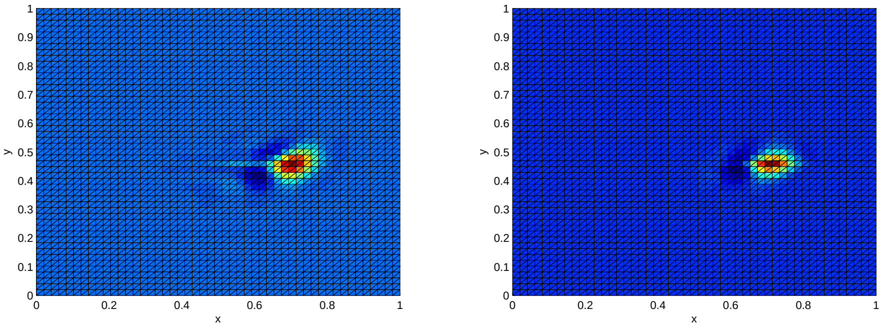
It seems that the SUPG classical upwind method with the Segal algorithm defined in Section 3.2.2 works well in most cases. The parameter h is the representative distance of the element in the direction of the velocity $\mathbf{q}$ calculated by the Segal algorithm. When q_x is positive and q_y is positive or when q_x is negative and q_y is negative, this approach works well. When q_x is positive and q_y negative or vice versa, another representative distance in the element is calculated. The algorithm developed by Segal used for the calculation of this representative distance takes the absolute value of the velocity $\mathbf{q}$ and the distance in that direction may be different than the distance in the direction q_x negative and q_y positive. See Appendix D for more details.



(a) SUPG Classical Upwind.

(b) Mizukami Hughes algorithm.

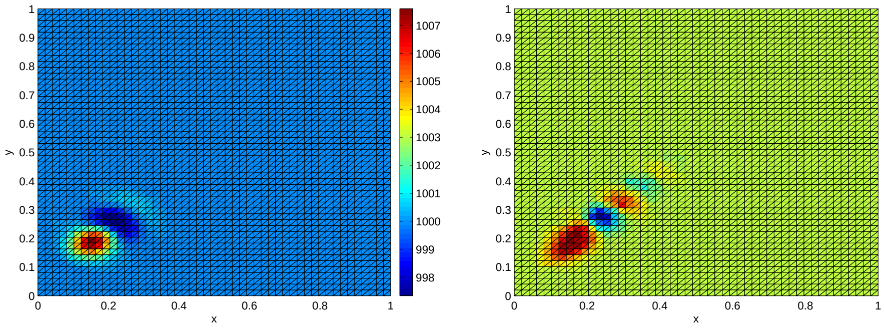
Figure 4.13: Advection-dispersion equation with $\mathbf{q_x} = \mathbf{0.1}$, $\mathbf{q_y} = -\mathbf{0.1}$, $D = 10^{-4}$, $\theta = 1$, $\tau = 0.05$, $T = 60$. In Figure 4.13(a) the SUPG Classical Upwind is shown and in Figure 4.13(b) the Mizukami Hughes algorithm.



(a) SUPG Classical Upwind.

(b) Mizukami Hughes algorithm.

Figure 4.14: Advection-dispersion equation with $\mathbf{q_x} = \mathbf{0.1}$, $\mathbf{q_y} = \mathbf{0}$, $D = 10^{-4}$, $\theta = 1$, $\tau = 0.05$, $T = 60$. In Figure 4.14(a) the SUPG Classical Upwind is shown and in Figure 4.14(b) the Mizukami Hughes algorithm.



(a) SUPG Classical Upwind.

(b) Mizukami Hughes algorithm.

Figure 4.15: Advection-dispersion equation with $\mathbf{q_x} = \mathbf{-0.1}$, $\mathbf{q_y} = \mathbf{-0.1}$, $D = 10^{-4}$, $\theta = 1$, $\tau = 0.05$, $T = 60$. In Figure 4.15(a) the SUPG Classical Upwind is shown and in Figure 4.15(b) the Mizukami Hughes algorithm.

4.1.3 2D diffusion equation

When the groundwater does not move ($\mathbf{q} = 0$) the dispersion coefficient or matrix D reduces to a diffusion coefficient or matrix. In the Figures 4.16 and 4.17 results are shown for the diffusion equation. As initial condition, Figure 4.2 is used. The parameters are chosen $q_x = q_y = 0$, $\theta = 1$, $\tau = 0.05$ and $T = 60$. As diffusion coefficient $D = 10^{-4}$ is taken in Figure 4.16 and $D = 10^{-2}$ is taken in Figure 4.17.

It can be seen that $D = 10^{-4}$ (Figure 4.16) has a large influence on the value of the density ρ. The initial density was $\rho = 1025$ kg/m^3 (Figure 4.2), in Figure 4.16 this value has decreased to $\rho = 1008$. On the other hand, the area affected by this diffusion coefficient is small. The maximum density in Figure 4.17 is $\rho = 1000.11$ kg/m^3, which is almost freshwater.

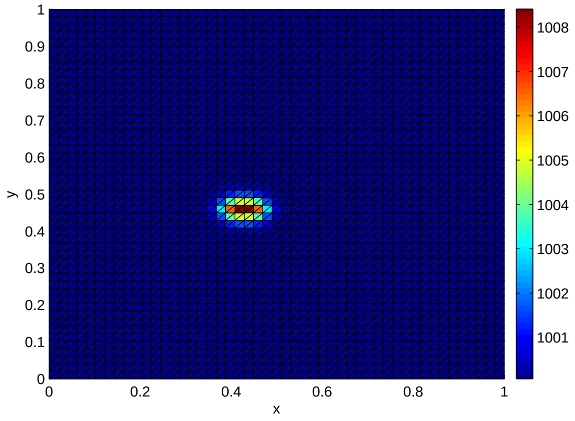


Figure 4.16: Standard Galerkin Approach. $q_x = q_y = 0$, $\mathbf{D = 10^{-4}}$, $\theta = 1$, $\tau = 0.05$, $T = 60$.

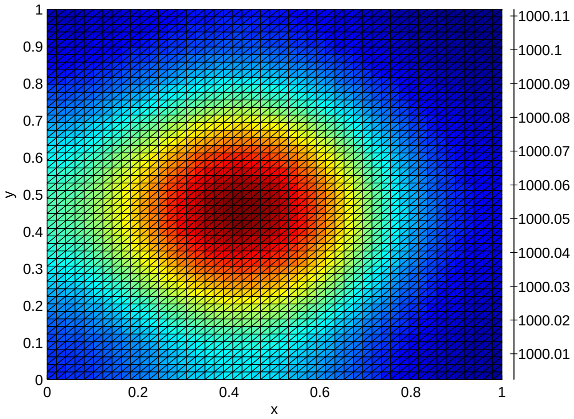
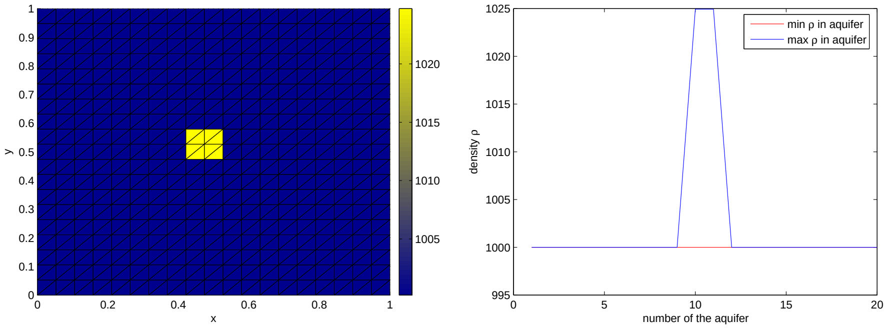


Figure 4.17: Standard Galerkin Approach. $q_x = q_y = 0$, $\mathbf{D} = \mathbf{10^{-2}}$, $\theta = 1$, $\tau = 0.05$, $T = 60$.

4.2 3D transport equation

In Figure 4.18 an initial condition for the three dimensional numerical experiments is shown. A domain Ω of $1 \times 1 \times 1$ meter ($1\ m^3$) is chosen, subdivided into 20 aquifers (hence 20 grid point in the vertical direction). Each aquifer has 400 nodes, so the total number of nodes in the domain Ω is 8000. In the horizontal direction (2D), the transport equation is solved with the FEM SUPG Classical Upwind. In the vertical direction (1D), the transport equation is solved with the FDM Upwind.

Figure 4.19(a) shows the contour lines of the density ρ in the aquifers 10 and 11. There is an initial salt concentration in the aquifers 10 and 11. Figure 4.18(b) shows the minimum and maximum density per aquifer. On the x-axes the number of the aquifer is shown and on the y-axes the density. This figure shows that the minimum as well as the maximum density in aquifer 1 is 1000 kg/m^3, hence freshwater. It can be seen that the initial density in all other aquifers besides aquifer 10 and 11 is 1000 kg/m^3.



(a) Density ρ in aquifer 10 and 11.

(b) Minimum and maximum density ρ per aquifer. On the x-axes the numer of the aquifer is shown and on the y-axes the density. The blue line shows the maximum density ρ in the corresponding aquifer, the red line the minimum density ρ in the corresponding aquifer.

Figure 4.18: Initial condition. The domain Ω is subdivide into 20 layers (aquifers) with 400 nodes in each layer. $\rho = 1000\ kg/m^3$ corresponds to freshwater, $\rho = 1025\ kg/m^3$ in the aquifers 10 and 11 to salt water.

4.2.1 3D advection equation

The CFL number in two dimensions is given in Equation (4.2). From the CFL condition it follows that for the grid as shown in Figure 4.18, the time step should be taken $\tau \leq 0.125$ when $q_x = 0.1$ and $q_y = 0.1$ and $\Delta x = \Delta y = 0.025$. The CFL condition for the z-direction is

$$|\frac{q_z \tau}{\theta \Delta z}| \leq 1, \tag{4.4}$$

which results in $\tau \leq 0.5$ for $q_z = 0.1$ and $\Delta z = 0.05$.

The most severe CFL condition for three dimensions is

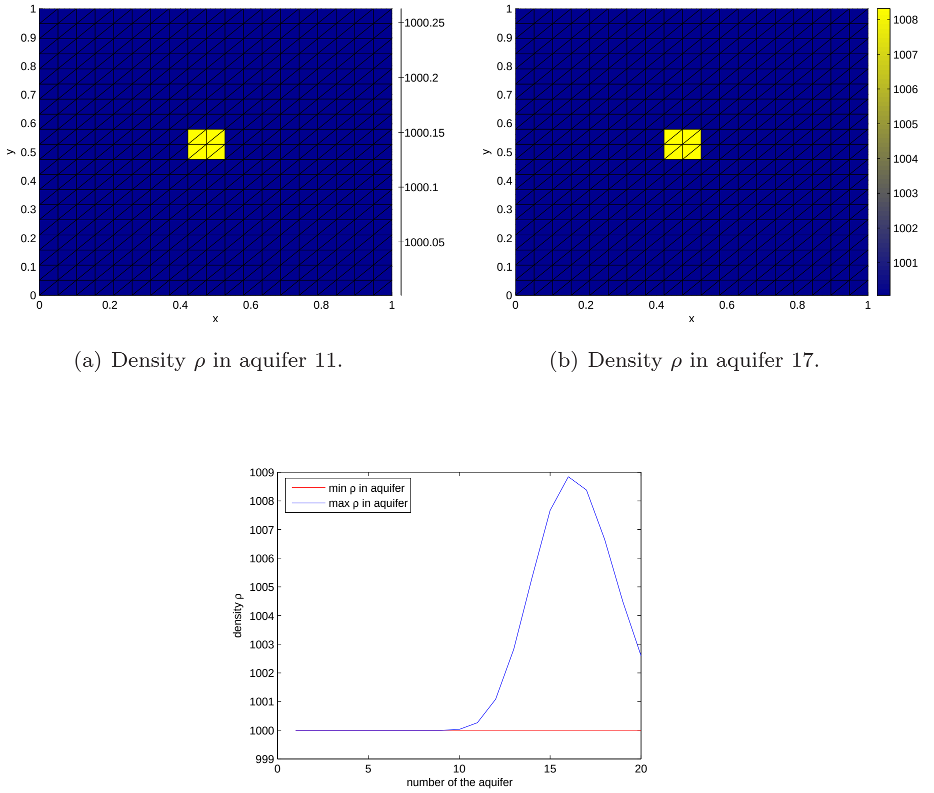
$$|\frac{q_x\tau}{\theta\Delta x}| + |\frac{q_y\tau}{\theta\Delta y}| + |\frac{q_z\tau}{\theta\Delta z}| \leq 1. \tag{4.5}$$

For $q_x = q_y = q_z = 0.1$ and $\Delta x = \Delta y = 0.025$ and $\Delta z = 0.05$ this results in $\tau \leq 0.1$. This condition indeed results in the smallest time step.

In the three dimensional numerical experiments τ is chosen $\tau = 0.05$ whenever $|q_x| \neq 0$ or $|q_y| \neq 0$ and $\tau = 0.1$ whenever $q_x = q_y = 0$ in order to obtain smoother solutions.

The advection equation with $q_x = q_y = 0.1$ and $q_z = 0$ with another initial condition is shown in the Figures in Section 4.1.1. The advection equation with $q_x = q_y = 0$ and $q_z = 0.1$ is shown in Figure 4.19. The velocity $q_z = 0.1$, the time step $\tau = 0.1$ and the number of time steps $T = 30$, hence the covered distance in the z-direction should be 0.3 meter. In Figure 4.19(c) can be seen that the maximum density is now in the aquifers 16 and 17. Each aquifer is 0.05 meters hence indeed the covered distance is 0.3 meter. In the Figures 4.19(a) and 4.19(b) the contour lines of the density in aquifer 11 and 17 is shown. The maximum density ρ is 1000.25 kg/m^3 in aquifer 11 (Figure 4.19(a)) and 1008 kg/m^3 in aquifer 17, these values can also be found in Figure 4.19(c).

In Figure 4.20(a) can be seen that the salt concentration has moved 0.3 meter in the x- and y-direction. Figure 4.20(b) shows that the salt concentration has moved to aquifer 16, which is indeed a distance 0.3 meter from aquifer 10.

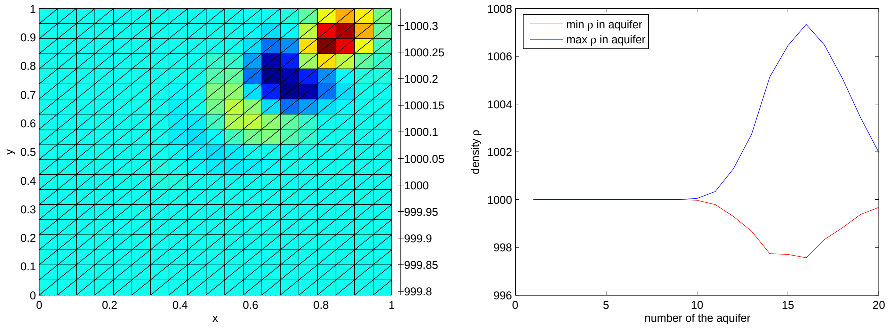


(a) Density ρ in aquifer 11.

(b) Density ρ in aquifer 17.

(c) Minimum and maximum density ρ per aquifer.

Figure 4.19: Advection equation with $\mathbf{q_x} = \mathbf{q_y} = \mathbf{0}$, $\mathbf{q_z} = \mathbf{0.1}$, $D = 0$, $\theta = 1$, $\tau = 0.1$, $T = 30$.



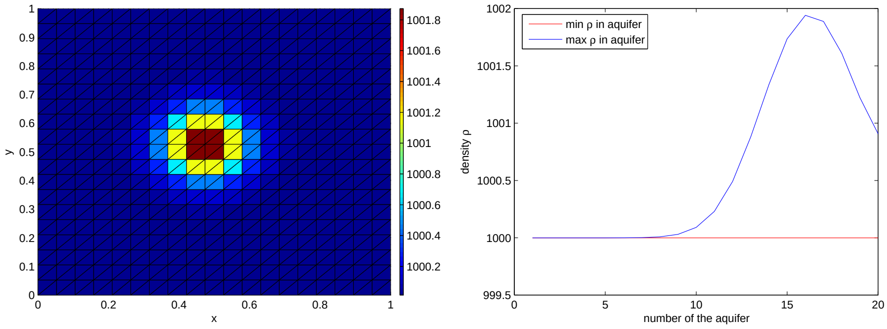
(a) Density ρ in aquifer 11.

(b) Minimum and maximum density ρ per aquifer.

Figure 4.20: Advection equation with $\mathbf{q_x} = \mathbf{q_y} = \mathbf{q_z} = \mathbf{0.1}$, $D = 0$, $\theta = 1$, $\tau = 0.05$, $T = 60$.

4.2.2 3D advection-dispersion equation

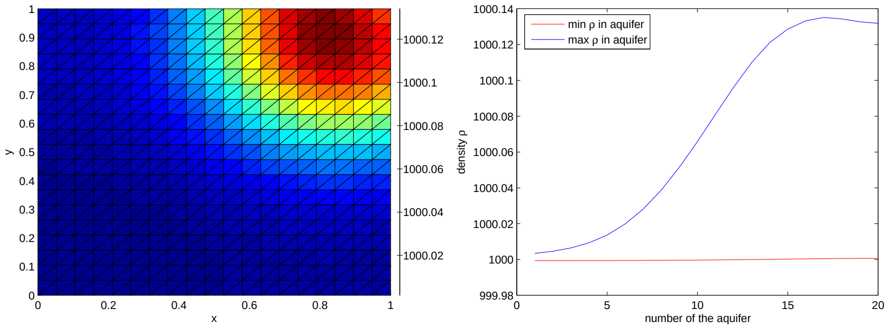
Figure 4.21 shows the advection-dispersion equation with $q_x = q_y = 0$, $q_z = 0.1$ and $D = 10^{-3}$. In Figure 4.21(a) the density ρ in aquifer 17 is shown, in Figure 4.21(b) the minimum and maximum density in each aquifer. In Figure 4.22 these figures are shown for $q_x = q_y = q_z = 0.1$ and $D = 10^{-2}$ and in Figure 4.23 with $q_x = q_y = q_z = 0.1$ and $D = 10^{-3}$.



(a) Density ρ in aquifer 17.

(b) Minimum and maximum density ρ per aquifer.

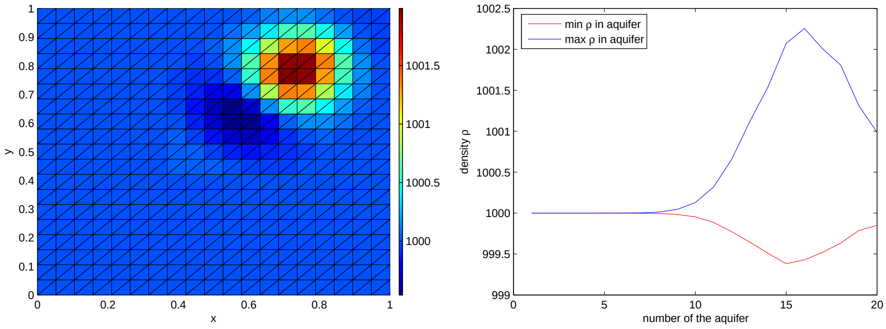
Figure 4.21: Advection-dispersion equation with $\mathbf{q_x = q_y = 0}$, $\mathbf{q_z = 0.1}$, $\mathbf{D = 0.001}$, $\theta = 1$, $\tau = 0.1$, $T = 30$.



(a) Density ρ in aquifer 17.

(b) Minimum and maximum density ρ per aquifer.

Figure 4.22: Advection-dispersion equation with $\mathbf{q_x = q_y = q_z = 0.1}$, $\mathbf{D = 0.01}$, $\theta = 1$, $\tau = 0.05$, $T = 60$.



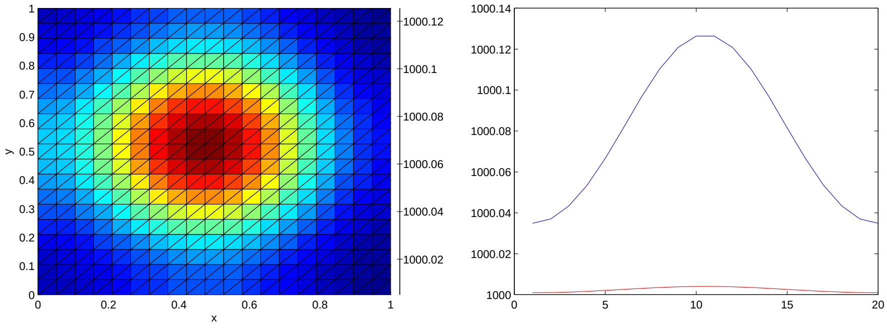
(a) Density ρ in aquifer 17.

(b) Minimum and maximum density ρ per aquifer.

Figure 4.23: Advection-dispersion equation with $\mathbf{q_x} = \mathbf{q_y} = \mathbf{q_z} = \mathbf{0.1}$, $\mathbf{D} = \mathbf{0.001}$, $\theta = 1$, $\tau = 0.05$, $T = 60$.

4.2.3 3D diffusion equation

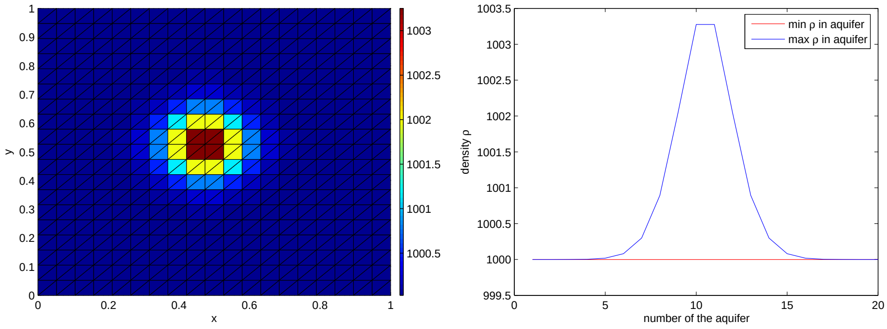
In the Figures 4.24 and 4.25 the solution of the three dimensional diffusion equation is shown. From these figures, it can be seen that the diffusion is the same in all three directions.



(a) Density ρ in aquifer 11.

(b) Minimum and maximum density ρ per aquifer.

Figure 4.24: Diffusion equation with $q_x = q_y = q_z = 0$, $\mathbf{D = 0.01}$, $\theta = 1$, $\tau = 0.1$, $T = 30$.



(a) Density ρ in aquifer 11.

(b) Minimum and maximum density ρ per aquifer.

Figure 4.25: Diffusion equation with $q_x = q_y = q_z = 0$, $\mathbf{D = 0.001}$, $\theta = 1$, $\tau = 0.1$, $T = 30$.

4.3 Density dependent flow: rotating brackish zone

An area with impermeable boundaries is considered with different densities. The benchmark problem rotating brackish zone (Figure 4.26) shows a clear coupling between the density differences in the area and the velocities of the water. Without sources of sinks, a flow appears in the water. This flow is only due to the density differences.

At the initial state, there are three zones: a saltwater, brackish water and freshwater zone with densities of $\rho = 1025$, $\rho = 1012.5$ and $\rho = 1000$ kg/m^3 as shown in 4.26 (a). Initially, at time $t = 0$, both interfaces are straight and make a 45^0 angle with the horizontal. Consider a two-dimensional, confined flow in a vertical cross-section. The aquifer is 40 meter thick and a 300 meter long section of the aquifer is considered with all boundaries impermeable. The hydraulic conductivity $k = 2$ m/d and the effective porosity $\theta = 0.2$. There is no diffusion or dispersion ($D = 0$). The brackish zone will rotate to a horizontal position through time, the results after $t = 2000$ days are shown in Figure 4.26 (b) [29, 7].

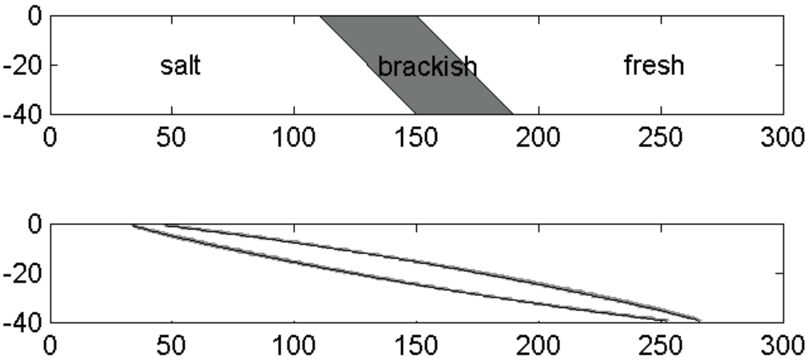


Figure 4.26: Experimental problem: a rotating brackish zone. a.) Setup of the example, b.) Boundary of brackish zone after 2000 days [7].

4.3.1 Cycles with Matlab

The flow equation is solved numerically using the two dimensional Standard Galerkin Approach in Section 3.7 and coupled to the two dimensional transport equation, which is solved with the SUPG method in Section 3.2.2. The coupled system of the flow equation and transport equation can be found in Figure 2.4. Both equations are solved with the aid of the numerical computing environment and programming language Matlab. The transport equation and flow equation are solved two dimensional in the x-/z-domain.

The boundaries of the area are impermeable. For the transport equation this means that the gradient of the concentration is zero at all boundaries, hence a homogeneous Neumann boundary condition is taken for all boundaries. For the flow equation the Darcy velocity $\mathbf{q}$ normal to all boundaries should be zero. Recall Darcy's law:

$$q_x = -k\frac{\partial h_f}{\partial x}, \tag{4.6}$$

$$q_z = -k\left(\frac{\partial h_f}{\partial z} + \frac{\rho - \rho_f}{\rho_f}\right). \tag{4.7}$$

At the left and the right boundary, the boundary condition is $q_x = 0$ which results in

$$\frac{\partial h_f}{\partial x} = 0, \tag{4.8}$$

the homogeneous Neumann boundary condition. At the bottom and the top of the area the boundary condition is $q_z = 0$. This results in

$$-k\left(\frac{\partial h_f}{\partial z} + \frac{\rho - \rho_f}{\rho_f}\right) = 0. \tag{4.9}$$

The Neumann boundary condition for the flow equation is defined as

$$\rho k \nabla h_f \cdot \mathbf{n} = g_2(\mathbf{x}). \tag{4.10}$$

Substitution of Equation (4.9) and Equation (4.10) and the boundary condition $q_z = 0$ results into

$$g_2(\mathbf{x}) = -\rho k\frac{\rho - \rho_f}{\rho_f}. \tag{4.11}$$

at the top and the bottom of the area. Note that $\frac{\rho-\rho_f}{\rho_f} = 0$ at the part with freshwater.

Initial state

The regular grid has 51 nodes in the x-direction and 41 nodes in the z-direction (total number of 2091 nodes). The total number of elements is 4000, the element width in the x-direction is 6 meter and in the z-direction 1 meter. The shapes of the elements are comparable to the shapes of the elements of the grid of Figure 4.1. The brackish zone with the initial condition of the density is shown in Figure 4.27. The left part (red part in Figure 4.27) of the water is salt water and has density $\rho = 1025\ kg/m^3$, the green part is brackish water with $\rho = 1012.5\ kg/m^3$ and the blue part of the area is freshwater ($\rho = 1000\ kg/m^3$).

Figure 4.28 shows the contour lines of the Darcy velocity q_x at the initial state calculated by the flow equation in Matlab. Note that the Darcy velocity has to be divided by the porosity ($\theta = 0.2$) in order to obtain the seepage velocity. The maximum Darcy velocity in the x-direction q_x is 0.04 m/day and is pointed to the right at the right side of the brackish zone and to the left at the left side of the brackish zone. The Darcy velocity at the z-direction q_z at the initial state has a maximum of 0.025 m/day and is upwards at the right side of the brackish zone and downwards at the left side. This can be see in Figure 4.29.

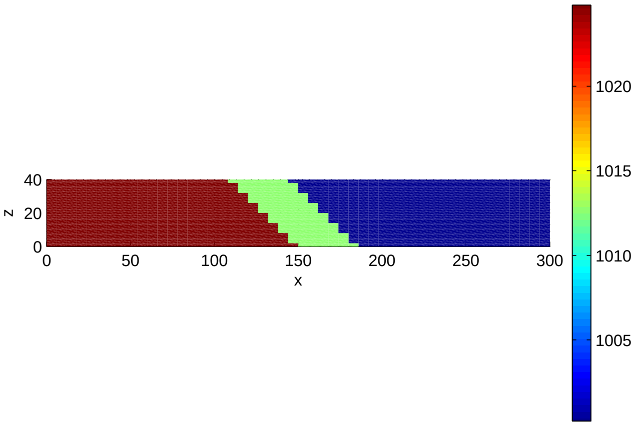


Figure 4.27: Initial condition rotating brackish zone: salt ($\rho = 1025$) - brackish ($\rho = 1012.5$) - fresh ($\rho = 1000$) front.

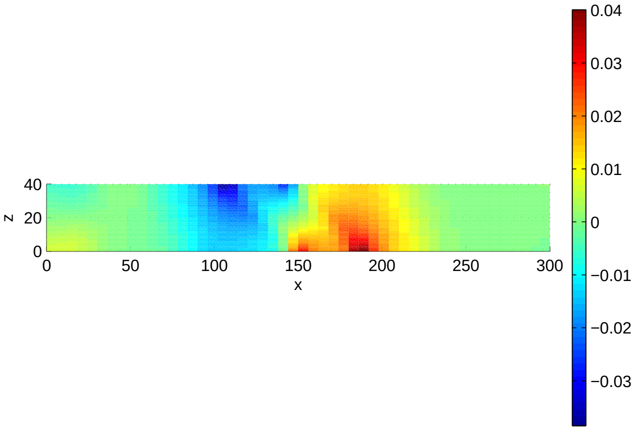


Figure 4.28: Initial state rotating brackish zone: Darcy velocity q_x.

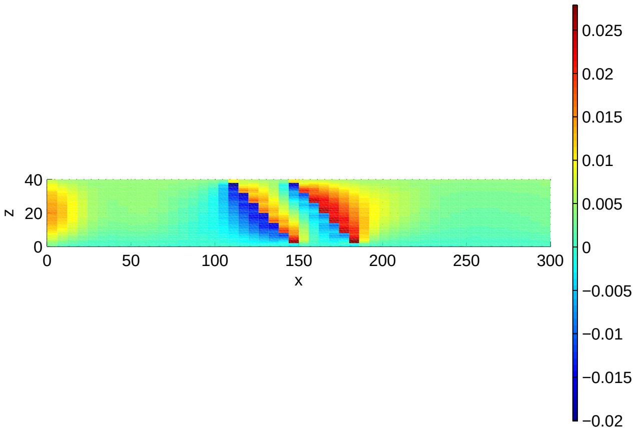


Figure 4.29: Initial state rotating brackish zone: Darcy velocity q_z.

Time step

The CFL condition is

$$|\frac{q_x\tau}{\theta\Delta x}| + |\frac{q_z\tau}{\theta\Delta z}| \leq 1.$$

For Δx and Δz half the maximum of the element width is taken, hence $\Delta x = 3$ and $\Delta z = 0.5$. In Figure 4.28 can be seen that the maximum Darcy velocity $q_x = 0.04$ and the maximum Darcy velocity $q_z = 0.03$. With $\theta = 0.2$, this leads to the stability condition

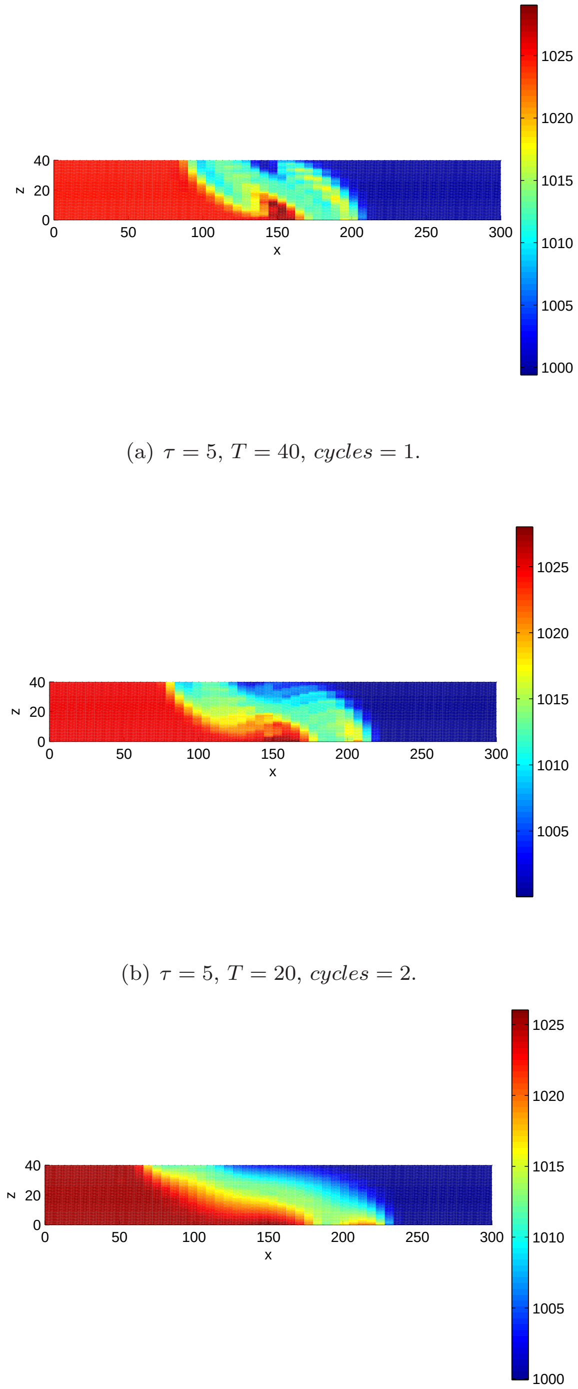
$$\tau \leq 2.2. \tag{4.12}$$

Rotating brackish zone after 200 days

The contour lines of the rotating brackish zone example are shown after 200 days with different values for the time step dt, the number of time steps of the transport equation T and the number of cycles of the coupled system *cycles*. In Figure 4.30 the results with the time step $dt = 5$ *days* are shown. In Figure 4.30(a) only one cycle is used, in Figure 4.30(b) 2 cycles are used and in Figure 4.30(c) 20 cycles of the coupled system are used to calculate the salt transport after 200 days. These figures show the necessity of the coupling of the flow equation and the transport equation. Note that τ does not satisfy the CFL condition, but Figure 4.30(c) still gives a stable solution. For the CFL condition, the most severe condition is used and the element distances are chosen half the maximum element distance of the triangle. In this example, it is possible to choose a less strict condition for the time step.

In Figure 4.31 the rotating brackish zone after 200 days in shown for the time step $dt = 0.5$. For Figure 4.31(a) only 2 cycles are used, it can be seen that the density transport is different than in Figure 4.31(b) or Figure 4.32(b). Also for the time step $dt = 0.05$ in Figure 4.34 the influence of the number of cycles is clear. The Figures 4.32(b) and 4.32(b) with 20 and 200 cycles give a better salt transport than Figure 4.32(a), with only 2 cycles.

From the Figures 4.30 ,4.31 and 4.34 can be concluded that the time passing by in the transport equation should be less than 100 days in this example. The Figures 4.30(b), 4.31(a) and 4.32(a) all have $dt \times T = 100$ *days* and give bad results. Taking $dt \times T = 10$ *days*, as in the Figures 4.30(c), 4.31(b) and 4.32(a) gives better results. Using more cycles does not influence the salt transport in this example, the Figures 4.32(b) and 4.32(c) show the same contour lines for the density.

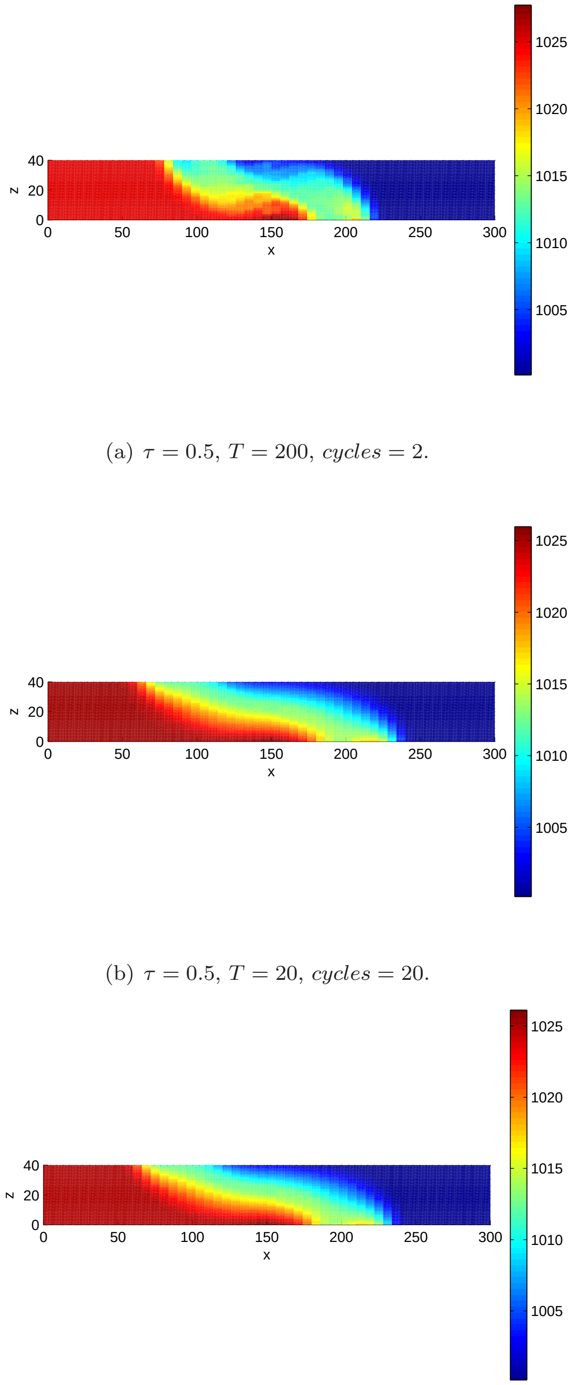


(a) $\tau = 5$, $T = 40$, $cycles = 1$.

(b) $\tau = 5$, $T = 20$, $cycles = 2$.

(c) $\tau = 5$, $T = 2$, $cycles = 20$.

Figure 4.30: Rotating brackish zone after **200 days**, with τ the time step of the transport equation, T the number of time steps of the transport equation and *cycles* the number of cycles of the coupled system of the flow equation and the transport equation. $\tau = \mathbf{5}$ and (a) $T = 40$, $cycles = 1$, (b) $T = 20$, $cycles = 2$ and (c) $T = 2$, $cycles = 20$.

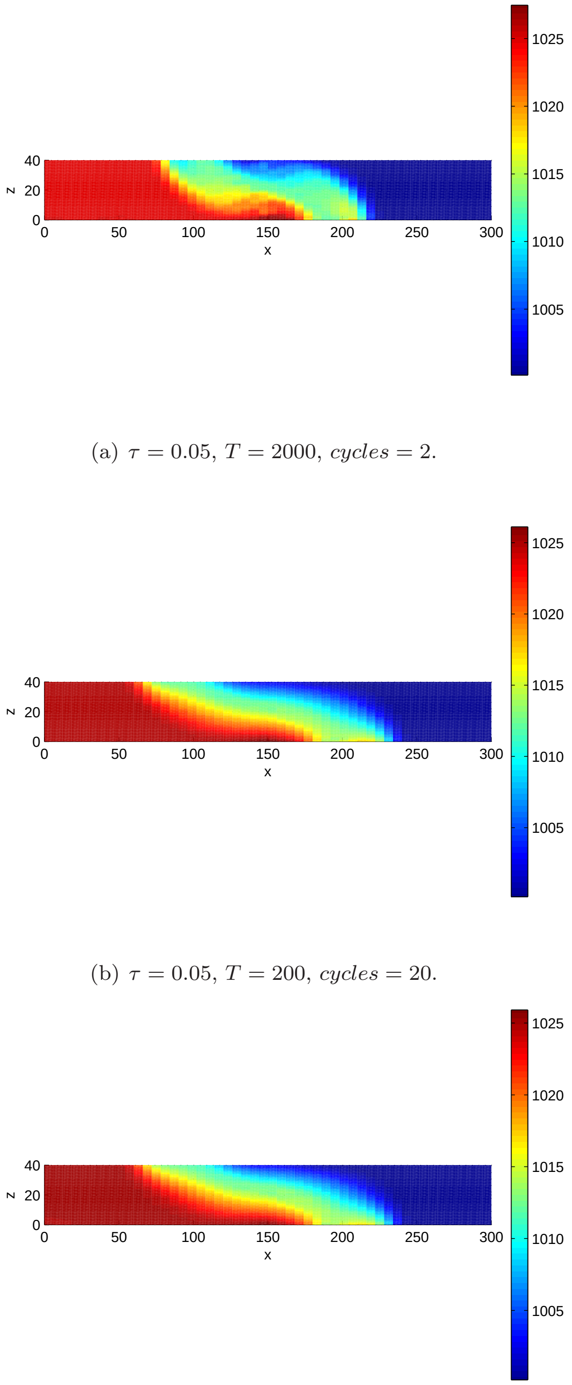


(a) $\tau = 0.5$, $T = 200$, *cycles* $= 2$.

(b) $\tau = 0.5$, $T = 20$, *cycles* $= 20$.

(c) $\tau = 0.5$, $T = 2$, *cycles* $= 200$.

Figure 4.31: Rotating brackish zone after **200 days**, with τ the time step of the transport equation, T the number of time steps of the transport equation and *cycles* the number of cycles of the coupled system of the flow equation and the transport equation. $\tau = \mathbf{0.5}$ and (a) $T = 200$, *cycles* $= 2$, (b) $T = 20$, *cycles* $= 20$ and (c) $T = 2$, *cycles* $= 200$.



(a) $\tau = 0.05$, $T = 2000$, $cycles = 2$.

(b) $\tau = 0.05$, $T = 200$, $cycles = 20$.

(c) $\tau = 0.05$, $T = 20$, $cycles = 200$.

Figure 4.32: Rotating brackish zone after **200 days**, with τ the time step of the transport equation, T the number of time steps of the transport equation and *cycles* the number of cycles of the coupled system of the flow equation and the transport equation. $\tau = \mathbf{0.05}$ and (a) $T = 2000$, $cycles = 2$, (b) $T = 200$, $cycles = 20$ and (c) $T = 20$, $cycles = 200$.

Rotating brackish zone after 1000 days

In Figure 4.33 the rotating brackish zone is shown after 1000 *days* with time step $dt = 0.5$, number of time steps of the transport equation $T = 20$ and number of cycles $cycles = 100$. It can be seen that the brackish zone rotates to a horizontal position through time, with the freshwater on top of the brackish water and the brackish water on top of the salt water.

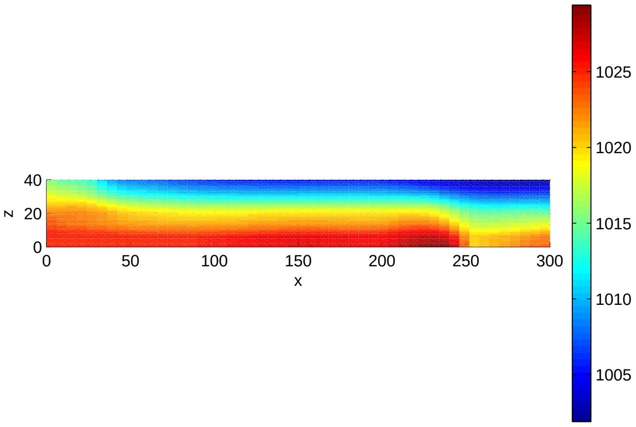


Figure 4.33: Density in the rotating brackish zone after 1000 days with $dt = 0.5$, $T = 20$ and $cycles = 100$.

Some notes on accuracy

The rotating brackish zone in Figure 4.33 is calculated by a coupling between the flow equation and the transport equation. In the initial Darcy velocities in Figure 4.28 and Figure 4.29 are some unexpected nonzero velocities at the left side of the rotating brackish zone. These velocities are calculated with Darcy's law after solving the flow equation with the SGA. The system of equations derived for the flow equation is solved with the BiCGSTAB method. In Section 3.7 some notes where made about solving this system of equations

$$Th_f = f.$$

In order to be able to find a solution, the freshwater head is given (fixed) in one point of the grid. This trick can help the Bi-CGSTAB method to choose a good solution. When this trick does not improve the solution another approach is necessary. If the problem $Th_f = f$ is singular (for example due to the Neumann boundary condition), there is no unique solution. In order to obtain a unique solution, the vector f should be in the Span(T).

The BiCGSTAB method in Matlab indeed finds a good solution by fixing the freshwater head in one point in the grid, but without a high accuracy. This method in Matlab gives for the example of Figure 4.33 a maximum relative residual of 0.41. This relative residual is defined as

$$\text{relative residual} = \text{norm}(f - T * h_f)/\text{norm}(f).$$

In order to investigate the accuracy of solving the system of equations $Th_f = f$, the area of the rotating brackish zone is fully filled with salt water with a density $\rho = 1025$. The freshwater head is defined as

$$h_f = \frac{p}{\rho_f g} + z, \tag{4.13}$$

in Section 2.1 and shown in Figure 2.1. In this equation, p is the pressure of the groundwater, ρ_f the density of freshwater, g the acceleration due to gravity and z the vertical coordinate of the location of measure. In Figure 4.34(a) the freshwater head calculated by the flow equation in Matlab is given for the salt water. From Equation (4.13) can be seen that the pattern of this figure is correct.

In Matlab, it is possible to define the accuracy tolerance for the BiCGSTAB method. For the Figures 4.34(b) and 4.34(c) an accuracy tolerance of 10^{-6} is used. Remember from Section 3.7 that the Darcy velocity is calculated by central differences of the gradient of the freshwater head ($q_x = -k\frac{\partial h_f}{\partial x}$). The differences between the freshwaterhead in the x-direction have to be zero in this example. In Figure 4.34(b) can be seen that there is an error of the size 10^{-7}. The Darcy velocity in the z-direction is calculated by

$$q_z = -k\left(\frac{\partial h_f}{\partial z} + \frac{\rho - \rho_f}{\rho_f}\right).$$

Central differences of the gradient of the freshwater head again gives an error of the order 10^{-7}. In the Figures 4.34(d) and 4.34(e) the Darcy velocities in the x- and z-direction are shown calculated by using an accuracy tolerance of 10^{-12} in the BiCGSTAB method. An error in these velocities can be seen of the order 10^{-13}.

In order to make the system of equations $Th_f = f$ solvable, the right hand side vector f can be updated. The matrix T is almost singular. In order to make the system of equations $Th_f = f$ solvable

$$T\mathbf{v} = \mathbf{0},$$

can be calculated. The vector $\mathbf{v}$ is now an eigenvector belonging to the eigenvalue 0. If

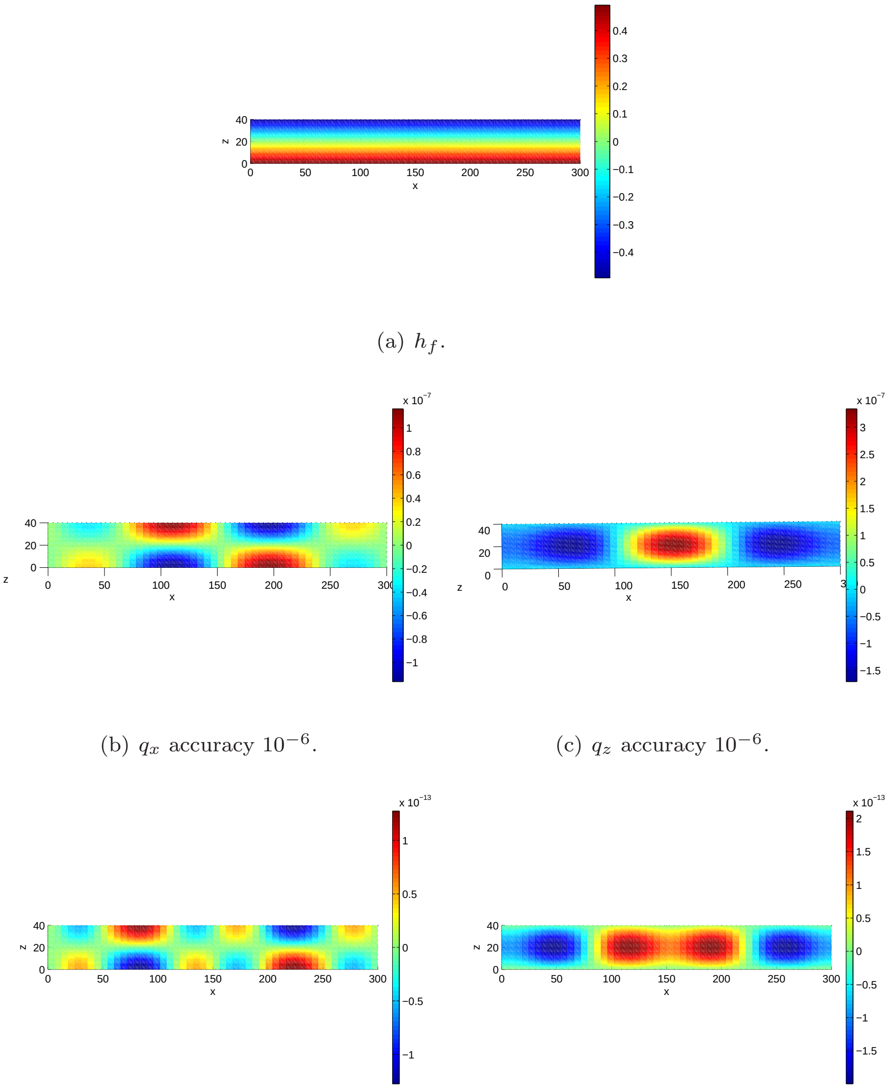
$$\mathbf{v}^T f = \mathbf{0},$$

the system of equations is solvable. Otherwise use

$$f^{new} = f - \frac{\mathbf{v}^T f}{||\mathbf{v}||_2^2} \cdot \mathbf{v}.$$

Now the system of equation is compatible.

Unfortunately, the matrix T does not have the eigenvalue zero for this example. The smallest eigenvalue is -1.1×10^{-11}. Because this eigenvalue is close to zero, the eigenvector belonging to this eigenvalue may be used as the vector



(a) h_f.

(b) q_x accuracy 10^{-6}.

(c) q_z accuracy 10^{-6}.

(d) q_x accuracy 10^{-12}.

(e) q_z accuracy 10^{-12}.

Figure 4.34: Impermeable domain Ω filled with salt water with $\rho = 1025\ kg/m^3$. (a) freshwater head calculated with the flow equation in Matlab (b) q_x with the BiCGSTAB method with a given accuracy of 10^{-6}, (c) q_z with the BiCGSTAB method with a given accuracy of 10^{-6}, (d) q_x with the BiCGSTAB method with a given accuracy of 10^{-12} and (e) q_z with the BiCGSTAB method with a given accuracy of 10^{-12}.

v. More research is needed in order to increase the accuracy of the solution of the flow equation.

It can be concluded that in order to obtain a more accurate solution for the density pattern of the rotating brackish zone, the method to solve the flow equation should be improved.

4.3.2 Cycles with Triwaco and Matlab

The example of the rotating brackish zone as shown in Figure 4.26 is again considered. A box of 300 meters in the x-direction, 3 meters in the y-direction and 40 meters in the z-direction is taken. One aquifer has 150 elements and 152 nodes, the z-direction is divided into 40 aquifers and 39 model aquitards. The flow equation is solved with Triwaco and coupled to the transport equation, which is solved in Matlab. The initial density profile with the salt, brackish and freshwater zones is shown in Figure 4.35. Note that the grid used in the rotating brackish zone example coupled in Matlab (Section 4.3.1) used 51 nodes per aquifer and 41 layers in the vertical direction, so the element sizes are comparable.

Note that the Darcy velocity is shown in the figures, the seepage velocity $\mathbf{v}$ can be calculated by

$$\mathbf{v} = \frac{\mathbf{q}}{\theta}.$$

A symmetric profile is expected as shown in Figure 4.28 and 4.29 for the Darcy velocities q_x and q_z:

$$q_{x_{min}}(\text{aquifer 1}) = -q_{x_{max}}(\text{aquifer 40}),$$

and

$$q_{x_{min}}(\text{aquifer 40}) = -q_{x_{max}}(\text{aquifer 1}).$$

The velocity profile calculated by the flow equation in Triwao is indeed symmetric, see Figure 4.36(a) and Figure 4.36(c). Note that these velocities show wiggles in the maximum and minimum velocity per aquifer. The velocity in the y-direction is expected to be zero, but due to the small amount of elements in this direction, the error of the Darcy velocity is of the order 10^{-4} (Figure 4.36(b)). Compared to the maximum Darcy velocity in the x-direction (2×10^{-2}) and the z-direction (8×10^{-3}), this is a large error.

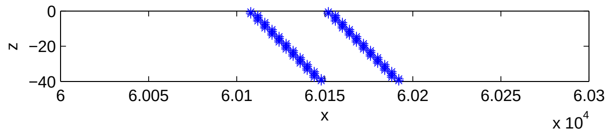
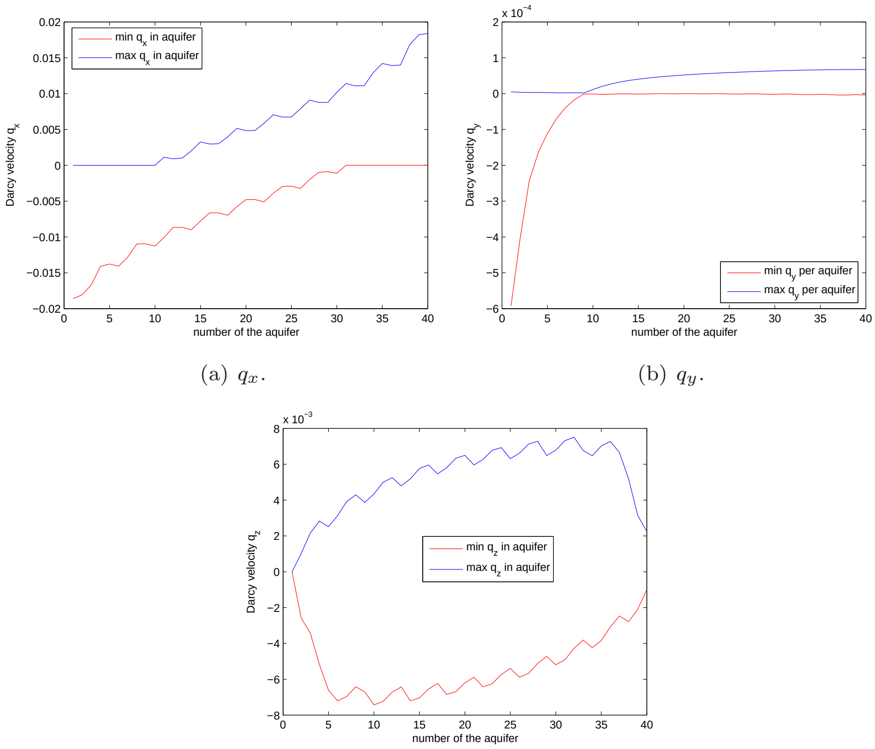


Figure 4.35: Initial Condition Rotating Brackish Zone: salt($\rho = 1024$)-brackish-fresh($\rho = 1001$) front.



(a) q_x.

(b) q_y.

(c) q_z.

Figure 4.36: Initial state Rotating Brackish Zone: (a) minimum and maximum Darcy velocity q_x per aquifer, (b) minimum and maximum Darcy velocity q_y per aquifer and (c) minimum and maximum Darcy velocity q_z per aquifer.

Darcy velocities in rotating brackish zone

To investigate what goes wrong, the rotating brackish zone is considered after 0.002 *days*, with time step $dt = 0.001$, number of time steps of the transport equation $T = 1$ and the number of cycles between the transport equation and flow equation $cycles = 2$. The salt, brackish and fresh zones are given in Figure 4.37, there is no visible difference between the initial zones (Figure 4.35) and the density pattern after 0.002 *days* (Figure 4.37). The time step $dt = 0.001$ is so small that even in the individual aquifers the density has not significantly changed.

The velocities in all directions calculated by the flow equation in Triwaco are expected to (almost) equal the velocities calculated at the initial state. The minimum and maximum velocities in the y- and z-direction, Figure 4.38(b) and Figure 4.38(c) are indeed equal to the the Figures 4.36(b) and 4.36(c). The minimum and maximum velocity in the x-direction in all aquifers, as shown in Figure 4.38(a), on the other hand differs from the initial velocities in this direction (Figure 4.36(a)). For example in aquifer number 39, the Darcy velocity q_x has at the initial state values between 0 and 0.17 m/day (Figure 4.36(a)), after 0.002 days this Darcy velocity has values between -0.15 and 0.017 m/day. Figure 4.39 shows the contour lines of the Darcy velocity q_x in aquifer 39. The minimum Darcy velocity of -0.15 m/day takes place around $x = 60190$ m, near the brackish-freshwater front.

The errors in the initial velocity in the y-direction, the wiggles in the initial velocities q_x and q_z and the large error in the velocity q_x after 0.002 *days* lead to unphysical solutions.

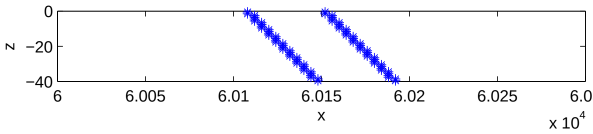
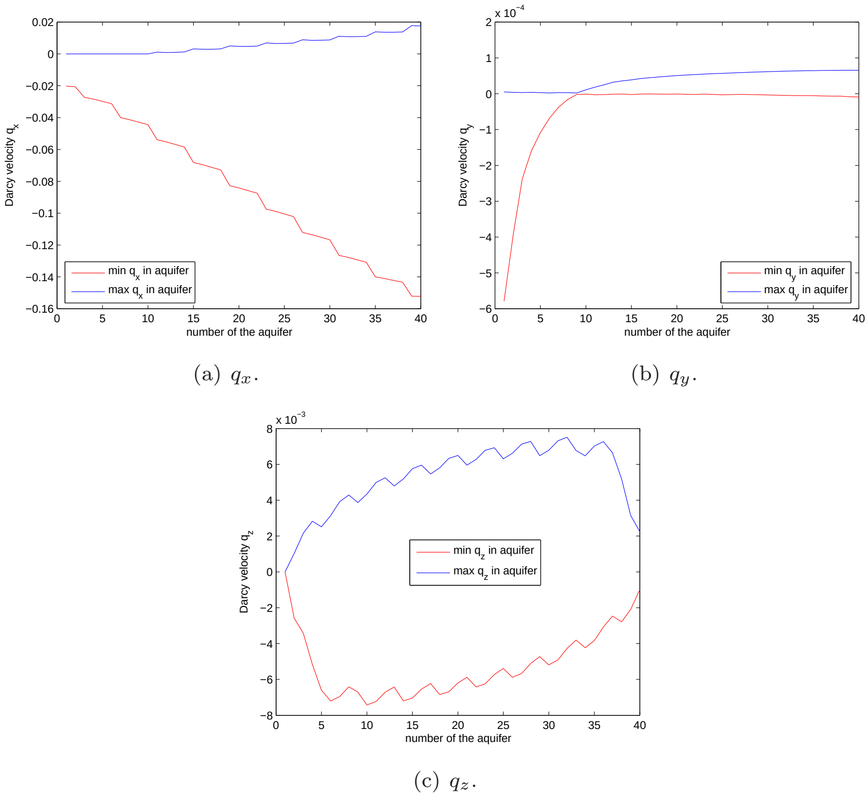


Figure 4.37: salt($\rho = 1024$)-brackish-fresh($\rho = 1001$) front.



(a) q_x.

(b) q_y.

(c) q_z.

Figure 4.38: Rotating Brackish Zone after 0.002 *days*, with time step $dt = 0.001$, number of time steps $T = 1$ and number of cycles $cycles = 2$, (a) minimum and maximum Darcy velocity q_x per aquifer, (b) minimum and maximum Darcy velocity q_y per aquifer and (c) minimum and maximum Darcy velocity q_z per aquifer.

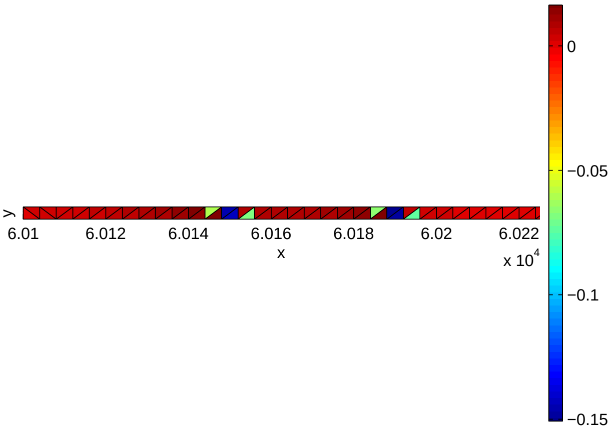


Figure 4.39: Darcy velocity q_x around the brackish zone in aquifer 39.

4.4 Density dependent flow: freshwater mining

In the introduction an example was given of a well in the groundwater (see Figure 1.2 in Section 1.1). The density of freshwater is less than the density of salt water, hence the freshwater is on top of the salt water in the subsurface. An application of density dependent flow is a tap in the dunes in the Netherlands. In the subsurface in the dunes is a so-called freshwater lens on top of the salt water. When mining of freshwater takes place, a brackish zone appears from the salt water in the direction of the sink, as shown in Figure 1.2. With this density dependent groundwater flow model, the influence of sinks with different capacities can be investigated and the necessity of a possible freshwater source to restore the freshwater lens can be explained. Simulations with the density dependent flow model can safe the freshwater lens for future generations.

The cross-section of the three dimensional area used for the simulations is shown in Figure 4.40. The area has the shape of a cylinder, with a radius of 1000 meters and a height of 40 meter. On top of the cylinder is a tap for mining freshwater with a capacity of 100 m^3/day. At the bottom of the cylinder is a continuous salt water source in order to keep the amount of water in the area constant. the porosity is chosen $\theta = 0.2$. The steady flow equation is solved in Triwaco and coupled to the transport equation, which is numerically solved with Matlab.

Neumann boundary conditions are used for both the flow and transport equation. The sides as well as the top of the cylinder, are impermeable, hence the homogeneous Neumann boundary condition is chosen for the transport equation. At the bottom is a continuous source of salt water, which results in a nonzero flux of salt. The boundary condition for the transport equation at the bottom is

$$\nabla C \cdot n = C_{salt} \times q_z \times V, \tag{4.14}$$

with $C_{salt} = 17.5\ kg/m^3$ the concentration of salt in salt water with a density of $\rho = 1025\ kg/m^3$, q_z the Darcy velocity normal to the bottom and V the node influence area (the area of the column of water for this node). For the flow equation, the condition $q_x = 0$ at the sides and the top of the cylinder leads to the homogeneous boundary condition $\frac{\partial h_f}{\partial x} = 0$. For the bottom a continuous flux of salt water leads to a nonzero Neumann boundary condition. The initial condition of the transport equation is determined by the given density in the cylinder at the initial state.

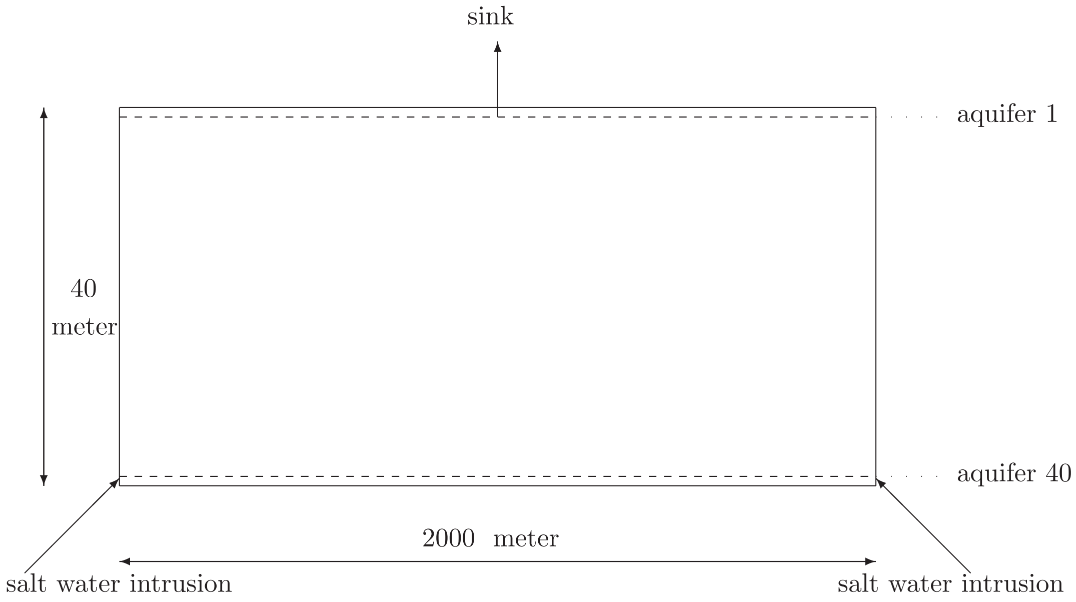


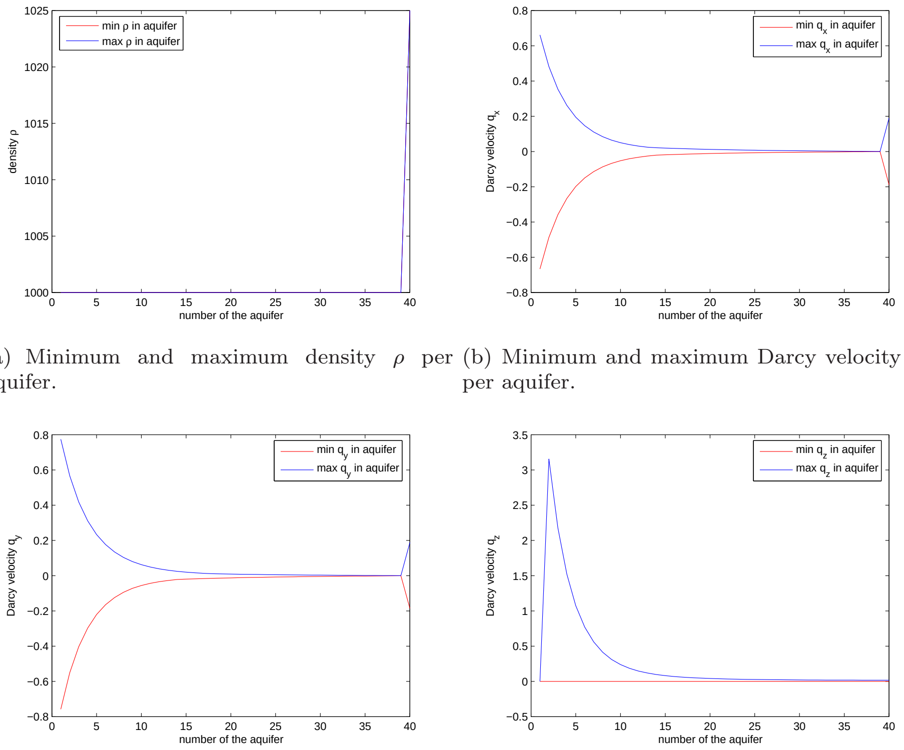
Figure 4.40: freshwater mining: the cross-section of the cylinder.

Initial state

The cylinder shown in Figure 4.40 is divided into 40 aquifers and 39 model aquitards. For the transport equation only the aquifers are used for the grid. The sink is placed in the middle of aquifer 1 and aquifer 40 is always filled with salt water. One aquifer has 142 nodes, so for the transport equation a total number of 5680 nodes is used to calculate the salt transport. The irregular and unstructured grid has 262 elements per aquifer and is shown in for example Figure 4.42.

In Figure 4.41 the minimum and maximum Darcy velocities at the initial state are shown. A realistic value for the dispersion parameter D is one tenth of the velocity of the groundwater. The maximum seepage velocity in the horizontal direction is $q_{y_{max}}/\theta = 0.78/0.2 = 3.9\ m/day$, hence $D = 0.4$ is taken in the numerical experiments.

The infiltration of salt water in aquifer 40 as well as the sink in aquifer 1 take care for the nonzero horizontal velocities q_x and q_y. The nonzero velocities q_x and q_y in the first 15 aquifers are due to the freshwater mining in aquifer 1, these velocities are pointed into the direction of the sink, as can be seen in the Figures 4.42 and 4.43. The velocity in the z-direction has its maximum in aquifer 2, one aquifer away from the freshwater mining (see Figure 4.41(d)). This maximum takes only place in the node beneath the sink as can be seen in Figure 4.43(f), all other nodes in aquifer 2 have a zero velocity in the z-direction. A comparison between the values of the velocities in the Figures 4.42 and 4.43 shows larger velocities in aquifer 1 (for q_x and q_y) and aquifer 2 (for q_z) than in aquifer 39. On the other hand, in aquifer 39 are more nodes with a nonzero velocity.



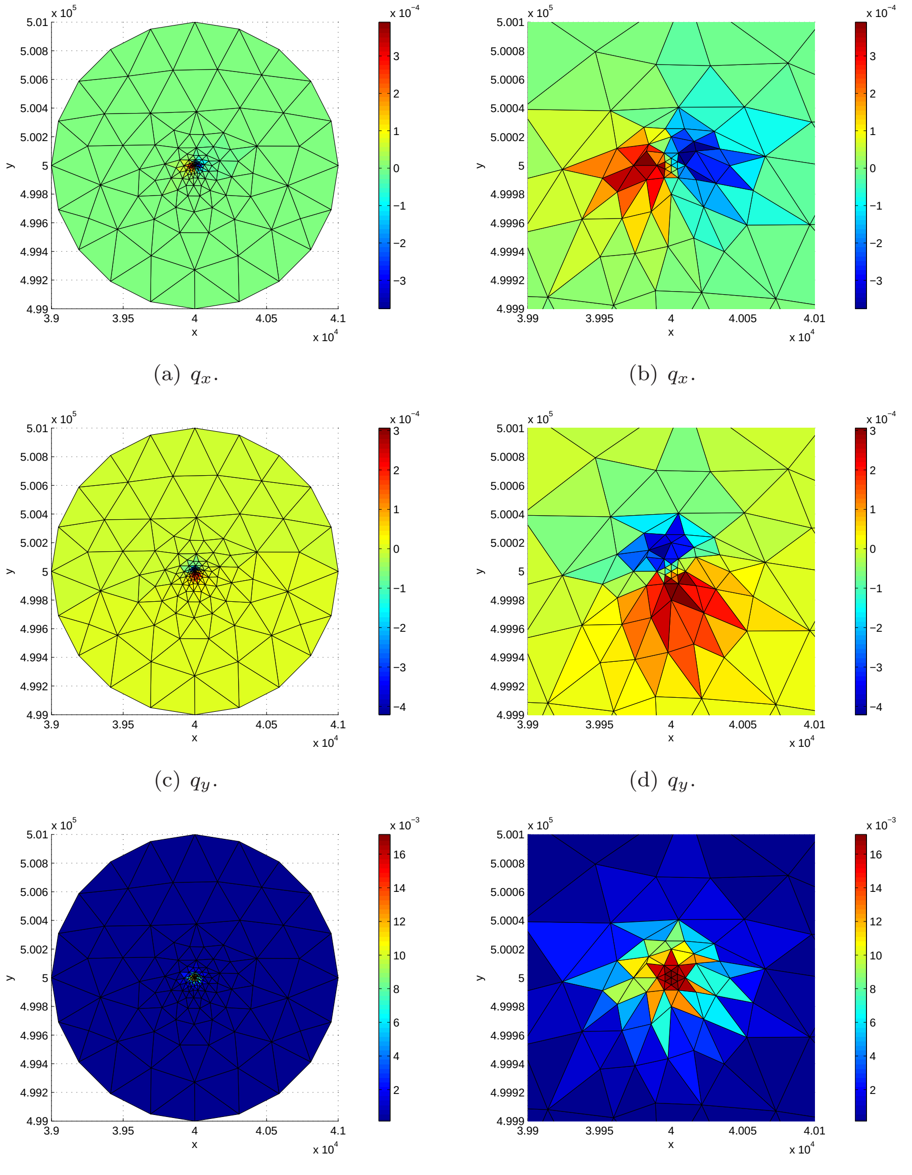
(a) Minimum and maximum density ρ per aquifer.

(b) Minimum and maximum Darcy velocity q_x per aquifer.

(c) Minimum and maximum Darcy velocity q_y per aquifer.

(d) Minimum and maximum Darcy velocity q_z per aquifer.

Figure 4.41: **Initial state** freshwater mining, (a) minimum (red) and maximum (blue) density ρ per aquifer, (b) minimum and maximum Darcy velocity $\mathbf{q_x}$ per aquifer, (c) minimum and maximum Darcy velocity $\mathbf{q_y}$ per aquifer and (d) minimum and maximum Darcy velocity $\mathbf{q_z}$ per aquifer.



(a) q_x.

(b) q_x.

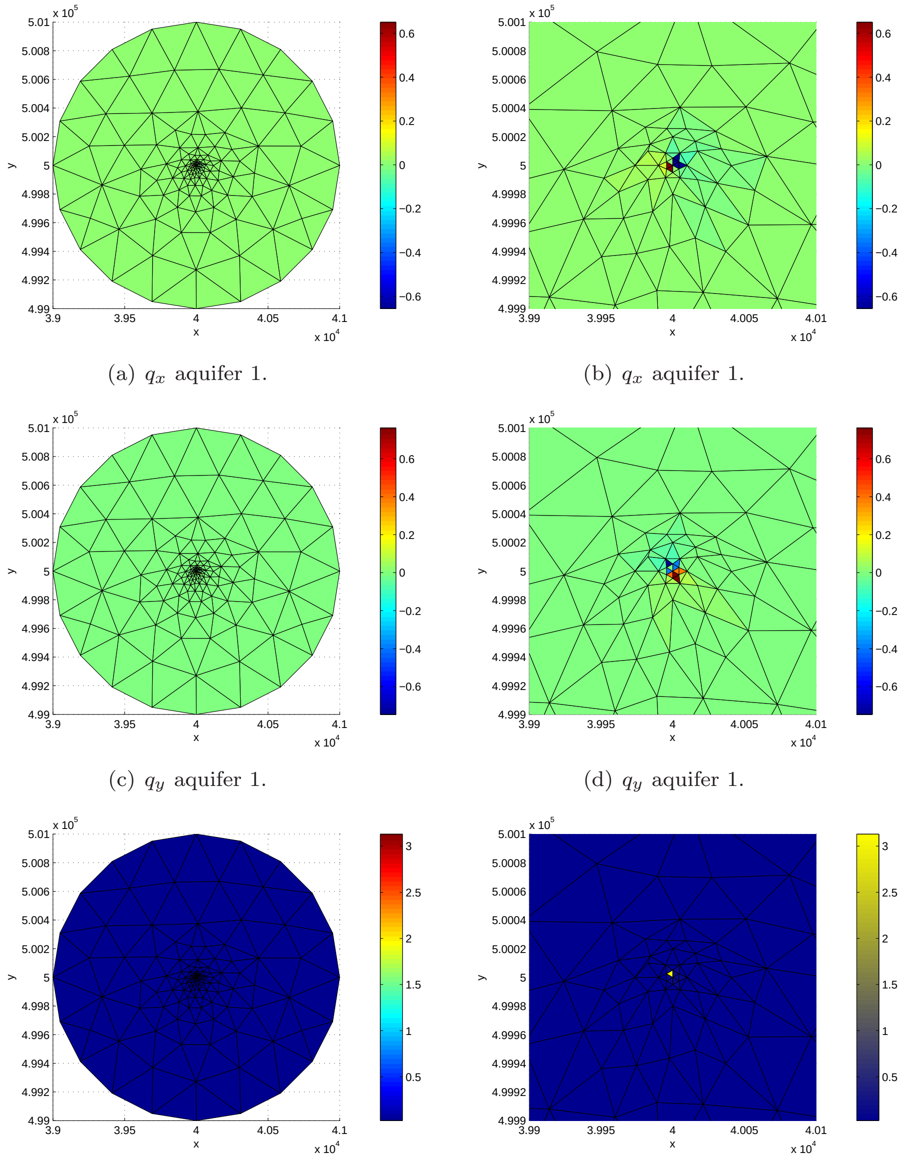
(c) q_y.

(d) q_y.

(e) q_z.

(f) q_z.

Figure 4.42: Initial state freshwater mining in **aquifer 39**, (a) Darcy velocity $\mathbf{q_x}$ in aquifer 39, (b) q_x in the center of aquifer 39, (c) Darcy velocity $\mathbf{q_y}$ in aquifer 39, (d) q_y in the center of aquifer 39, (e) Darcy velocity $\mathbf{q_z}$ in aquifer 39 and (e) q_z in the center of aquifer 39.



(a) q_x aquifer 1.

(b) q_x aquifer 1.

(c) q_y aquifer 1.

(d) q_y aquifer 1.

(e) q_z aquifer 2.

(f) q_z aquifer 2.

Figure 4.43: Initial state freshwater mining in **aquifer 1**, (a) Darcy velocity $\mathbf{q_x}$ in aquifer 1, (b) q_x in the center of aquifer 1, (c) Darcy velocity $\mathbf{q_y}$ in aquifer 1, (d) q_y in the center of aquifer 1, and in **aquifer 2** (e) Darcy velocity $\mathbf{q_z}$ in aquifer 2 and (e) q_z in the center of aquifer 2.

Time step

The CFL condition is defined as

$$|\frac{q_x \tau}{\theta \Delta x}| + |\frac{q_y \tau}{\theta \Delta y}| + |\frac{q_z \tau}{\theta \Delta z}| \leq 1, \tag{4.15}$$

and determines an upper bound for the time step τ.

The porosity in this model is $\theta = 0.2$, hence the Darcy velocities have to be divided bij 0.2 in order to determine the seepage velocities. Figure 4.41(b) shows the maximum Darcy velocity in the x-direction q_x, from which the seepage velocity can be calculated: $q_{x_{max}}/\theta = 0.68$. Figure 4.41(c) shows these values for q_y: $q_{y_{max}}/\theta = 0.78$, and Figure 4.41(d) for q_z: $q_{z_{max}}/\theta = 3.2 \ m/day$.

For regular triangular elements the distance Δx is chosen half the maximum distance of the element in the x-direction. Figure 4.42 shows an irregular and unstructured grid. In this case half the maximum element distance of the smallest element in the grid is chosen as Δx. The parameter Δy is determined equivalent. The minimum distance in the x-direction as well as in the y-direction of one element of the irregular grid is approximately 5 meter, hence $\Delta x = \Delta y = 2.5$ in the CFL condition. The element distance in the z-direction equals the distance from the middle of an aquifer to the middle of the next aquifer and is 1 meter ($\Delta z = 1$).

The CFL condition can now be used in order to determine the upper bound for the time step τ

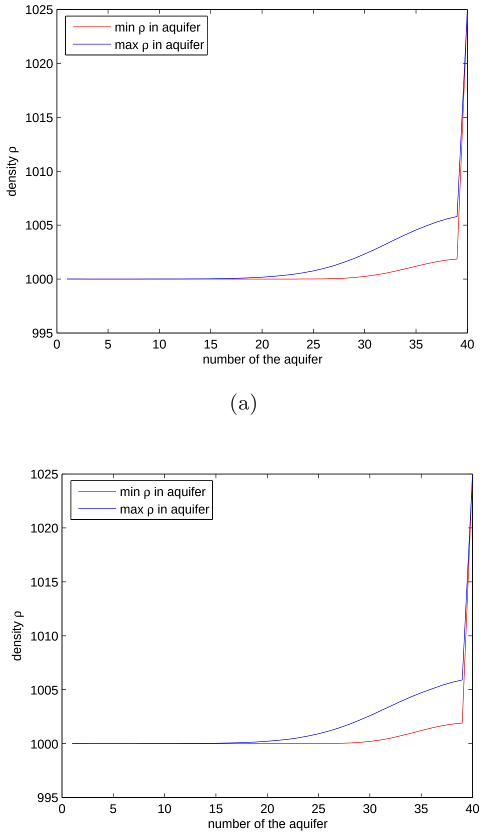
$$\tau \leq 0.052. \tag{4.16}$$

Numerical experiments have shown that taking $\tau = 0.05$ indeed satisfies, hence in all numerical experiments of the freshwater mining example the time step is chosen $\tau = 0.05$.

150 days

The velocity induced by the density differences is much smaller than the velocity caused by the freshwater mining, hence there is no visible difference in salt distribution between using the coupling between the flow and transport equation or just using the transport equation with the velocity given by the flow equation ($cycle = 1$). Besides that, the dispersion coefficient is $D = 0.4$, which also overrules the solute transport by the velocity caused by the density differences.

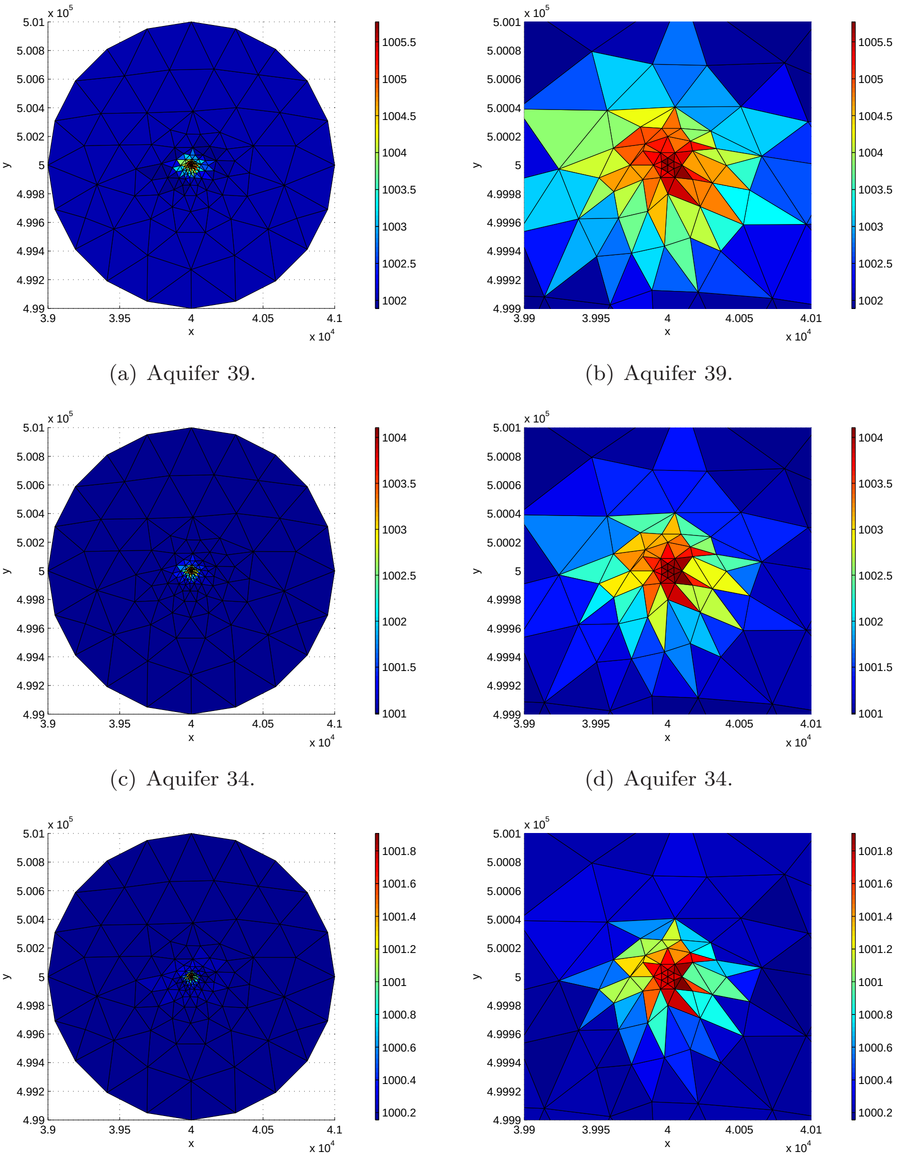
In Figure 4.44(a) the minimum and maximum density ρ per aquifer is shown after 150 days with time step $dt = 0.05$, number of time steps of the transport equation $T = 10$ and number of cycles $cycles = 300$. This figure gives the same results as Figure 4.44(b), which shows the minimum and maximum density ρ per aquifer after 150 days with time step $dt = 0.05$, number of time steps of the transport equation $T = 3000$ and number of cycles $cycles = 1$. The density in aquifer 40 is 1025 kg/m^3 (salt water), the density decreases in the direction of the sink. In Figure 4.45 the contour lines of the density ρ in the aquifers 39, 34 and 29 is shown. There is no visible difference between these figures and the contour plots of the density after 150 days with $cycle = 1$.



(a)

(b)

Figure 4.44: Minimum (red) and maximum (blue) density ρ per aquifer after **150 days** with $dt = 0.05$, (a) number of time steps transport equation $T = 10$ and $cycles = 300$ and (b) $T = 3000$ and number of cycles $cycles = 1$.



(a) Aquifer 39.

(b) Aquifer 39.

(c) Aquifer 34.

(d) Aquifer 34.

(e) Aquifer 29.

(f) Aquifer 29.

Figure 4.45: Freshwater mining example after **150 days** with time step $dt = 0.05$, number of time steps transport equation $T = 10$ and $cycles = 300$. Figure (a) shows the density ρ of aquifer 39, (b) ρ in the center of aquifer 39, (c) ρ in aquifer 34, (d) ρ in the center of aquifer 34, (e) ρ in aquifer 29 and (f) ρ in the center of aquifer 29.

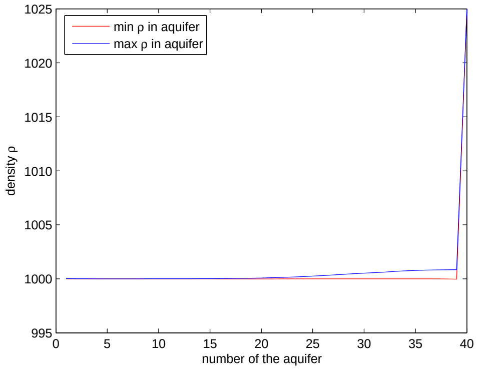
Pure advection

The freshwater mining example is investigated without dispersivity ($D = 0$). In Figure 4.45(b) the minimum density $\rho_{min} = 1002\ kg/m^3$ in aquifer 39. In Figure 4.47(b) the minimum density $\rho_{min} = 1000\ kg/m^3$, hence freshwater. The extra 2 kg/m^3 after 150 days in Figure 4.45(b) are due to the dispersion coefficient $D = 0.4$.

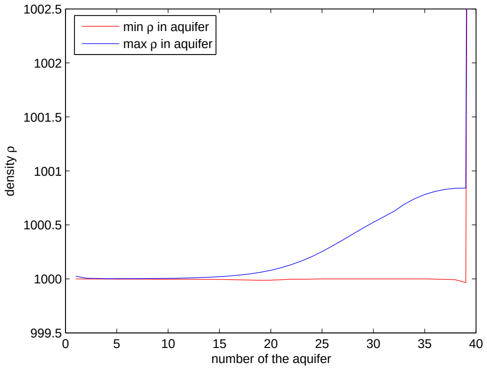
The transport induced by density differences instead of the sink may be a bit more visible when only advective transport is considered ($D = 0$). In the Figures 4.46 and 4.48 the minimum and maximum density ρ per aquifer is shown after 225 *days*. In Figure 4.46 the density is shown after 225 days with time step $dt = 0.05$, number of time steps of the transport equation $T = 10$ and $cycles = 300$. For Figure 4.48 only one cycle is used with time step $dt = 0.05$ and number of time steps of the transport equation $T = 4500$. There is no visible difference between both figures.

In Figure 4.47 the contour plots of the density in the aquifers 39, 30 and 21 is shown after 225 days with 450 cycles. In Figure 4.49 these contour plots are shown after 225 days with 1 cycle. In Figure 4.49 the density ρ is a bit larger in all grid points (difference is between 0.02 - 0.05 kg/m^3). The more spreading of the salt in Figure 4.47 may be caused by the transport due to density differences in the x- and y-direction. On the other hand, the density differences are that small that they can be caused by a numerical error.

For examples where a velocity caused by a source or sink is involved which overrules the velocity caused by density differences, it is not necessary to use the coupled model. In order to calculate the salt transport after a certain time, it satisfies to use the transport equation with the initial velocities calculated by the flow equation. The flow will not be influenced by the density differences because of the large velocities caused by the source or sink.

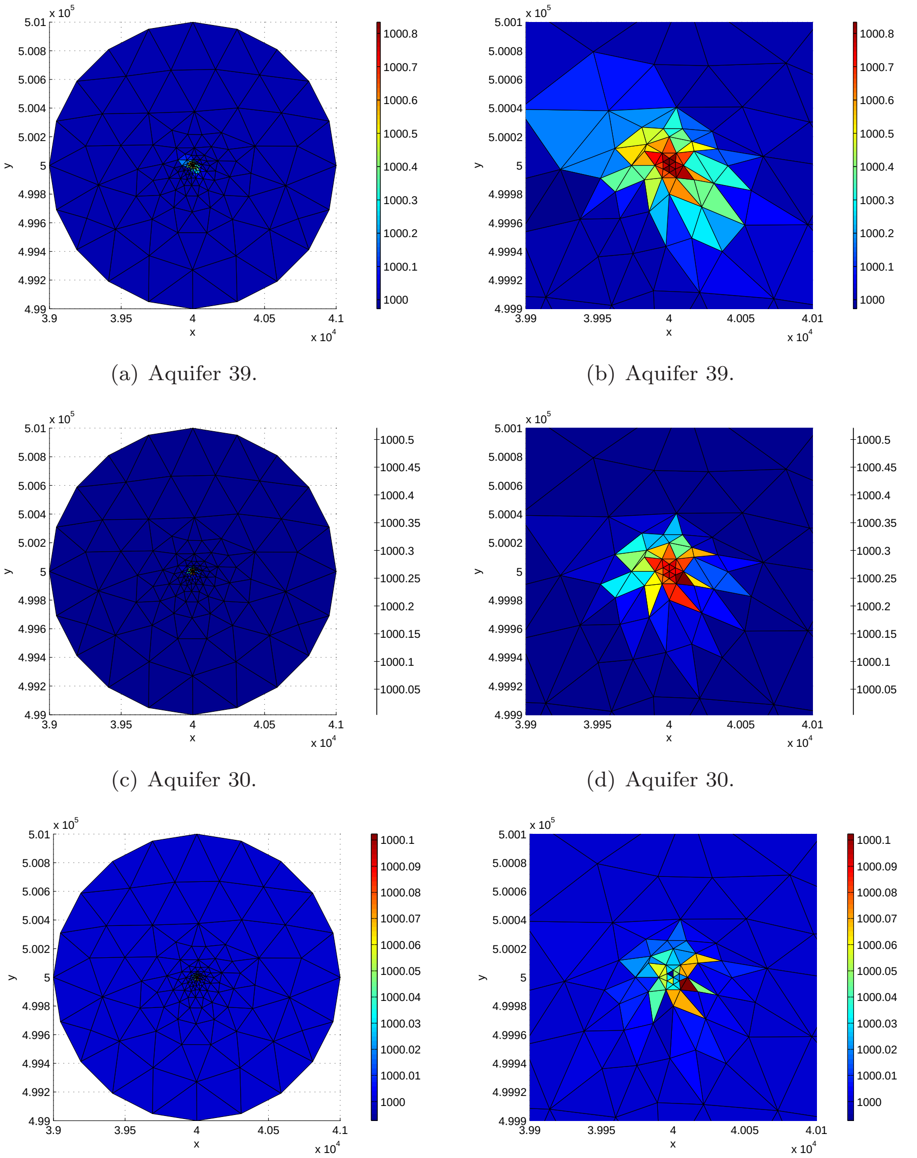


(a)



(b)

Figure 4.46: Minimum (red) and maximum (blue) density ρ per aquifer after 150 days with $dt = 0.05$, number of time steps transport equation $T = 10$ and $cycles = 300$, dispersion coefficient $\mathbf{D} = \mathbf{0}$. (b) Same as Figure (a), enlarged on the vertical axis.



(a) Aquifer 39.

(b) Aquifer 39.

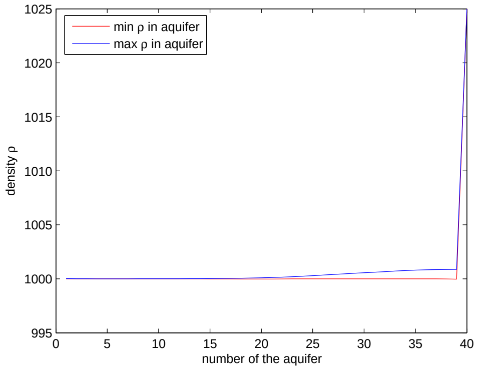
(c) Aquifer 30.

(d) Aquifer 30.

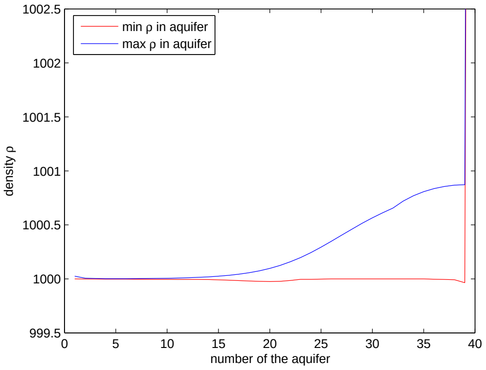
(e) Aquifer 21.

(f) Aquifer 21.

Figure 4.47: Freshwater mining example after **225days** with time step $dt = 0.05$, number of time steps transport equation $\mathbf{T = 10}$ and **cycles** $= \mathbf{450}$. Dispersion coefficient $\mathbf{D = 0}$. Figure (a) shows the density ρ of aquifer 39, (b) ρ in the center of aquifer 39, (c) ρ in aquifer 30, (d) ρ in the center of aquifer 30, (e) ρ in aquifer 21 and (f) ρ in the center of aquifer 21.

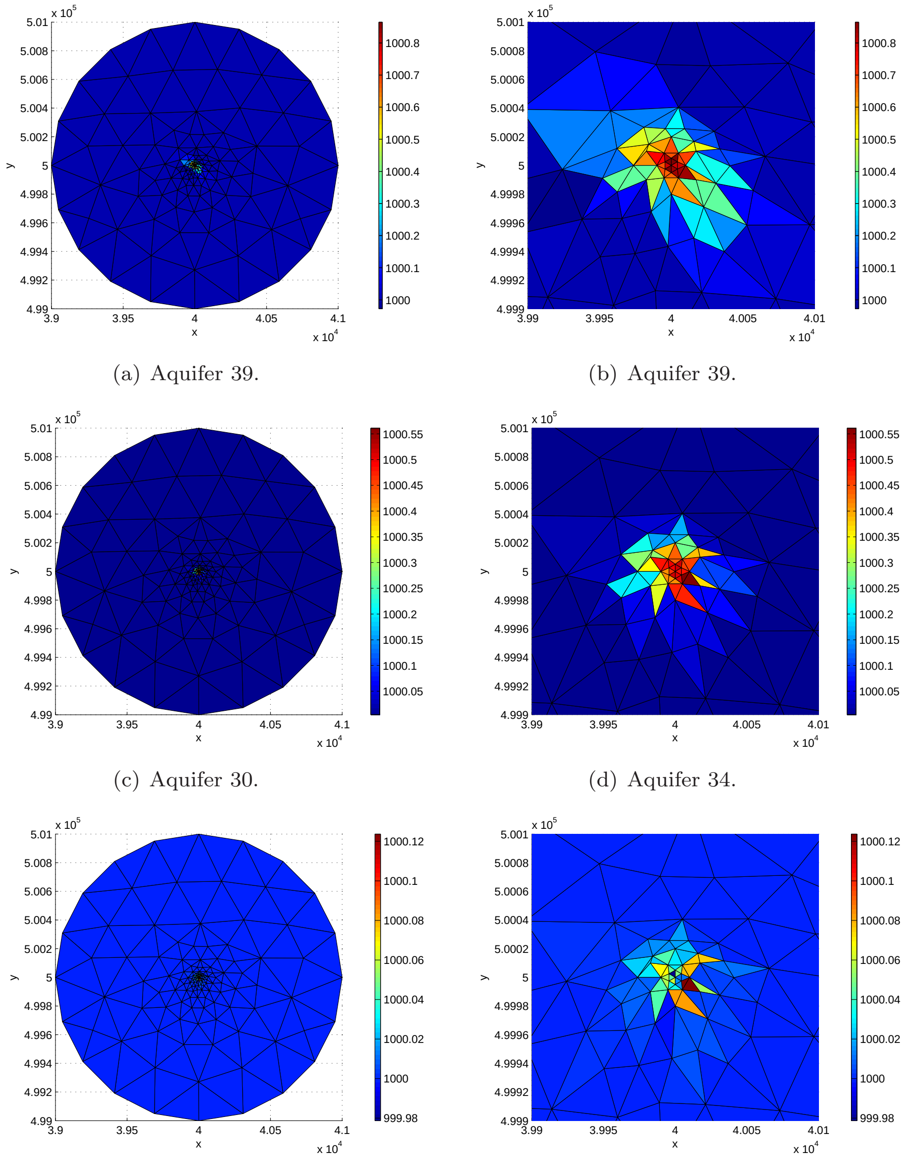


(a)



(b)

Figure 4.48: (a) Minimum (red) and maximum (blue) density ρ per aquifer after **225 days** with $dt = 0.05$, number of time steps transport equation $T = 4500$ and $cycles = 1$. Dispersion coefficient $\mathbf{D} = \mathbf{0}$. (b) Same as Figure (a), enlarged on the vertical axis.



(a) Aquifer 39.

(b) Aquifer 39.

(c) Aquifer 30.

(d) Aquifer 34.

(e) Aquifer 21.

(f) Aquifer 21.

Figure 4.49: Freshwater mining example after **225 days** with time step $dt = 0.05$, number of time steps transport equation $\mathbf{T} = \mathbf{4500}$ and **cycles** $= \mathbf{1}$. Dispersion coefficient $D = 0$. Figure (a) shows the density ρ of aquifer 39, (b) ρ in the center of aquifer 39, (c) ρ in aquifer 30, (d) ρ in the center of aquifer 30, (e) ρ in aquifer 21, and (f) ρ in the center of aquifer 21.

Chapter 5

Conclusions and recommendations

5.1 Conclusions

5.1.1 Solute transport

For the two dimensional transport of salt, research is done on four variations of the Finite Element Method; the Standard Galerkin Approach, the SUPG pure advection algorithm by Mizukami, the SUPG classical upwind method and the Mizukami Hughes algorithm. For diffusive transport all methods give good results for solving the transport equation.

For advective transport on the other hand, there are differences in the quality of these methods. Numerical experiments were done on the two dimensional advection equation with constant velocity. The Standard Galerkin Approach shows wiggles in a large part of the domain and is instable for the advection equation. The SUPG pure advection algorithm by Mizukami shows smaller wiggles than the Standard Galerkin Approach, but still shows wiggles in a large part of the domain. The SUPG Classical Upwind method gives good results in most cases although with some wiggles. The Mizukami Hughes algorithm shows wiggles in some cases, but only at the path from the initial concentration to the concentration after a certain time. Special cases for the SUPG classical upwind method and the Mizukami Hughes algorithm can be distinguished.

SUPG classical upwind

The SUPG Classical Upwind method uses the test function $\eta(\mathbf{x}) = w(\mathbf{x}) + b(\mathbf{x})$, with $w(\mathbf{x})$ the classical test function and $b(\mathbf{x})$ a parameter used to take care for the upwind behaviour. The parameter $b(\mathbf{x})$ is defined as

$$b(\mathbf{x}) = \frac{h\xi}{2}\frac{\nabla\phi_i \cdot \mathbf{q}}{||\mathbf{q}||},$$

with

$$\xi = sign(\mathbf{q} \cdot \Delta\mathbf{x}).$$

The parameter h is the representative distance of the element in the direction of the velocity $\mathbf{q}$. When q_x and q_y are both positive or when q_x and q_y are both negative, this approach works well. When q_x is positive and q_y negative or vice versa, the representative distance in the element for both q_x and q_y positive is calculated. The algorithm developed by Segal used for the calculation of this representative distance takes the absolute value of the velocity $\mathbf{q}$ and the distance in that direction may be different than the distance in the direction q_x negative and q_y positive.

Another parameter to be discussed is the distance $\Delta\mathbf{x}$ in the definition of ξ. Research was done on different definitions for $\Delta\mathbf{x}$, but taking $\xi = 1$, independent on the sign of $\mathbf{q}$ or $\Delta\mathbf{x}$, gives the best results.

Mizukami Hughes algorithm

It seems that the Mizukami Hughes algorithm is better than the SUPG Classical Upwind method only when at least for half of the elements the velocity will be in the direction of the edge zone of node 1 of the element. When the direction of the velocity is in the direction of the vertex zone, the Mizukami Hughes algorithm gives unwanted wiggles.

Computations are much more time consuming for the Mizukami Hughes algorithm than for the SUPG Classical Upwind method, because the element matrices have to be recalculated every time step. This is necessary because the coefficients of the upwind part of the Mizukami Hughes algorithm depend on the gradients of the concentration C.

3D solute transport

The combination of the two dimensional Finite Element Method in the x- and y-direction and the 1D Finite Difference Method in the z-direction works well. A system of equations is derived with matrices with coefficients depending on the FEM as well as the FDM. The software developped with Matlab simulates three dimensional advective, dispersive and diffusive solute transport with equal covered distances in all directions. This three dimensional method has in the horizontal direction the characteristics of the SUPG classical upwind method (some wiggles for advective transport) and in vertical direction the characteristics of the FDM (numerical diffusion for advective transport).

Time

In the Interim Master's thesis [19], numerical experiments were presented for several temporal discretization schemes for the transport equation. The scheme which uses Backward Euler for the dispersion part and Forward Euler for the advective part of the solute transport gave the best results. A severe stability criterion for the time step τ for this scheme is

$$|\frac{q_x\tau}{\theta\Delta x}| + |\frac{q_y\tau}{\theta\Delta y}| + |\frac{q_z\tau}{\theta\Delta z}| \leq 1. \tag{5.1}$$

For structured grids with 2D FEM in the horizontal direction and 1D FDM in the vertical direction, a safe condition is to take for Δx and Δy half of the maximum element distances in respectively the x-direction and y-direction.

For an unstructured grids with irregular shaped triangular elements, the CFL condition is based on these distances in the smallest element. This is a severe but safe condition and a bigger time step may keep the method stable. The element distance Δz can be taken equal to the element size of the FDM, which is the distance between the middle of two neighbouring aquifers.

5.1.2 The coupled model

Numerical experiments were done on two ways of coupling the transport and flow equation. In the first coupling both the flow and transport equation are solved in Matlab. In the second coupling the flow equation is solved with Triwaco and coupled to the transport equation solved in Matlab.

In both cases, first the flow equation is solved with a given density pattern. From the freshwater head h_f calculated by the flow equation, the Darcy velocities q_x, q_y and q_z can be calculated by Darcy's law. These initial velocities are given to the transport equation. The initial condition for the transport equation is given by transformation of the given density pattern into a concentration pattern with an experimentally derived formula. The transport equation determines the new concentration C after some time. This new concentration is transformed with the experimentally derived formula into the density ρ. With this new density, the new freshwater head is calculated with the flow equation. This coupling is called one cycle. A new cycle can be made by repeating the process.

Time

The transport equation is time dependent, the flow equation is steady. Hence the time passing by takes only place in the transport equation. For the transport equation a number of time steps T has to be chosen. Numerical experiments have shown that there is an upper bound for the time passing by in one cycle ($dt \times T$) in cases where the flow is only induced by density differences.

Coupling in Matlab

In this thesis numerical experiments of the rotating brackish zone benchmark problem are presented. The coupling within Matlab, where the flow as well as the transport equation is solved with the Finite Element Method, gives good results. Due to density differences in the domain filled with salt, brackish and freshwater, the brackish zone will rotate until the freshwater is on top of the salt water. This example shows the importance of the coupling between the flow and transport equation. Only using the transport equation with the velocities given by the flow equation (hence using 1 cycle) gives wrong results, it is necessary to keep the time passing by in one cycle ($dt \times T$) bounded.

Although the results are promising, the accuracy of the calculation of the freshwater head should be improved. In the initial Darcy velocities are some unexpected nonzero velocities at the left side of the rotating brackish zone. The system of equations derived for the flow equation with the Standard Galerkin Approach results in a singular matrix. The BiCGSTAB method used to solve this system of equations does find a good solution for the freshwater head, but keeps a relative residual which is not always very small. A possible approach in order to increase the accuracy is using an updated vector for the right hand side of the system of equations.

Coupling with Triwaco

The benchmark problem rotating brackish zone is also solved with a coupling between the flow equation solved with Triwaco and the transport equation solved in Matlab. The initial velocities calculated by Triwaco show some problems. The first problem are the nonzero velocities in the y-direction (q_y) due to the small amount of elements in the y-direction. The second problem in the calculated initial velocities are the wiggles in the maximum and minimum velocities per aquifer in the x- and z-direction (q_x and q_z). The first problem can be handled by using more elements in the y-direction. The second problem can perhaps also be handled by taking more elements in the $x-$ as well as the z-direction.

A bigger problem with the calculation of the velocities appears after 2 cycles. A very small time step τ is used in the transport equation with only 1 time step ($T = 1$), in order to investigate the behaviour of the initial velocities. After those 2 cycles the density has hardly changed, but the velocity in the x-direction increases eight times in some points and becomes negative instead of positive.

The errors in the initial velocity in the y-direction, the wiggles in the initial velocities q_x and q_z and the large error in the velocity q_x after 2 cycles with a very small time step lead to unphysical solutions.

Triwaco models usually contain 3-6 aquifers. For simulation of density dependent groundwater flow more grid points in the vertical direction are needed for a reasonable accuracy. When the number of grid points in the domain increases, the computational time will also increase. For large problems this may cause computer memory and calculation time problems. For the rotating brackish zone example the velocity as well as the concentration is needed on a fine grid. For more practical problems, the density differences are more diffusive. Hence the concentration should be calculated on a fine grid, but the velocities can be calculated on a coarser grid.

Sustainable applications

The freshwater mining example shows the importance of calculating the salt transport when a sink is placed to mine the freshwater which is on top of salt water. After some time the pump will extract brackish water instead of freshwater. Often, these pumps in the dunes in the Netherlands are meant to mine freshwater for drinking water. It is too expensive to purge brackish water, hence the mine is not useful anymore. New simulations can be made with for example a freshwater source with rainwater, which injects the rainwater into the freshwater lens in the dunes. These simulations can be made by only using the transport equation with the velocities calculated by the flow equation. The velocities induced by the sink will overrule the velocities caused by the density differences, hence it is unnecessary to use more cycles.

When the freshwater mining example is extended to a problem that takes place near coast lines, the density dependence of the water for the flow may become important. In the so-called Henry problem (see Appendix H) a given seawater pressure head takes place near the coast line on one side and a constant freshwater flux from the groundwater takes place on the other side of the domain. Freshwater mining often takes place at the top of the domain, for example in the dunes in Scheveningen in the Netherlands. Calculating the migration of salt with the transport equation will give other results than calculating the salt migration with the coupled model. The developed model in this thesis will be able to simulate this problem and can become an important tool for waterworks.

In general, the developed model for the simulation of salt migrations in groundwater can be useful for predicting the effects of the changing climate. The density differences induced by these effects will have an impact on the groundwater flow. With this model, a farmer can indeed know whether ditches bordering his fields become too salt to be used as drinking water for his cattle and the waterworks can change their approach of mining freshwater.

Aim of the study

The aim of this study was to investigate the possibilities of modelling salt migrations in density dependent groundwater with modelling environment Triwaco. The advective, dispersive and diffusive transport of salt with the same grid as used in Triwaco can be calculated with this model. The coupling between the flow equation and transport equation within Matlab works well, even for pure advective transport. Hence when Triwaco is able to calculate the right velocities, no big problems are expected for simulation of density dependent groundwater flow or salt migrations.

5.2 Recommendations

5.2.1 Solute transport

Recommendations with respect to the solute transport model:

1. For the *SUPG classical upwind method*, the representative element distance h in the case that the velocity q_x is negative and q_y is positive or vice versa should be improved.

2. Another parameter in the SUPG classical upwind method to do further research on is the parameter ξ. Research can be done on an algorithm to determine the sign of the inner product of the element distance $\Delta \mathbf{x}$ and the velocity $\mathbf{q}$. This may become important specially in cases where the grid is unstructured and irregular and the velocity not constant.

3. In order to obtain a better accuracy, the coefficients of the *Mizukami Hughes algorithm* in the direction of the vertex zone of node 1 should be improved. The computation time of the Mizukami Hughes algorithm can be reduced by updating the element matrices not every time step. This is only possible when the velocity is not to big.

4. A better accuracy and less numerical diffusion for the 2D simulations of advective transport can be obtained by using a *flux* in the Finite Element Method. For example the flux developed by Kuzmin [30, 31], summerized in the interim Master's thesis [19], may be very useful.

5. In order to reduce the numerical diffusion in the z-direction caused by the Finite Difference Method, the Finite Volume Method *MC limiter* can be used. One dimensional experiments in the Interim Master's thesis [19] show no wiggles and less numerical diffusion for the MC limiter for advective transport. One dimensional implementation is already developed in Matlab.

6. In the numerical experiments the *diffusion* D is a coefficient. When D is the matrix
$$D\nabla C = \begin{bmatrix} D_{xx} & D_{xy} & D_{xz} \\ D_{yx} & D_{yy} & D_{yz} \\ D_{zx} & D_{zy} & D_{zz} \end{bmatrix},$$
the coefficients D_{zx}, D_{zy}, D_{xz} and D_{yz} are neglected because of the combination of the 2D FEM and the 1D FDM. This may cause an asymmetric density profile for diffusive transport. It is recommended to do research on this problem before using the full dispersion matrix.

5.2.2 The coupled model

Recommendations with respect to the coupled model:

1. For the transport equation, the CFL condition is a good criterion to determine an upper bound for the time step τ. It would be nice if it is also possible to determine a criterion for the time step of the coupled process, hence a general criterion for the time passing by during one cycle might be investigated.

2. The use of inner iterations means using the coupling with the steady transport equation until a stable solution appears. These inner iterations can improve the accuracy, but the computational time will increase.

5.2.3 Software in Matlab

Recommendations with respect to the developed software:

1. Matlab can only handle problems with less than 10000 nodes, otherwise an out of memory notification will be given. For realistic problems, often more than 10000 nodes are used. The software can be rewritten in for example Fortran in order to be able to handle bigger problems.

2. A feature that can be useful to implement is the use of Dirichlet boundary conditions. Now only a choice can be made between Neumann and Robbins boundary conditions.

5.2.4 Triwaco

Recommendations with respect to Triwaco:

1. Triwaco has some problems with calculating the velocities when large density contrasts move in the aquifers. Before continuing the research on accurate and fast calculation of density dependent groundwater flow, it is recommended to solve the problem of calculating the velocities of the groundwater flow.

2. The vertical velocity q_z is determined on top and at the bottom of the aquifers. All other parameters, like the horizontal velocities and the densities, are given in the middle of the aquifers. Determination of q_z in the middle of the aquifers will reduce the memory needed for storage.

Bibliography

[1] *http://www.mnp.nl/en/publications/2006/TheeffectsofclimatechangeintheNetherlands.html.* website, August 2006.

[2] G.H.P. Oude Essink. *Impact of Sea Level Rise on Groundwater Flow Regimes.* Ph.D. thesis Delft University of Technology, Delft, 2006.

[3] *http://www.mnp.nl/nl/publicaties/2007/ConclusiesIPCC-rapport.html.* Milieu en Natuur Planbureau, 2007.

[4] *http://knmi.nl.* KNMI, 2007.

[5] *http://www.uwsp.edu/geo/faculty/ritter.* UWSP, 2007.

[6] *http://www.droogtestudie.nl.* Korbee and Hovelynck, 2007.

[7] W. Guo and C.D. Langevin. *User's guide to SEAWAT.* USGS, United States of America.

[8] C. van den Akker and H. Savenije. *Hydrologie 1 CT1310.* Delft University of Technology, Delft, 2006.

[9] M.P. Anderson and W.W. Woessner. *Applied Groundwater Modeling.* Academic Press, California, 1992.

[10] *http://www.seed.slb.com/qa2/FAQView.cfm?ID=561.* SEED Science Center, 2007.

[11] Royal Haskoning. *Triwaco User's manual.* Royal Haskoning, Nederland, 2004.

[12] W.J. Zaadnoordijk. *Variabele dichtheid in het eindige-elementengrondwaterstromingssimulatiepakket TRIWACO op basis van zoet-waterstijghoogten.* CiTG, Delft University of Technology, Delft, 1998.

[13] J. van Kan, A. Segal, and F. Vermolen. *Numerical Methods in Scientific Computing.* VSSD, Delft, 2004.

[14] J. Bear and Y. Bachmat. *Introduction to Modeling of Transport Phenomena in Porous Media.* Kluwer, Dordrecht, 1991.

[15] C. Zheng and G.D. Bennett. *Applied Contaminant Transport Modeling.* Wiley-interscience, United States of America, 2002.

[16] P.J. Stuyfzand. *Hydrochemistry and Hydrology of the Coastal Dune area of the Western Netherlands.* KIWA, Nieuwgein, 1993.

[17] A. Leijnse. *Three-dimensional modeling of couples flow and transport in porous media.* Department of Civil Engineering and Geological Sciences, Indiana, 1992.

[18] A. Segal and C. Vuik. *Lecture notes Computational Fluid Dynamics 2, PhD course JM Burgerscentrum.* Delft University of Technology, Delft, 2004.

[19] E.S. van Baaren. *Interim Master's thesis: Numerical methods for the simulation of salt migration in regional groundwater flow.* Delft/Rotterdam, 2006.

[20] P. Clement and B. de Pagter. *Variationele methoden.* Delft University of Technoloty, Delft, 2003.

[21] A.N. Brooks and T.J.R. Hughes. Streamline upwind/petrov-galerkin formulations for convection dominated flows with particular emphasis on the incompressible navier-stokes equations. *Computer methods in applied mechanics and engineering*, 32:199–259, 1982.

[22] C.W. Oosterlee and C. Vuik. *Scientific Computing.* Delft University of Technology, Delft, 2005.

[23] P. Wesseling. *Elements of computational fluid dynamics.* Delft University of Technology, Delft, 2002.

[24] A. Mizukami. An implementation of the streamline-upwind/petrov-galerkin method for linear triangular elements. *Computer methods in applied mechanics and engineering*, 49.3:357–364, 1985.

[25] A. Mizukami and T.J.R. Hughes. A petrov/galerkin finite element method for convection dominated flows: an accurate upwinding technique for satisfying the maximum principle. *Computer methods in applied mechanics and engineering*, 50:181–193, 1985.

[26] R.J. Levèque. *Finite volume methods for hyperbolic problems.* Cambridge university press, Cambridge, 2002.

[27] R.J. Levèque. *Numerical Methods for Conservation Laws, Lectures Math. Birkhauser Verlag.* ETH Zurich, Zurich, 1992.

[28] D. Braess. *Finite Elements.* Cambridge, United Kingdom, 2001.

[29] M. Bakker and F. Schaars. *The Sea Water Intrusion (SWI) Package Manual.* Artesia and The University of Georgia, Georgia, 2005.

[30] M. Moller D. Kuzmin and S. Turek. *High-resolution FEM-FCT schemes for multidimensional conservation laws.* Institute of Applied Mathematics (LS III), University of Dortmund, Dortmund, 2003.

[31] D. Kuzmin and S. Turek. *High-resolution FEM-TVD schemes based on a fully multidimensional flux limiter.* Institute of Applied Mathematics (LS III), University of Dortmund, Dortmund, 2003.

[32] T.N. Olsthoorn. Variable density groundwater flow modelling with modflow. *Proceedings Salt Water Intrusion Meeting SWIM 1996*, Malmo Sweden, 1996.

Appendix A

List of symbols

Roman symbols

Symbol	Definition	Dimension
a_L	longitudinal dispersivity	$[m]$
a_T	transversal dispersivity	$[m]$
b	upwind test function SUPG	
b_i	constant of the MH algorithm	
C	concentration	$[kg/m^3]$
C_s	solute concentration of water entering	
	from sources or sinks	$[kg/m^3]$
D	dispersion coefficient or matrix	$[m^2/day]$
e_k	element k	
e_l	boundary element l	
f	right hand side vector of system of equations	
g	acceleration due to gravity	$[m/day^2]$
h	representative distance in the element	
	in the direction of $\mathbf{q}$	$[m]$
h_f	freshwater head	$[m]$
k_f	hydraulic conductivity	$[m/day]$
$\dot{m}$	mass flow	$[kg/day]$
M	mass matrix	
$\mathbf{q}$	Darcy velocity	$[m/day]$
p	pressure	$[kg/m * day^2]$
q_{so}	volumetric flow rate per unit volume due to source/sink	$[1/day]$
$q_{so}C_s$	source	$[kg/day * m^3]$
S	stiffness matrix	
$S1$	dispersive stiffness matrix	
$S2$	advective stiffness matrix	
S_s	specific storage	[1/day]
t	time	$[day]$
$\mathbf{v}$	seepage velocity	$[m/day]$
w	classical test function SUPG	
x	spatial coordinate	$[m]$
y	spatial coordinate	$[m]$
z	spatial coordinate	$[m]$

Greek symbols

Symbol	Definition	Dimension
α	Peclet number	[-]
Γ_1	Dirichlet boundary	
Γ_2	Neumann boundary	
Γ_3	Robbins boundary	
$\|\Delta\|$	two times the area of a triangle	$[m^2]$
$\|\tilde{\Delta}\|$	length of boundary element	$[m]$
η	test function numerical method	
θ	porosity	$[-]$
κ	intrinsic permeability	$[m^2]$
μ	dynamic viscosity	$[kg/m * day]$
ρ	density	$[kg/m^3]$
ρ_f	freshwater density	$[kg/m^3]$
τ	time step	$[day]$
ϕ	basis function	
Ω	area	

Appendix B

Definitions

Mathematics

CFL number Named after Courant-Friedrich-Levy. For discretized transport problems, the CFL number determines how many mesh cells, a fluid element passes during a timestep. Or rather, the fraction of a timestep to pass one cell. The CFL condition is a condition for stability of the numberical method. In 2D the CFL condition is

$$|\frac{q_x\tau}{\theta\Delta x}| + |\frac{q_y\tau}{\theta\Delta y}| \leq 1.$$

Consistency Scheme (3.132) is called *consistent* if the local truncation error vanishes as $\tau \downarrow 0$ for all smooth functions $C(\mathbf{x}, t)$ satisfying the differential equation.

Convergence The method is *convergent* at time T in the norm $||.||$ if

$$lim_{\tau\to 0, N\tau=T}||E^N|| = 0.$$

Here N is used to indicate the time level corresponding to time $T = N\tau$.

Essential boundary condition Condition that has to be satisfied by all functions in the function class where the solution is sought. $C|_{\Gamma_1} = g_1(\mathbf{x})$ is the essential boundary condition.

Global truncation error The *global truncation error* is defined as

$$E^n \equiv C^n - C(n),$$

with C^n the numerical solution at $t = n$ and $C(n)$ the exact solution at $t = n$.

Local truncation error Scheme (3.132) can be written as $C^{n+1} = \mathcal{N}(C^n)$, where $\mathcal{N}$ represents the numerical operator mapping the approximate solution at one time step to the approximate solution at het next. The *local truncation error* is defined as

$$e^n = \frac{1}{\tau}[\mathcal{N}(C^n) - C^{n+1}].$$

Natural boundary condition Condition that appears naturally from the minimization problem once the corresponding Euler-Lagrange equations are derived. $((\theta D \nabla C) \cdot \mathbf{n})\,|_{\Gamma_2} = g_2(\mathbf{x})$ and $(\sigma C + (\theta D \nabla C \cdot \mathbf{n}))\,|_{\Gamma_3} = g_3(\mathbf{x})$ are the natural boundary conditions.

Solenoidal vector field In vector calculus a solenoidal vector field is a vector field $\mathbf{v}$ with divergence zero:

$$\nabla \cdot \mathbf{v} = 0.$$

This condition is satisfied whenever $\mathbf{v}$ has a vector potential. A vector potential is a vector field whose curl is a given vector field. The curl is a vector operator that shows a vector field's rate of rotation: the direction of the axis of rotation and the magnitude of rotation. Here rotation is used for properties of a vector function or position (they are nog about changes with time).

Stability A method is said to be *stable* if a small deviation from the true solution does not tend to grow as the solution is iterated.

Let $\{\delta_n, n = 0, 1, ..., N\}$ and $\{\delta_n^*, n = 0, 1, ..., N\}$ be any two perturbations of the discretized problem and let $\{\tilde{C}_n, n = 0, 1, ..., N\}$ and $\{\tilde{C}_n^*, n = 0, 1, ..., N\}$ be the resulting perturbed solutions. Then if there exist positive constant S and Δx_0 such that, for all $\Delta x \in (0, \Delta x_0]$:

$$||\tilde{C}_n - \tilde{C}_n^*|| \leq S\epsilon,$$

whenever

$$||\delta_n - \delta_n^*|| \leq \epsilon, \qquad 0 \leq n \leq N,$$

then the method is said to be zero-stable.

Hydrology

Advection Transportation of contaminants by the flow of a current of water. This implies that the solute contaminant moves passively with the same velocity as the groundwater.

Anisotropic Not possessing the same properties in all directions (the opposite of isotropic).

Aquifer An aquifer is a body of rock or sediment that is sufficiently porous and permeable to store, transmit and yield significant quantities of groundwater to wells and springs.

Aquitard An aquitard is a geologic formation that is not permeable enough to yield significant amounts of water to wells, but on a regional scale can supply significant water to the underlying or overlaying aquifers. In an aquitard only vertical velocity is assumed, the horizontal velocity of the flow is zero.

Convection Convection is the internal movement of currents within fluids (i.e. liquids and gases). It cannot occur in solids due to the atoms not being able to flow freely. Convection may cause a related phenomenon called advection, in which mass or heat is transported by the currents or motion in the fluid.

Diffusion The transport of matter solely by the random motions of individual molecules not moving together in coherent groups. It is a consequence of concentration gradients.

Dispersion Dispersive transport describes the dilution or mixing of a solute due to different velocities of groundwater, which is moving at rates that are both greater and smaller than the average advective pore velocity. Dispersion is observed on both the microscopic and the macroscopic scale. The three main reasons for the different velocities at the microscopic scale are friction in pores, varying travel path lengths and pore sizes. Macroscopic dispersion is caused by variable permeability's of single layers inducing different velocities.

Fick's law The relation between the flux $\mathbf{F}$ and the concentration C is known as Fick's law:

$$\mathbf{F} = -D\nabla C,$$

with D the diffusion coefficient or matrix.

freshwater head The measured head if the piezometer tube were filled over its full height with water of specific weight.

Gauge pressure Pressure measured greater than atmospheric pressure.

Hydraulic gradient Hydraulic head drop between two points a and b divided by the distance between them.

Hydraulic head Measure for the amount of energy groundwater flowing through aquifer has per unit weight. Quantity is expressed in terms of a length of water.

Hydrostatic pressure The pressure which is exerted on a portion of a column of fluid as a result of the weight of the fluid above it.

Permeability The ability of a geologic formation to transmit water.

Phreatic The term phreatic is used in geology to refer to matters relating to underground water below the water table.

Phreatic zone The layer(s) of soil or rock below the water table in which voids are permanently saturated with water, as opposed to the higher vadose zone in which the pore spaces are not completely filled with water.

Piezometer A device used for the measurement of hydraulic head of groundwater in aquifers.

Porosity The percentage of the volume of that material that can be occupied by water.

Pressure head Same as gauge pressure, unless absolute pressure is explicitly specified

Saturation Generally means water content is equal to porosity and pressure head is greater than atmospheric pressure. / The relative amount of water, oil and gas in the pores of a rock, usually as a percentage of volume

Seepage velocity Percolation of water through the soil from unlined canals, ditches, laterals, watercourses, or water storage facilities.

Specific Storage The amount of water which a given volume of aquifer will produce, provided a unit change in hydraulic head is applied to it (while it still remains fully saturated). it has units of inverse length, [L-1].

Transient Varying in time.

Transmissiviteit The rate at which water passes through an aquifer.

Water table or phreatic surface The upper limit of abundant groundwater. The surface where the pressure head is equal to atmospheric pressure.

Appendix C

Software

The during this research developed software was written in Matlab. The transport equation is three dimensionally solved with several numerical methods. Matlab functions are made in order to automatically read and write files with information written or read by Triwaco in order to have a continuous coupling between both programmes.

The coupling between the transport equation and the flow equation within Matlab is two dimensionally.

C.1 Structure of the software

Main

Main makes use of the following functions in Matlab:

- EAD
- EbAD
- InitialCondition
- FlowEquation if coupling within Matlab is used
- SaveRho if coupling with Triwaco is used
- TransportEquation
- rhof
- VADf

TransportEquation

TransportEquation makes use of the following functions in Matlab:

- MADf
- S1ADf
- S2ADf
- fADf
- D1ADf
- D2ADf

- MADUf
- S1ADUf
- S2ADUf
- MADMHf
- S2ADMHf
- MADPUf
- S2ADPUf

MADUf, S1ADUf and S2ADUf

MADUf, S1ADUf and S2ADUf make use of the following function in Matlab:

- hf

VADf

VADf makes use of the following function in Matlab:

- litcount

FlowEquation

FlowEquation makes use of the following functions in Matlab:

- Tf
- fhff

Plaatjesmaker

C.2 Description of all functions in Matlab

D1ADf FDM for the third dimension of the transport equation, dispersive part.

D2ADf FDM for the third dimension of the transport equation, advective part.

dos dos('run3.bat') runs the Fortran file in Triwaco.

EAD constructs the matrix with all vertices per element.

EbAD constructs the matrix with all vertices per boundary element.

fADf constructs the right hand side of the system of equations for the transport equation.

fADUf constructs the right hand side of the upwind method for the transport equation.

fhff constructs the right hand side for the flow equation.

FlowEquation calculates the freshwater heads by solving the flow equation with the BiCGSTAB method and uses Darcy's law to determine the velocities of the flow.

hf determines the representative distance in the element in the direction of **q** by the Segal algorithm.

litcount reads the adore blocks of the *.ado, *.TRO, *.TEO files constructed by Triwaco.

MADf constructs the mass matrix for the SGA for the transport equation.

MADMHf constructs the mass matrix for the Mizukame Hughes algorithm for the transport equation.

MADPUf constructs the mass matrix for the SUPG pure advection method for the transport equation.

MADUf constructs the mass matrix for the SUPG method for the transport equation.

Main Mainprogramma with cycles of the coupled system. Input: number of elements *ne*, number of boundary elements *neb*, number of vertices per aquifer *nv* and number of aquifers *nl*, time step *dt*, number of time steps transport equation *It*. A choice has to be made between the methods SGA, SUPG classical upwind, Mizukami Hughes algorithm and the SUPG pure advection method by Mizukami.

plaatjesmaker makes figures after the simulation is completed.

rhof converts concentration into density.

S1ADf constructs the stiffness matrix elementwise for the SGA for the dispersive part of the transport equation.

S1ADUf constructs the stiffness matrix elementwise for the SUPG for the dispersive part of the transport equation.

S2ADf constructs the stiffness matrix elementwise for the SGA for the advective part of the transport equation.

S2ADMHf constructs the stiffness matrix elementwise for the Mizukami Hughes algorithm for the advective part of the transport equation.

S2ADPUf constructs the stiffness matrix elementwise for the SUPG pure advection algorithm for the advective part of the transport equation.

S2ADUf constructs the stiffness matrix elementwise for the SUPG for the advective part of the transport equation.

Tf constructs the matrix for the flow equation.

TransportEquation solves the advection-dispersion equation for 1 iteration, the system of equations formed for the transport equation is solved with the BiCGSTAB method.

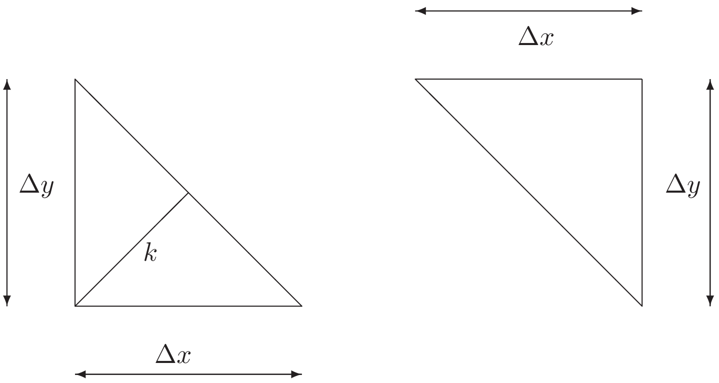
VADf reads the information per vertex and stores this information in a matrix.

Appendix D

Representative element distance

Algorithm to determine h_j, the representative distance of an element in the direction of $\mathbf{q}$:

```
For each vertex $i$ within the element
        $\phi_{max} = \max_k \left(|\mathbf{q}_i| \cdot \nabla\phi_k|\right)$
        if $\phi_{max} > \epsilon$
                $h(i) = \frac{||\mathbf{q}_i||}{\phi_{max}}$
        else
                $h(i) = 0.$
```



(a) (b)

Figure D.1: Two elements of the Finite Element grid, with element distances Δx and Δy.

The gradients of the basis functions are given by

$$\nabla\phi_1(\mathbf{x}) = \begin{bmatrix} \frac{\partial\phi_1(\mathbf{x})}{\partial x} \\ \frac{\partial\phi_1(\mathbf{x})}{\partial y} \end{bmatrix} = \begin{bmatrix} \frac{y_3-y_2}{\Delta} \\ \frac{x_2-x_3}{\Delta} \end{bmatrix} = \begin{bmatrix} \frac{1}{\Delta x} \\ \frac{1}{\Delta y} \end{bmatrix}, \tag{D.1}$$

$$\nabla\phi_2(\mathbf{x}) = \begin{bmatrix} \frac{\partial\phi_2(\mathbf{x})}{\partial x} \\ \frac{\partial\phi_2(\mathbf{x})}{\partial y} \end{bmatrix} = \begin{bmatrix} \frac{y_1-y_3}{\Delta} \\ \frac{x_3-x_1}{\Delta} \end{bmatrix} = \begin{bmatrix} \frac{-1}{\Delta x} \\ 0 \end{bmatrix}, \tag{D.2}$$

$$\nabla\phi_3(\mathbf{x}) = \begin{bmatrix} \frac{\partial\phi_3(\mathbf{x})}{\partial x} \\ \frac{\partial\phi_3(\mathbf{x})}{\partial y} \end{bmatrix} = \begin{bmatrix} \frac{y_2-y_1}{\Delta} \\ \frac{x_1-x_2}{\Delta} \end{bmatrix} = \begin{bmatrix} 0 \\ \frac{-1}{\Delta y} \end{bmatrix}. \tag{D.3}$$

Different possible velocities $\mathbf{q}$ are considered for Figure D.1(a), for the above algorithm:

$$\mathbf{q} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \text{ leads to } \phi_{max} = \tfrac{1}{\Delta x}, \text{ hence } h(i) = \Delta x. \tag{D.4}$$

$$\mathbf{q} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \text{ leads to } \phi_{max} = \tfrac{1}{\Delta y}, \text{ hence } h(i) = \Delta y. \tag{D.5}$$

$$\mathbf{q} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \text{ leads to } \phi_{max} = 0, \text{ hence } h(i) = 0. \tag{D.6}$$

$$\mathbf{q} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \text{ leads to } \phi_{max} = \tfrac{1}{\Delta x} + \tfrac{1}{\Delta y}, \text{ hence } h(i) = \tfrac{\sqrt{2(\Delta x+\Delta y)}}{\Delta x+\Delta y}. \tag{D.7}$$

$$\mathbf{q} = \begin{bmatrix} 1 \\ 2 \end{bmatrix} \text{ leads to } \phi_{max} = \tfrac{1}{\Delta x} + \tfrac{2}{\Delta y}, \text{ hence } h(i) = \tfrac{\sqrt{5}}{1/\Delta x+2/\Delta y}. \tag{D.8}$$

In the case of Equation (D.7), when $\Delta x = \Delta y$,

$$h(i) = \frac{1}{2}\sqrt{2}\Delta x.$$

This is the distance k in Figure D.1(a). In the case of Equation (D.8)

$$h(i) = \frac{\sqrt{5}}{3}\Delta x,$$

when $\Delta x = \Delta y$, this distance can also be derived with simple goniometric formulas.

It is noted that the sign of $\mathbf{q}$ has no influence on $h(i)$, hence the representative distance of the element in the direction of $\mathbf{q}$ is for $\mathbf{q} = [1\ 1]$ equal to the representative distance for $\mathbf{q} = [1\ -1]$. When $\Delta x = \Delta y$ and $\mathbf{q} = [1\ -1]$, the element distance in the element in Figure D.1 should be $\sqrt{2}\Delta x$ instead of $\frac{1}{2}\sqrt{2}\Delta x$.

Appendix E

Triwaco

E.1 Groundwater flow equation

For constant density, the groundwater flow equation in Triwaco is two dimensional. When the density becomes dependent on the location, the height of the aquifer becomes important. Z, the elevation or height of the aquifer is now introduced which depends on the x and y coordinate.

Darcy's law in terms of the freshwaterhead h_f for a coordinate x_i can be written as:

$$q_i = -k_i \left(\frac{\partial h_f}{\partial x_i} + \frac{\rho - \rho_f}{\rho_f} \frac{\partial z}{\partial x_i} \right). \tag{E.1}$$

With k_i again the freshwater hydraulic conductivity and ρ_f the freshwater density. The Dupuit-assumption allows to express Darcy's law for vertical flow through aquitards and vertically integrated horizontal flow in aquifers.

E.1.1 Vertical flow

The vertical flow in Triwaco is solved with the Finite Difference Method. Define

$$k_i = \frac{K_i \rho_f g}{\mu} = \frac{1}{c_i},$$

where g is the acceleration due to gravity, μ is the dynamic viscosity of water and c_j the resistance of aquitard j. The *vertical* flow from aquifer j with freshwater head h_{f_j} at elevation Z_j in the center of the aquifer *through aquitard* $j-1$ with thickness d_{j-1} and vertical intrinsic permeability K_{j-1} to aquifer $j-1$ with freshwater head $h_{f_{j-1}}$ at elevation Z_{j-1} in the center of the aquifer is equal to:

$$\begin{aligned} q_{z,j-1} &= -k_{j-1} \left(\frac{\partial h_f}{\partial x_{j-1}} + \frac{\rho}{\rho_f} \frac{\partial z}{\partial x_{j-1}} - \frac{\partial z}{\partial x_{j-1}} \right). \\ &\cong -k_{j-1} \left(\frac{-h_{f_j} + h_{f_{j-1}}}{d_{j-1}} + \frac{\int_{z=Z}^{Z_{j-1}} \frac{\rho}{\rho_f} dz}{d_{j-1}} + \frac{Z_j - Z_{j-1}}{d_{j-1}} \right), \end{aligned} \tag{E.2}$$

or

$$q_{z,j-1} \cong \frac{h_{f_j} - h_{f_{j-1}} - Z_j + Z_{j-1} - \int_{z=Z_j}^{Z_{j-1}} \frac{\rho}{\rho_f} dz}{c_{j-1}}. \tag{E.3}$$

With c_{j-1} the resistance of aquitard $j-1$. Note that $\frac{\partial h_f}{\partial z}=0$ within an aquifer because of the hydraulic pressure within an aquifer. The hydraulic pressure is defined as the pressure which is exerted on a portion of a column of fluid as a result of the weight of the fluid above it. So in the application of the finite difference method the used grid size is d_j. More information and the derivation of the integral $\frac{\int_{z=Z_j}^{Z_{j-1}}\frac{\rho}{\rho_f}dz}{d_{j-1}}$ can be found in Olsthoorn [32].

E.1.2 Horizontal flow

Let Q_i denote the horizontal flow in the aquifer ($[L^2T^{-1}]$). The *horizontal* flow *in aquifer* i with thickness H is equal to:

$$\begin{aligned} Q_i &= \int_{z=Z-1/2H}^{Z+1/2H} q_i dz \\ &= \int_{z=Z-1/2H}^{Z+1/2H}\left(-k_i\frac{\partial h_f}{\partial x_i}-k_i\frac{\rho-\rho_f}{\rho_f}\frac{\partial Z}{\partial x_i}\right)dz \\ &= -\kappa_i\int_{z=Z-1/2H}^{Z+1/2H}\frac{\partial h_f}{\partial x_i}dz - -\kappa_i\frac{\partial Z}{\partial x_i}\int_{z=Z-1/2H}^{Z+1/2H}\frac{\rho-\rho_f}{\rho_f}dz, \end{aligned} \tag{E.4}$$

where Z denotes the center of the aquifer and the index $i=1,2$ indicates the x and y coordinates.

Remark: In the second step of Equation (E.4) is z replaced by Z without any comment. It is unknown why this is permitted, it might be possible that z in Equation (E.1) must be Z.

Define the transmissivity as $T=kH$ with H the thickness of the aquifer and rewrite Equation (E.4) as:

$$Q_i = -\frac{T_i}{H}\int_{z=Z-1/2H}^{Z+1/2H}\frac{\partial h_f}{\partial x_i}dz - \frac{T_i}{H}\frac{\partial Z}{\partial x_i}\int_{z=Z-1/2H}^{Z+1/2H}\frac{\rho-\rho_f}{\rho}dz. \tag{E.5}$$

With S the storage coefficient and q the sink term, the equation of continuity becomes:

$$\frac{\partial Q_1}{\partial x_1}+\frac{\partial Q_2}{\partial x_2} = q_{z,j}-q_{z,j-1}-S\frac{\partial h_f}{\partial t}-q. \tag{E.6}$$

Substitution of Equation (E.3) and (E.4) in Equation (E.6) results in:

$$\begin{aligned} &\frac{\partial}{\partial x_1}\left(\frac{T_1}{H}\int_{z=Z-1/2H}^{Z+1/2H}\frac{\partial h_f}{\partial x_1}dz\right)+\frac{\partial}{\partial x_2}\left(\frac{T_2}{H}\int_{z=Z-1/2H}^{Z+1/2H}\frac{\partial h_f}{\partial x_2}dz\right)= \\ &\quad = -\frac{h_{f_{j+1}}-h_f-\int_{z=Z_{j+1}}^{Z}\frac{\rho}{\rho_f}dz}{c_j}-\frac{h_f-h_{f_{j-1}}-\int_{z=Z}^{Z_{j-1}}\frac{\rho}{\rho_f}dz}{c_{j-1}} \\ &\quad - S\frac{\partial h_f}{\partial t}-q-q^*. \end{aligned} \tag{E.7}$$

With q^* the correction flux:

$$q^* = -\frac{\partial}{\partial x_1}\left(\frac{T_1}{H}\frac{\partial Z}{\partial x_1}\int_{z=Z-1/2H}^{Z+1/2H}\frac{\rho-\rho_f}{\rho_f}dz\right)-\frac{\partial}{\partial x_2}\left(\frac{T_2}{H}\frac{\partial Z}{\partial x_2}\int_{z=Z-1/2H}^{Z+1/2H}\frac{\rho-\rho_f}{\rho_f}dz\right)+$$

$$+\frac{Z_{j+1}-Z-\int_{z=Z_{j+1}}^{Z}\frac{\rho}{\rho_f}dz}{c_j}-\frac{Z-Z_{j-1}-\int_{z=Z}^{Z_{j-1}}\frac{\rho}{\rho_f}dz}{c_{j-1}}. \tag{E.8}$$

First part of E.7 It has linear shaped functions and numerical calculations based on Galerkin's method. It is assumed that the density is constant in the vertical direction within each aquifer. Equation (E.7) can now be simplified. The density within an aquifer is called ρ, the density in the underlying aquitard is called γ_j and in the above aquitard γ_{j-1}. With d_j the thickness of aquitard j will be denoted and with H_j the thickness of aquifer j. The correction flux can be rewritten

$$\begin{aligned} q^* = & -T_1\frac{\partial^2 Z}{\partial x_1^2}\frac{\rho-\rho_f}{\rho_f}-T_1\frac{\partial Z}{\partial x_1}\frac{\partial(\rho/\rho_f)}{\partial x_1}-\frac{\partial T_1}{\partial x_1}\frac{\partial Z}{\partial x_1}\frac{\rho-\rho_f}{\rho_f} \\ & -T_2\frac{\partial^2 Z}{\partial x_2^2}\frac{\rho-\rho_f}{\rho_f}-T_2\frac{\partial Z}{\partial x_2}\frac{\partial(\rho/\rho_f)}{\partial x_2}-\frac{\partial T_2}{\partial x_2}\frac{\partial Z}{\partial x_2}\frac{\rho-\rho_f}{\rho_f} \\ & +\frac{Z_{j+1}-Z+\frac{1}{2}H_{j+1}\frac{\rho_{j+1}}{\rho_f}+d_j\frac{\gamma_j}{\rho_f}+\frac{1}{2}H_j\frac{\rho}{\rho_f}}{c_j} \\ & -\frac{Z-Z_{j-1}+\frac{1}{2}H\frac{\rho}{\rho_f}+d_{j-1}\frac{\gamma_{j-1}}{\rho_f}+\frac{1}{2}H_{j-1}\frac{\rho_{j-1}}{\rho_f}}{c_{j-1}}. \end{aligned} \tag{E.9}$$

E.1.3 FEM for the correction flux

The correction flux of Equation (E.9) is discretized with the Finite Element Method. First, integrate the flux (E.9) over the surface A:

$$Q^* = \int\int_A q^* dx_1 dx_2 \tag{E.10}$$

Split the correction flux into a flux that takes care for the lateral effects within the aquifer Q_l^* and a flux that takes care for the vertical effects to the underlying and above aquifers Q_v^*:

$$\begin{aligned} Q_l^* = \int\int_A \Bigg\{ & -T_1\frac{\partial^2 Z}{\partial x_1^2}\frac{\rho-\rho_f}{\rho_f}-T_1\frac{\partial Z}{\partial x_1}\frac{\partial(\rho/\rho_f)}{\partial x_1}-\frac{\partial T_1}{\partial x_1}\frac{\partial Z}{\partial x_1}\frac{\rho-\rho_f}{\rho_f} \\ & -T_2\frac{\partial^2 Z}{\partial x_2^2}\frac{\rho-\rho_f}{\rho_f}-T_2\frac{\partial Z}{\partial x_2}\frac{\partial(\rho/\rho_f)}{\partial x_2}-\frac{\partial T_2}{\partial x_2}\frac{\partial Z}{\partial x_2}\frac{\rho-\rho_f}{\rho_f}\Bigg\} dx_1 dx_2. \end{aligned} \tag{E.11}$$

$$\begin{aligned} Q_v^* = \int\int_{A_e} \Bigg\{ & \frac{Z_{j+1}-Z+\frac{1}{2}H_{j+1}\frac{\rho_{j+1}}{\rho_f}+d_j\frac{\gamma_j}{\rho_f}+\frac{1}{2}H_j\frac{\rho}{\rho_f}}{c_j} \\ & -\frac{Z-Z_{j-1}+\frac{1}{2}H\frac{\rho}{\rho_f}+d_{j-1}\frac{\gamma_{j-1}}{\rho_f}+\frac{1}{2}H_{j-1}\frac{\rho_{j-1}}{\rho_f}}{c_{j-1}}\Bigg\} dx_1 dx_2. \end{aligned} \tag{E.12}$$

First, the lateral flux is described. Take a triangular shaped element e and assume that the parameters are linear within the element:

$$T_i = T_{i,1}^e x_1 + T_{i,2}^e x_2 + T_{i,0}^e, \tag{E.13}$$

$$Z = Z_1^e x_1 + Z_2^e x_2 + Z_0^e, \tag{E.14}$$

$$\rho = \rho_1^e x_1 + \rho_2^e x_2 + \rho_0^e. \tag{E.15}$$

The second order derivative of Z, $\frac{\partial^2 Z}{\partial x_i^2} = 0$, due to Equation (E.14), so Equation (E.11) simplifies. Call the remaining part the element flux Q_e^*:

$$Q_e^* = -A_e \left\{ T_{1\mu}^e Z_1^e \frac{\rho_1^e}{\rho_f} + T_{1,1}^e Z_1^e \frac{\rho_\mu^e - \rho_f}{\rho_f} + T_{2\mu}^e Z_2^e \frac{\rho_2^e}{\rho_f} + T_{2,2}^e Z_2^e \frac{\rho_\mu^e - \rho_f}{\rho_f} \right\}, \tag{E.16}$$

with μ the mean of the three vertices of the element and A_e the surface of the element. During the linearization of the height Z in Equation (E.14), the second order derivative is neglected. The corresponding term of Equation (E.11) can be important and has to be added:

$$Q_z^* = \int \int_A \left\{ -T_1 \frac{\partial^2 Z}{\partial x_1^2} \frac{\rho - \rho_f}{\rho_f} - -T_2 \frac{\partial^2 Z}{\partial x_2^2} \frac{\rho - \rho_f}{\rho_f} \right\} dx_1 dx_2. \tag{E.17}$$

The flux Q_z^* has to be calculated for each vertex. The number of neighboring vertices has to be determined for each vertex (≥ 2). Dependent on the number and location of the vertices, it is possible to determine 0, 1 or 2 curvatures. The Laurent- series in the local coordinates ξ and η around the central vertex parallel to x_1 and x_2 as explained in [12] shows the number of curvatures

$$Z \simeq Z_0 + Z_1 \xi + Z_2 \eta + \frac{1}{2} Z_{11} \xi^2 + Z_{12} \xi \eta + \frac{1}{2} Z_{22} \eta^2, \tag{E.18}$$

where Z_0 is the value of the central vertex, Z_1 and Z_2 are the slopes, Z_{12} the cross-term and Z_{11} and Z_{22} the curvatures: $Z_{ii} = \partial^2 Z / \partial x_i^2$. The definition of the Laurent series can be found in Appendix B. If there are more than five neighboring vertices, the terms can be determined with the Mean Square Error. The Z-curvature flux becomes

$$Q_z^* = -A_n \frac{\rho - \rho_f}{\rho_f} \left\{ T_1 Z_{11} + T_2 Z_{22} \right\}. \tag{E.19}$$

And the lateral flux becomes

$$Q_l^* = \sum \left(\frac{1}{3} Q_e^* \right) + Q_z^*. \tag{E.20}$$

The vertical correction flux (E.12) can be calculated for each vertex:

$$Q_v^* = A_n \left\{ \frac{Z_{j+1} - Z + \frac{1}{2} H_{j+1} \frac{\rho_{j+1}}{\rho_f} + d_j \frac{\gamma_j}{\rho_f} + \frac{1}{2} H_j \frac{\rho}{\rho_f}}{c_j} - \frac{Z - Z_{j-1} + \frac{1}{2} H \frac{\rho}{\rho_f} + d_{j-1} \frac{\gamma_{j-1}}{\rho_f} + \frac{1}{2} H_{j-1} \frac{\rho_{j-1}}{\rho_f}}{c_{j-1}} \right\}, . \tag{E.21}$$

with A_n the surface of the vertex. The total correction flux can now be calculated by

$$Q_c^* = Q_v^* + Q_l^*. \tag{E.22}$$

E.1.4 FEM for the flow equation

The discretization of the correction flux q^* is explained in the previous section. The other terms of Equation (E.7) are also discretized with the finite element method. The matrices and vectors belonging to these terms can only be found in Triwaco's source code.

E.1.5 Particle tracking

In [19] more information can be found about particle tracking. Once the groundwater flow situation has been calculated for a given hydrogeological situation, groundwater flow lines may be computed using the particle-tracking program Trace. Trace determines pathlines and travel times in groundwater flow, based on groundwater flow simulations.

The horizontal movement is derived from the discharges that have been calculated for the aquifers. The thickness and the porosity of the aquifer are used to calculate the velocity corresponding to the discharge. The vertical movement within the aquifers is derived from vertical fluxes through the aquitards, using the principle of continuity. Vertical movement is been taken into account for the slope of the aquifer. The transport through the aquitards is vertical only. Together with the porosity and thickness of the aquitard, the time needed for the passage through the aquitard is calculated. The pathlines can be determined following the flow upstream and downstream.

Appendix F

Temporal discretization 1D

For the steady 1D advection dispersion equation the condition $|p_h| \leq 1$ is needed to have a monotone solution. p_h is called the *mesh Péclet number* and is defined as

$$p_h \equiv \frac{Pe\Delta x}{2\theta} \equiv \frac{q\Delta x}{2\theta D}. \tag{F.1}$$

Pe is called the Péclet number and is a measure for by how much the advection dominates the dispersion.

For the time dependent advection dispersion equation another analysis is needed that can give a stability condition.

F.1 Amplification factors

Recall the spatial discretized advection dispersion equation of the form $M\frac{dC}{dt} = SC + f$ with M the mass matrix, S the stiffness matrix and f the source term. Each numerical procedure has an amplification matrix G which is given by the numerical solution of the error equation $\frac{d\epsilon}{dt} = M^{-1}S\epsilon$:

$$\epsilon^{n+1} = G(\tau M^{-1}S)\epsilon^n. \tag{F.2}$$

A numerical solution method is absolutely stable if and only if for the eigenvalues μ_k of $G(\tau M^{-1}S)$ holds $|\mu_k| < 1$. If the error equation consists of one equation only, i.e. $\epsilon' = \lambda\epsilon$, then the amplification of the numerical solution is referred to as the amplification factor, which is denoted by $V(\tau\lambda)$. The eigenvalues μ_k of $G(\tau M^{-1}S)$ are obtained by substitution of the eigenvalues λ_k of the matrix $M^{-1}S$ into the amplification factor

$$\mu_k = V(\tau\lambda_k).$$

Hence for stability we need

$$|V(\tau\lambda_k)| < 1. \tag{F.3}$$

Note that all eigenvalues λ_k are real-valued and negative ($\lambda < 0$) when S is negative definite and M is positive definite (see Section 10.5 [13]). The amplification matrix for the $\omega-$method is:

$$G(\tau M^{-1}S) = \left(I - \omega\tau M^{-1}S\right)^{-1}\left(I + (1-\omega)\tau M^{-1}S\right).$$

With this theory it is hard to derive a stability condition for methods that solve the equation $M\frac{dC}{dt} = SC + f$ because the eigenvalues of the matrix $M^{-1}S$ have to be calculated. Though it can be used to say something about the boundedness of the error. For Forward Euler, $\omega = 0$:

$$|V(\tau\lambda)| = |1 + \tau\lambda| \to \infty \text{ as } |\lambda| \to \infty.$$

Provided $\lambda \in \mathbb{R}$, the interval for stability for Forward Euler can be calculated by using Equation (F.3):

$$\tau|\lambda| \leq 2 \tag{F.4}$$

For Backward Euler

$$|V(\tau\lambda)| = |\frac{1}{1 - \tau\lambda}| \to 0 \text{ as } |\lambda| \to \infty,$$

and the interval for stability is unbounded:

$$\tau\lambda \in (-\infty, 0) \tag{F.5}$$

and for Crank-Nicholson, $\omega = 1/2$, the amplification factor is

$$|V(\tau\lambda)| = |\frac{1 + \frac{\tau\lambda}{2}}{1 - \frac{\tau\lambda}{2}}| \to 1 \text{ as } |\lambda| \to \infty.$$

and the interval for stability of this explicit method is again (F.5). For the Modified Euler method (Runge-Kutta-2) the amplification factor is given by

$$|V(\tau\lambda)| = |1 + \tau\lambda + \frac{1}{2}(\tau\lambda)^2|,$$

and the interval for stability is given by

$$\tau|\lambda| \leq 2. \tag{F.6}$$

So for Forward Euler the error does not extinguish and can become large outside the small interval for stability. Backward Euler and Crank-Nicholson are unconditionally stable, but only for Backward Euler errors in time in the initial condition will always be damped out. For Crank-Nicholson the error of the previous time steps is bounded but does not extinguish for $|\lambda|$ large. The Runge-Kutta-2 scheme has a stability condition that is better than the stablity condition for Forward Euler. [13]

F.2 Stability temporal discretization scheme

As an alternative method to estimate the eigenvalues of the matrix $M^{-1}S$, Von Neumann analysis can be used. More information can be found in Chapter 8 in [26] or in Chapter 4 in [23]. In [23] the following results are obtained for the advection-dispersion equation with the $\omega-$scheme: unconditional stability for $1/2 \leq \omega \leq 1$. So the Backward Euler and Crank-Nicholson schemes are unconditionally stable.

For $\omega = 0$ in the $\omega-$scheme (Forward Euler) the necessary and sufficient stability condition according to [23] is:

$$2D\tau\left(\frac{1}{\Delta x^2}\right) \leq 1 \text{ and } \frac{\tau}{2D}\left(\frac{q^2}{1+|q|\Delta x}\right) \leq 1. \tag{F.7}$$

This is a disadvantage of the Forward Euler method, the time step is restricted in order to get a stable solution.

For the advection equation spatial discretized with the first order upwind method with positive velocity q and temporal discretized with Forward Euler, the Von Neumann stability analysis results in the stability condition:

$$0 \leq \frac{q\tau}{\Delta x} \leq 1. \tag{F.8}$$

The derivation can be found in Chapter 8 of Leveque [26]. $\frac{q\tau}{\Delta x}$ is known as the Courant number. The same condition is derived in Section 12.3 [13] for the 1D advection equation discretized with Forward Euler and central differences.

For the 1D dispersion equation according to [23] the time step after discretization with Forward Euler must satisfy

$$\tau \leq \frac{\Delta x^2}{2D}. \tag{F.9}$$

This is the reason why explicit methods are less suitable for the dispersion part of the advection dispersion equation.

Temporal discretization with Forward Euler for the advective part and Backward Euler for the dispersive part results in the condition $|q\tau/dx| \leq 1$, because the dispersive part discretized with Backward Euler is unconditionally stable.

For the Runge-Kutta-2 method the stability conditions are:

$$|\frac{q\tau}{\Delta x}| \leq 1, \qquad \frac{D\tau}{\Delta x^2} \leq \frac{1}{2}. \tag{F.10}$$

F.3 TVD methods

For nonlinear numerical methods, like the high resolution method MC-limiter of the finite volume method a different approach for stability must be adopted. The total variation (TV) turns out to be an effective tool for studying stability of nonlinear problems. In Section 8.3.5 in Leveque [26] it can be seen that the high resolution TVD method MC limiter is convergent for the advection equation provided the CFL condition is satisfied:

$$|\frac{q\tau}{\Delta x}| \leq 1. \tag{F.11}$$

The methods: Fromm (3.120), Beam-Warming (3.121) and Lax-Wendroff (3.122) are not TVD methods and hence not necessary monotonicity preserving (see Section 6.7 [26]). The first order upwind FVM is TVD for the advection equation, so this method for this equation cannot introduce oscillations.

F.4 Numerical experiments

Three different temporal discretizations wer used in the Interim Master's thesis: $T1$ is the temporal discretization that refers to the use of Backward Euler for the dispersion part and Forward Euler for the advection part and the source term. $T2$ is the temporal discretization that refers to the use of the Crank-Nicolson scheme. $T3$ is the temporal discretization that refers to the use of the Runge-Kutta-2 scheme.

Note that $T3$ does not have to be stable for the given parameters for the advection dispersion equation. $T3$ costs 2 calculation per time step, so in order to obtain the same computer work take a double step size for $T3$, $\Delta x = 0.2$ and take a look after 50 instead of 100 time steps. The results for the advection-dispersion equation for the FEM, FVM and FDM can be found in the Figures F.1, F.2, F.3. In Figure F.4 the results can be found for the advection equation. It can be seen that $T1$ has less numerical dispersion than $T3$ and $T3$ less than $T2$.

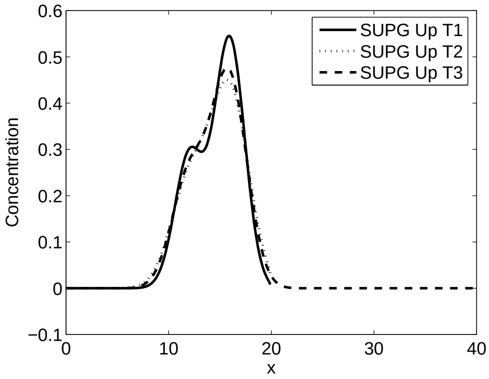


Figure F.1: SUPG classical upwind with $T1$, $T2$ with $\Delta x = 0.1$ and $T3$ with $\Delta x = 0.2$ after 100 time steps for the advection dispersion equation.

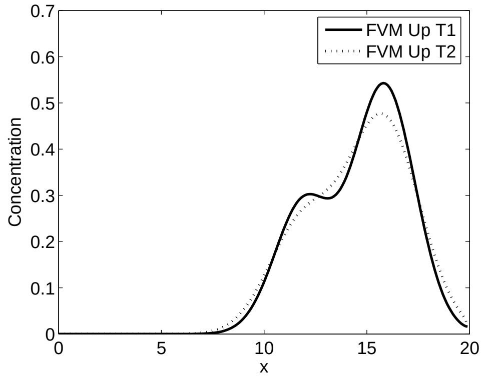


Figure F.2: FVM upwind with $T1$ and $T2$ after 100 time steps with $\Delta x = 0.1$ for the advection dispersion equation.

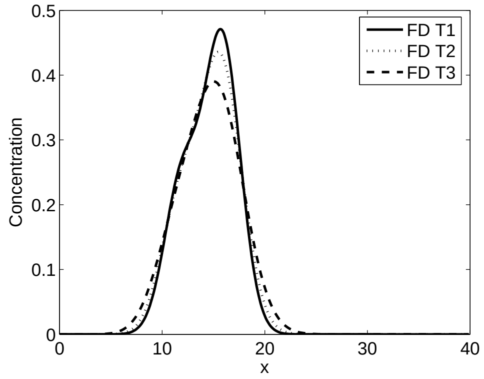


Figure F.3: FDM with $T1$, $T2$ with $\Delta x = 0.1$ and $T3$ with $\Delta x = 0.2$ after 100 time steps for the advection dispersion equation.

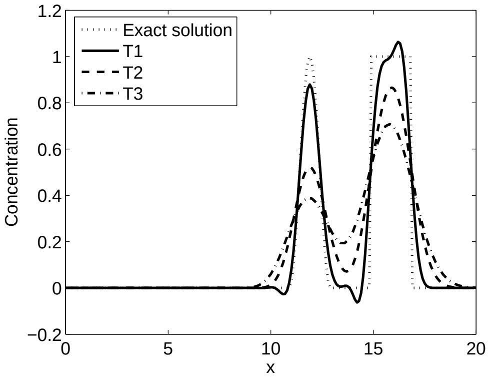


Figure F.4: SUPG classical upwind with $T1$, $T2$ and $T3$ for the advection equation after 100 time steps.

Appendix G

Advection equation 1D

In the Interim Master's thesis 1D numerical experiments were presented for the advection equation. Choose $D = 0$ and $q = 0.03$ and note that the exact solution is $C(x,t) = f(x - qt)$, with $f(x)$ the initial condition. Results after 100 time steps can be found in Figure G.1 for the FEM. The SGA is unstable and gives large wiggles. In Figure G.2 the results for the FVM and FDM are shown. SUP

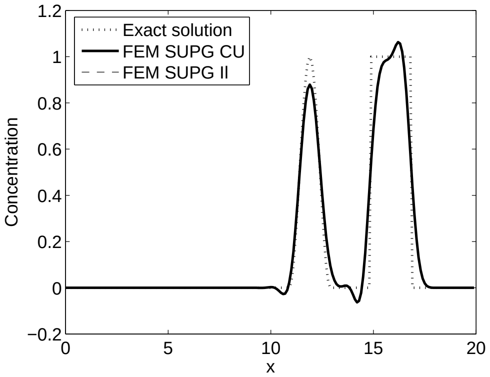


Figure G.1: SUPG classical upwind and Il'in scheme for the advection equation after 100 time steps with $T1$.

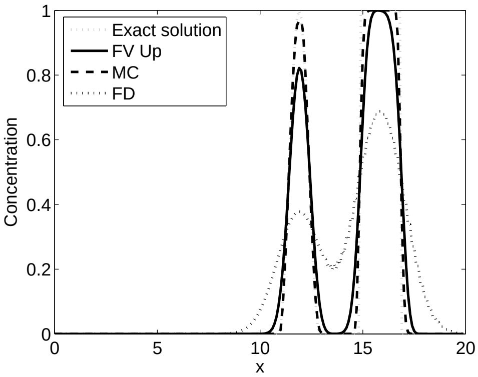


Figure G.2: FVM and FDM for the advection equation after 100 time steps.

Appendix H

Applications

Some other applications of density dependent groundwater flow are presented.

H.1 Coast line

The first experimental problem is a representative problem for the coast in the western part of the Netherlands. A coast line is presented with a convex and concave shoreline. Adjacent to the sea the freatic inland can be found with a sink, where groundwater abstraction takes place. In this part of the land the groundwater supply is given. Adjacent to the freatic inland a polder topsystem can be found.

This problem has several aspects. The first is a sharp interface between brackish and freshwater on the coastline caused by freshwater flow to the sea. The second aspect is a diffusive transition from salt to brackish to fresh water under the polder behind the dunes. The last aspect is a sink that causes an upward transition from salt to freshwater. The well can be very sensitive for salinization. For drinking water the limit is 150 mg CL^- per liter water while salt groundwater contains about 10000 mg CL^- per liter water. The interface will not be sharp near the well.

Possible changes in the model are the elevation of the sealevel, the lowering of water levels in the polder due to subsidence and the increase of the abstraction rate of the well.

This example is shown in Figure H.1.

H.2 Coast line with water ways

This example is equal to the first experimental problem, but now waterways are present in the polder topsystem. For these waterways, the salt supply from the groundwater is important. This experimental problem can be found in Figure H.2.

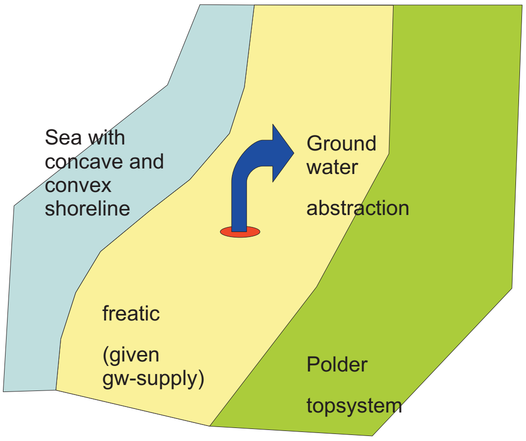


Figure H.1: Experimental problem: coast line.

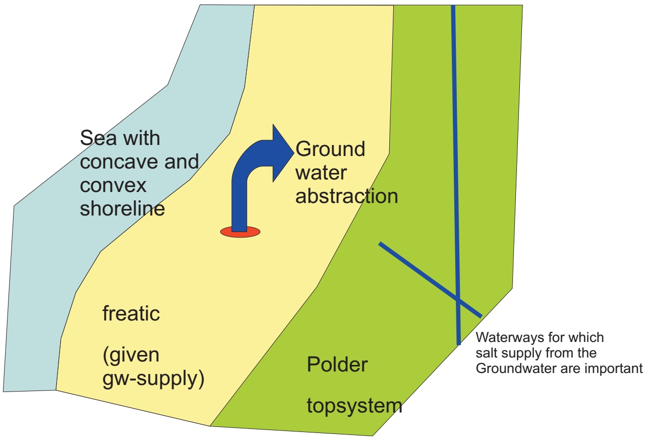


Figure H.2: Experimental problem: coast line with water ways.

H.3 Henry problem

The transient Henry problem describes the saline intrusion caused by a sudden change in fresh groundwater discharge. This approach is also effective for other variable density flow scenario, since it allows one to decouple the flow and transport equations.

Henry (1964) presented an analytical solution for a problem of groundwater flowing toward a seawater boundary. Because an analytical solution was available for the Henry problem, many numerical codes have been evaluated and tested with the Henry solution. Segol (1993) showed, however, that the Henry solution was not exact because Henry (1964) eliminated, for computational reasons, mathematical terms from the solution that he thought to be insignificant. When Segol (1993) recalculated Henrys solution with the additional terms, the improved answer was slightly different from the original solution.

The basic design of the Henry problem is shown in Figure H.3. The cross-sectional box is 2-m long, by 1-m high, and by 1-m wide. A constant flux of fresh ground water is applied to the right boundary at a rate of 6.6×10^{-5} m^3/d with a concentration equal to zero. A constant head boundary is applied to the left side of the box to represent seawater hydrostatic conditions. The upper and lower model boundaries are no flow.

The Henry problem caused further confusion among the modeling community because some researchers attempting to verify numerical codes calculated an erroneous value for molecular diffusion that did not correlate with the original value used by Henry (Voss and Souza, 1987). For this reason, some researchers consider there to be two cases of the Henry problem: one in which the value for molecular diffusion is 1.62925 m^2/d and another with a value of 0.57024 m^2/d. The values of other parameters can be found in Table C.1 [7].

porosity	0.35
seawater concentration	35 kg/m^3
inflow rate	5.702 m^3/day
equivalent freshwater hydraulic conductivity	864 m/d
molecular diffusion (case 1)	1.62925 m^2/day
molecular diffusion (case 2)	0.57024 m^2/day

Table C.1.: parameters of The Henry problem.

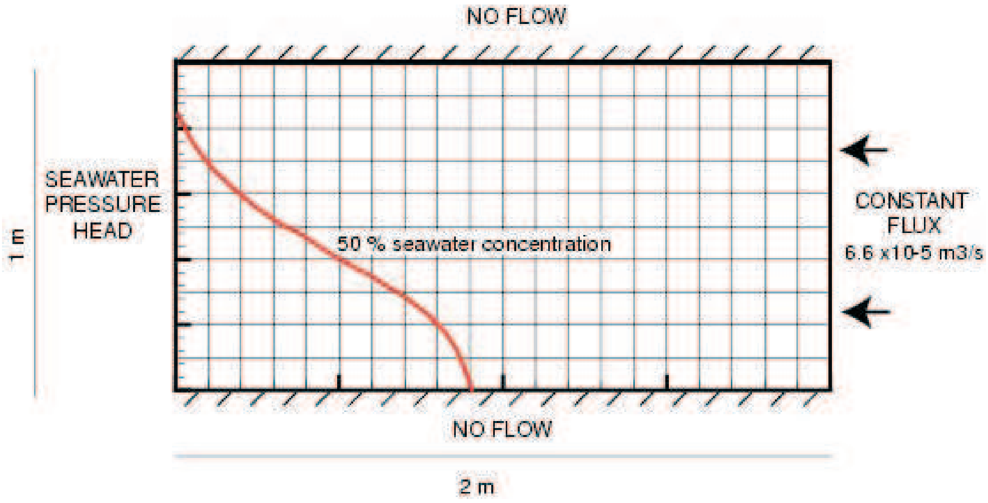


Figure H.3: Experimental problem: Henry problem.