Chapter 1

Water movement

The soil water, θ is in the model divided into 3 parts:

$$\theta = \theta_1 + \theta_2 + \theta_3 \tag{1.1}$$

The third domain is made for describing the macroporous flow whereas the primary and secondary domain are representing the water in the soil matrix

$$\theta_m = \theta_1 + \theta_2 \tag{1.2}$$

where θ_m is volumetric water content in the matrix domain. The division of the matrix domain into 2 subdomains is solely made for a better description of solute movement - see more in chapter 2.

1.1 Richards' Equation

The water flow in porous media can be described with the formula of Richard. The equation is derived here. The water flux density vector, $\mathbf{q}_m$ can be calculated by the Darcy´s law. For a two-dimensional vertical transect it yields:

$$\mathbf{q}_m = -\mathbf{K}(\psi)\nabla(\psi + z) \tag{1.3}$$

where $\mathbf{K}(\psi)$ is the hydraulic conductivity tensor, ψ is the potential head. The x-axis is chosen in horizontal direction and the z-axis is positive upwards. The conductivity tensor can be expressed as:

$$\mathbf{K} = \begin{bmatrix} K_{xx} & K_{xz} \\ K_{zx} & K_{zz} \end{bmatrix} \tag{1.4}$$

For a model with rectangular cells we have chosen that the principal directions of the anisotropic medium are parallel to the x- and z-axis, i.e.

$$\mathbf{K} = \begin{bmatrix} K_{xx} & 0 \\ 0 & K_{zz} \end{bmatrix} \tag{1.5}$$

The mass balance for the system gives

$$\frac{\partial \theta_{\mathrm{m}}}{\partial t} = -\nabla \cdot \mathbf{q}_{\mathrm{m}} - \Gamma_{\mathrm{wm}} \tag{1.6}$$

where θ_{m} is the volumetric water content and Γ_{wm} is the sink term for water. The partial differential equation can be developed by combining Darcy´s law, equation (1.3) and the mass balance, equation (1.6), thus

$$\frac{\partial \theta_{\mathrm{m}}}{\partial t} = \nabla \cdot (\mathbf{K}(\psi)\nabla(\psi + z)) - \Gamma_{\mathrm{wm}} \tag{1.7}$$

This is known as Richard's equation. For the modeling is assumed that the soil-water retention is without hysteresis, i.e. there is a unique relation between the matrix pressure potential and the water content.

To solve Richard's equation it is necessary to specify initial and boundary conditions. The boundary conditions specify a combination of ψ and its derivative on the boundary. Furthermore it is possible to use different forms of flux (Neumann) and predescribed pressure (Dirichlet) boundary conditions. The problem to be solved for determining the water movement can be summarized to

$$\begin{cases} \frac{\partial \theta_{\mathrm{m}}}{\partial t} = \nabla \cdot (\mathbf{K}(\psi)\nabla(\psi + z)) - \Gamma_{\mathrm{wm}} & \text{in } \Omega \\ \bar{\mathbf{n}} \cdot (\mathbf{K}(\psi)\nabla(\psi + z)) = -q_{\mathrm{m}} & \text{on } \partial\Omega^{N} \\ \psi = \psi_{0} & \text{on } \partial\Omega^{D} \end{cases} \tag{1.8}$$

where $\bar{\mathbf{n}}$ is the outward unit normal, and q_{m} is the magnitude of the outward flow from the domain. ψ_{0} is the predescribed pressure at the boundary. Ω is the soil domain. $\partial\Omega^{N}$ and $\partial\Omega^{D}$ are part of the boundary of Ω with Neumann and Dirichlet boundaries, respectively such that $\partial\Omega = \partial\Omega^{N} \cup \partial\Omega^{D}$. Each of $\partial\Omega^{N}$ and $\partial\Omega^{D}$ are not necessarily one continuous curve piece. A special case of the Neumann boundary conditions is often applied for the lower boundary condition, viz. it is assumed that the flow it is only driven by gravity (gravity boundary condition), i.e. $\partial\psi/\partial x = \partial\psi/\partial z = 0$ which gives

$$q_{\mathrm{m}} = \bar{\mathbf{n}} \cdot \begin{bmatrix} 0 \\ K_{zz} \end{bmatrix} \tag{1.9}$$

Another often used boundary condition is the seepage boundary condition for atmospheric boundaries. If a seepage face does not develop, the boundary acts as no flow. If a seepage face occurs we have a Dirichlet boundary condition with $\psi = 0$ and allow water to flow out of the domain. The condition can for instance be applied in connection with estuaries or streams.

1.2 Macropore flow

In the concept all macropores are vertical oriented. The macropore (tertiary) domain in the model contains a number of user specified macropore clases. In

a macropores class all the macropores have the same physical properties such as length. Each of the classes are characterized by distribution in the horizontal plane, radius of the pores and depth where the macropores start and ends. Also the pressures where the water starts and stops moving from the matrix to the macropore domain must be known. The macropores are also characterized by resistance for transferring water from a filled macropore to the matrix domain. The macropores can either end in the soil matrix or in a drain. When the macropores ends in a drain, matrix water which flows intro the macropore is instantaneously moved to the drain, and as consequence can macropores connected with drains newer be filled, and water can not be moved from the macropore to the matrix. The pressures where the water starts and stops moving from the matrix to the macropore domain must be known and the values are common for all the classes.

1.2.1   Macropore interaction with matrix water

The condition for macroporous flow to initiate and water move from the matrix to macropores in a macropores class is that the matrix pressure exceeds a certain value ψ_{initiate}.

$$\psi \geq \psi_{\text{initiate}} \tag{1.10}$$

When water is transferred from the matrix to the macroporous domain, the water is instantaneously moved to the top of the current water level in the macropores. If the whole macropore is empty, the incoming water is moved instantaneous to the bottom of the macropore or alternatively to the drain (if the macropore ends in a drain).

The water transfer from the matrix domain to the macroporous domain terminates if the matrix pressure is below a certain level, i.e.

$$\psi < \psi_{\text{terminate}} \tag{1.11}$$

In a location where the macropore class is filled with water, water is transferred from the macropore to the matrix domain if

$$\psi_{3,\text{ c}} > \psi \tag{1.12}$$

where $\psi_{3,\text{ c}}$ is the pressure potential in the macropore.

The quantification of the water movement toward a macropore is based on a relatively simple approach, very similar to theory of water the movement in a confined aquifer towards a well. For a confined aquifer of thickness D the stationary solution for water movement towards a well is

$$Q = \frac{2\pi K D(s_{\text{well}} - s)}{\ln(\frac{r}{r_{\text{well}}})} \tag{1.13}$$

where K is the (saturated) hydraulic conductivity, r_{well} is the radius of the well, s_{well} is the drawdown at the wall of well and s is the drawdown at the distance r from the center of the well.

If the macropores are equidistant placed, the density in the horizontal plane M_{c} can be approximated as:

$$M_{\text{c}} \approx \frac{1}{\pi r_{\text{c, mean}}^2} \tag{1.14}$$

where $2r_{\text{c, mean}}$ is the mean distance between the macropores.

In a small time step, the flow towards a macropore is considered as stationary and at the distance $r_{\text{c, mean}}$, the pressure in the current time step is considered as unaffected of the macropore, i.e. no pressure drawdown. Thus the flow to a piece of a single macropore with the height, Δz can be approximated as

$$Q_{\text{c, macro}} = \frac{2\pi K(\psi)\Delta z(\psi - \psi_{3, \text{ c}})}{\ln(\frac{r_{\text{c, mean}}}{r_{\text{c, macro}}})} \tag{1.15}$$

where $\psi_{3, \text{ c}}$ is the pressure potential in the macropores and $r_{\text{c, macro}}$ is the radius of the macropore and $K(\psi)$ is the hydraulic conductivity. Preventing that the hydraulic conductivity is very high in fractured media the $K(\psi)$ is computed as

$$K(\psi) = \min(K_{xx}(\psi), K_{xx}(\psi_{\text{initiate}})) = K_{xx}(\min(\psi, \psi_{\text{initiate}})) \tag{1.16}$$

where K_{xx} is the conductivity in the x-direction (see equation (1.5)). It is assumed that the flow towards the macropores is horizontal. Using equation (1.14) the sink term can be calculated

$$\Gamma_{\text{wm, c}} = \frac{M_{\text{c}}Q_{\text{c, macro}}}{\Delta z} = \frac{-4\pi M_{\text{c}}K_{xx}(\psi)(\psi - \psi_{3, \text{ c}})}{\ln(\pi M_{\text{c}}r_{\text{c, macro}}^2)} \tag{1.17}$$

For flow from the macropore domain into the matrix domain are the calculations made in a similar manner, but instead of using the conductivity is a resistance, $R_{\text{c, macro}}$ for flow out from the macropores introduced.

$$\Gamma_{\text{wm, c}} = \frac{-4\pi M_{\text{c}}(\psi - \psi_{3, \text{ c}})}{R_{\text{c, macro}}\ln(\pi M_{\text{c}}r_{\text{c, macro}}^2)} \tag{1.18}$$

The pressure at a given position in the macropore depends on the water level in the macropore

$$\psi_{3, \text{ c}} = z_{\text{c, macro}} - z \tag{1.19}$$

where $z_{\text{c, macro}}$ is the water level en the macropore. If the macropore is empty is $z_{\text{c, macro}} = z_{\text{c, bottom}}$ where $z_{\text{c, bottom}}$ is the z-coordinate of the bottom of the macropore. As a consequence of equation (1.19), we have for macropores which ends in drains:

$$\psi_{3, \text{ c}} = z_{\text{drain}} - z \tag{1.20}$$

where z_{drain} is the z-coordinate of the drain.

All the considerations above are for the transfer of water between a macropore class and the matrix. To calculate the total transfer between the macropores and the matrix it is necessary to sum up the contributions from each of the macropore classes. Thus the sink the macropores contributes to in the maxtrix flow is

$$\Gamma_{\text{wm, macro}} = \sum_{c=1}^{NC} \Gamma_{\text{wm, c}} \tag{1.21}$$

where NC is the number of macropore classes.

1.2.2 Macropore interaction with surface water

When the surface is ponded, water can directly enter the macropores without first entering the soil matrix. The rate is calculated very roughly and based on Poisseuilles law (e.g. Hillel, 1998). In the assumption made here only gravity drives the flow. The vertical flow in a macropore can be computed as:

$$Q_{\text{infiltration}} = \frac{\pi r_{\text{c, macro}}^4 \rho_w g (l + H_{\text{pond}})}{8 l \mu} \approx \frac{\pi \rho_w g r_{\text{c, macro}}^4}{8 \mu} \tag{1.22}$$

where μ is the dynamic viscosity, ρ_w the density of water, g the gravitational acceleration, l the distance from the surface to the water level in the macropore class (or the bottom of the macropore if it is empty) and H_{pond} is the ponding depth. The infiltration rate into the macropore class is:

$$i_{\text{c, macro}} = \frac{\pi M_{\text{c}} \rho_w g r_{\text{c, macro}}^4}{8 \mu} \tag{1.23}$$

The total infiltration into macropores is the sum of the infiltration into the different macropore classes

$$i_{\text{macro}} = \sum_{c=1}^{NC} i_{\text{c, macro}} \tag{1.24}$$

In the numerical model in a timestep of size Δt the implemented routine allows no more water for infiltration than present at the surface on the start of the timestep. Furthermore there can not infiltrate more water into a macropore class as there is space for in the start of the timestep. If all water is infiltrated in the timestep, the water is distributed between the classes proportional to the area density, M_{c} of the classes.

1.3 Finite Volume Method

1.3.1 Mesh

In Daisy2D, the domain, Ω is divided into N non-overlapping polygons, also denoted control volumes or cells. In Daisy2D it should be possible to choose be-

tween grids consisting of only rectangular cells or meshes consisting of trapezoids with two vertical faces. Figure 1.1 shows a grid only consisting of rectangular cells. The domain Ω in the figure is in divided into 3 subdomains, each consisting of a number of cells. Each subdomain is characterized by different hydraulic properties. The grid shown in figure 1.2 consists of trapezoids (where most of them also are rectangles). Only in the proximity of the drainpipe (see figure 1.3), the cells are not rectangular.

Figure 1.1: Example of grid consisting of rectangular cells.

The quadrilateral (rectangular or trapezoid) cells are denoted Q_i where $i = 1, 2, \cdots, N$. $|Q_i|$ denotes the area of Q_i, and ∂Q_i is the boundary of Q_i i.e. the edges (or faces) of Q_i. All internal edges e_{ij} are labeled by indices, i and j of the adjacent cells that shares face. The grid is constructed such that only whole faces are shared ($e_{ij} = Q_i \cap Q_j$). The length of e_{ij} is $|e_{ij}|$ and the unit normal vector pointing from Q_i into Q_j and orthogonal to e_{ij} is denoted $\bar{\mathbf{n}}_{ij}$. σ_i contains cell indices of cells sharing faces with cell i. σ'_i contain indices of cell faces of cell i which are placed on $\partial \Omega$, i.e. it is not shared with another cell. σ'_i is divided into two subsets, σ'^D_i and σ'^N_i of boundary cell faces with a Dirichlet and Neumann boundary condition, respectively.

Figure 1.2: Example of grid consisting of trapezoids.

1.3.2   Cell mass-balances

Richards equation is integrated over control volume (here a cell), Q_i. By applying the divergence theorem by Green-Gauss, we obtain

$$\int_{Q_i} \frac{\partial \theta_{\mathrm{m}}}{\partial t} d\Omega = \int_{\partial Q_i} \left(\mathbf{K}(\psi)\nabla(\psi+z) \right) \cdot \bar{\mathbf{n}} dl - \int_{Q_i} \Gamma_{\mathrm{wm}} d\Omega \qquad (1.25)$$

where $\bar{\mathbf{n}}$ is the outwarded unit normal and ∂Q_i the boundary of Q_i. The cell averages of θ_{m} and ψ are denoted θ_i and ψ_i. θ_i and ψ_i, $i = 1, 2, \cdots N$ where N is the number of cells that are collected in the vectors $\boldsymbol{\theta}$ and $\boldsymbol{\psi}$. Discretization of equation (1.25) based on a grid consisting of quadrilaterals yield

$$|Q_i| \left(\frac{d\theta_{\mathrm{m}}}{dt} \right)_i = \sum_{j \in \sigma_i} D_{ij}(\boldsymbol{\psi}) + \sum_{j \in \sigma_i} G_{ij}(\boldsymbol{\psi}) + \sum_{j' \in \sigma_i'} B_{ij'}(\boldsymbol{\psi}) - S_i(\boldsymbol{\psi}) \qquad (1.26)$$

where:

- $D_{ij}(\boldsymbol{\psi})$ describe the diffusive transport between internal borders

- $G_{ij}(\boldsymbol{\psi})$ describe the gravitational transport between internal boundaries

Figure 1.3: Close picture of grid near the drain pipe.

- $B_{ij'}(\boldsymbol{\psi})$ describe flux for external boundaries $j' \in \sigma_i'$

- $S_i(\boldsymbol{\psi})$ is the integrated sink term (point and area distributed sinks) in the cell.

The diffusive transport from cell i to cell j can be calculated as

$$D_{ij}(\boldsymbol{\psi}) = |e_{ij}|(\mathbf{K}(\boldsymbol{\psi}) \cdot (\nabla\psi)_{ij}) \cdot \bar{\mathbf{n}}_{ij} \tag{1.27}$$

For evaluating equation (1.27) it is necessary to estimate the gradient $(\nabla\psi)_{ij}$. $(\nabla\psi)_{ij}$ is evaluated by a different method for meshes with rectangular cells than for the more general and complicated case with meshes consisting of trapezoid cells. The gravitational transport from cell i to cell j can be calculated as

$$G_{ij}(\boldsymbol{\psi}) = |e_{ij}|(\mathbf{K}(\boldsymbol{\psi}) \cdot ([0 \; 1]^T)) \cdot \bar{\mathbf{n}}_{ij} \tag{1.28}$$

The boundary flux term is split into the contribution from boundaries with Neumann and Dirichlet condition respectively:

$$\sum_{j' \in \sigma_i'} B_{ij'}(\boldsymbol{\psi}) = \sum_{j' \in \sigma_i'^N} B_{ij'}^N(\boldsymbol{\psi}) + \sum_{j' \in \sigma_i'^D} B_{ij'}^D(\boldsymbol{\psi}) \tag{1.29}$$

For the boundaries with Neumann conditions we have

$$B_{ij'}^N(\boldsymbol{\psi}) = -q_{\mathrm{m},ij'}|e_{ij'}| \tag{1.30}$$

where $q_{\mathrm{m},ij'}$ is the size of the Darcy flux, perpendicular to the cell face and positive for flux out from cell i. The easiest way to implement Dirichlet boundary conditions is simply to force ψ_i to the value that ψ has on the face with Dirichlet conditions. Conflicts can arise if cell i has more than one face with a Dirichlet condition. Instead, the Dirichlet boundary condition is implemented as if the midpoint of the Dirichlet face was a neighbor cell. Similar to an interior cell face, a diffusive and a gravitational contribution can be calculated:

$$B_{ij'}^D(\boldsymbol{\psi}) = D_{ij'}^D(\boldsymbol{\psi}) + G_{ij'}^D(\boldsymbol{\psi}) \tag{1.31}$$

where

$$D_{ij'}^D(\boldsymbol{\psi}) = |e_{ij'}|(\mathbf{K}(\psi_i) \cdot (\nabla\psi)_{ij'}) \cdot \bar{\mathbf{n}}_{ij'} \tag{1.32}$$

$$G_{ij'}^D(\boldsymbol{\psi}) = |e_{ij'}|(\mathbf{K}(\psi_i) \cdot ([0\ 1]^T)) \cdot \bar{\mathbf{n}}_{ij'} \tag{1.33}$$

where the pressure associated with cell i has been used for calculating the hydraulic conductivity. The sink term used in equation (1.6) can be divided into two parts

$$\Gamma_{\mathrm{w}1} = \Gamma_{\mathrm{wma}} + \Gamma_{\mathrm{wmp}}\delta(x_p - x)\delta(z_p - z) \tag{1.34}$$

where Γ_{wma} is the contribution from a area distributed sink and Γ_{wma} is the contribution from a point sink. (x_p, z_p) are the coordinates of the point sink which shall be placed in the interior of a cell(not on the cell faces). δ is the Dirac delta function. Thus, the contribution from the sink terms to a cell yields

$$S_i(\boldsymbol{\psi}) = \Gamma_{\mathrm{wma}}|Q_i| + \Gamma_{\mathrm{wmp}} \tag{1.35}$$

Area distributed sinks are typically extraction from roots or (in Daisy2D) water flow between the soil matrix and macro pore domain. The point sinks can be tile drains or drip irrigation systems (point sources). Both Γ_{wma} and Γ_{wmp} can be dependent on the solution (ψ).

1.3.3 Rectangular cells

For the situation with a mesh consisting of rectangular cells, only matrix pressure in the four neighbor cells (see figure 1.4) are applied for calculating the fluxes through the faces of the cell (five point stencil). In the present section we will only evaluate the gradient for the "eastern" cell face of cell i. The theory can easily be applied for the 3 remaining directions. The distances necessary for evaluating the flux from a cell to the cell placed east of the cell are shown in figure 1.5.

The value of ψ in the midpoint of the eastern cell (ψ_E) can be expressed by a Taylor expansion of the value of ψ at the midpoint of the cell face:

Figure 1.4: Cell i and the neighbor cells it share faces with.

$$\psi_E = \psi(x + \delta x^+) = \sum_{k=0}^{m} \frac{1}{k!} \left(\frac{d^k \psi}{dx^k} \right)_f (\delta x^+)^k + R^+ \qquad (1.36)$$

where m is the order of the Taylor expansion and R^+ is the Lagrange remainder. Similar can ψ_i be computed

$$\psi_i = \psi(x - \delta x^-) = \sum_{k=0}^{m} \frac{1}{k!} \left(\frac{d^k \psi}{dx^k} \right)_f (-\delta x^-)^k + R^- \qquad (1.37)$$

It can be assumed that $R^+ - (-1)^{m+1} R^- \approx 0$. Thus if a Taylor expansion of first order ($m = 1$) is chosen we get

$$\left(\frac{d\psi}{dx} \right)_f (\delta x^+ + \delta x^-) \approx \psi_E - \psi_i \qquad (1.38)$$

If a higher order Taylor expansion is chosen we get

$$\left(\frac{d\psi}{dx} \right)_f (\delta x^+ + \delta x^-) \approx \psi_E - \psi_i - \epsilon_{Ei} \qquad (1.39)$$

where the correction term can be calculated as

$$\epsilon_{Ei} \approx \sum_{k=2}^{m} \frac{1}{k!} \left(\frac{d^k \psi}{dx^k} \right)_f \left[(\delta x^+)^k - (-\delta x^-)^k \right] \qquad (1.40)$$

Figure 1.5: Distances used for calculation of flux between cell i and its "eastern" neighbor.

It can be seen that a second order precision is obtained with $m = 1$ and $\delta x^+ = \delta x^-$. $m = 1$ is chosen for the relative simple model for rectangular cells. The width and height of cell i are denoted $(\Delta x)_i$ and $(\Delta z)_i$ respectively, thus $\delta x^- = \frac{(\Delta x)_i}{2}$, $\delta x^+ = \frac{(\Delta x)_E}{2}$ and $|e_{iE}| = (\Delta z)_i = (\Delta z)_E$. The outwarded unit normal, $\bar{\mathbf{n}}_{iE} = [1\ 0]^T$. By applying equation (1.27), the diffusive transport through the cell eastern face is:

$$D_{iE}(\boldsymbol{\psi}) = (K_{xx})_{iE} \frac{2(\Delta z)_i}{(\Delta x)_E + (\Delta x)_i} (\phi_E - \phi_i) \tag{1.41}$$

The gravitational transport from cell i to cell E is:

$$G_{iE}(\boldsymbol{\psi}) = 0 \tag{1.42}$$

If the eastern cell face of cell i belongs to the boundary of Ω (no eastern neighbor), $B_{iE'}$ shall be calculated. If the cell face has a Neumann boundary condition we have

$$B^N_{iE'}(\boldsymbol{\psi}) = -q_{iE'}(\Delta z)_i \tag{1.43}$$

where $q_{iE'}$ is the magnitude of the flux transported out from through the cell face. If the cell face have a Dirichlet boundary condition:

$$D^D_{iE'}(\boldsymbol{\psi}) = (K_{xx})_i \frac{2(\Delta z)_i}{(\Delta x)_i} (\psi_{E'} - \psi_i) \tag{1.44}$$

where $\psi_{E'}$ is the value of ψ in the midpoint on the eastern cell face of cell i. The gravitational part gives:

$$G^D_{iE'}(\boldsymbol{\psi}) = 0 \tag{1.45}$$

1.3.4 Conductivity at cell faces

The conductivity at the cell faces between adjacent cells (as used in equations (1.27)) are in Daisy calculated by either the arithmetic, geometric or harmonic mean. For steady state flow speaks physical arguments for applying the harmonic mean:

$$\frac{1}{K_{ij}} = \frac{1}{2} \left[\frac{1}{K(\psi_i)} + \frac{1}{K(\psi_j)} \right] \tag{1.46}$$

Simulation have shown that using the harmonic average can have the effect that water practically not can be transported in some cases with sharp gradients in the pressure potentials which can occurs in situations with evaporation and layered soil. Of that reason is the arithmetic mean chosen as default:

$$K_{ij} = \frac{1}{2} \left[K(\psi_i) + K(\psi_j) \right] \tag{1.47}$$

1.3.5   Upper boundary condition

The upper boundary condition describes how much of the applied water and surface water that infiltrates into the soil. For instance if the rate of the applied water exceeds the amount of water that can infiltrate into the soil, (the infiltrability) water is stored on the surface.

In the start of each of the iterations, within the time step, the infiltrability is calculated using Darcy's law (based on the pressure at surface in the last time step and the pressure in the surface cell.) If the amount of available water (surface water + applied water in the current time step) exceeds the amount of water that can infiltrate into the soil as calculated with the infiltrability, a Dirichlet (pressure) boundary condition is applied. If the amount of water which can infiltrate into the soil, as calculated with the infiltrability exceeds the amount of available water then a Neumann (flux) boundary condition is applied. The upper boundary can at a given time consists of parts with Dirichlet and parts with Neumann condition.

Surface flow

In order to take care of the surface water in simulations with a rectangular soil domain, a very simple surface flow module is developed.

In Daisy2D the surface flow model is executed after each time step. In a later version more physical based model can be included, for instance a solver of the Saint-Venant equations (see for example Chow *et al.*, 1988). In the present *0D* model is the surface water distributed so the resulting water level is equal for the whole surface. The water over a predefined level (detention storage) is removed.

1.3.6   Aquitard boundary condition

As in the existing one-dimensional Daisy it is possible to simulate the existence of an aquitard below the lower boundary of the soil domain. The aquitard is described by a thickness, a hydraulic conductivity and the pressure potential in the aquifer just below the bottom of the aquitard.

In start of the iteration loop, inside each time step, the flow across the lower boundary is estimated using Darcy´s law where the pressure in the boundary cells and the properties of the aquitard are required. The aquitard is then implemented as a Neumann boundary condition.

1.3.7   Tile drains

It is possible to simulate a (user defined) number of tile drains. Tile drains removes water when the matrix pressure potential in the soil around the drain is positive. The actual pressure in a drain pipe depends on position in the drain system, the hydraulic radius, etc, etc. An often applied simplification codes for variably saturated flow is to regard the pressure in the drain pipe as atmospheric.

When the soil in the drain point is unsaturated ($\psi < 0$) the solution corresponds to the solution for an undrained soil. If the soil is saturated ($\psi > 0$) the drains removes water from the soil matrix hence $\psi = 0$.

In the numerical model, the drain pipe is described as a point. The drain points shall be placed in the interior of a cell and cannot be placed at cell edges.

For obtaining a numerical stable solution it is in the beginning of a new iteration in the time step tested if the mean value of the matrix pressure in the drain cell and its eastern and western neighbors (if they exists) exceeds 0. If the mean value is positive the pressure in the drain cell is forced to zero. After each time step a mass balance for each of the drain cells is made to calculate the amount of drained water.

Test simulations show that the code both is able to turn on the drain when the soil is getting wetter and turn of the drain when the soil is getting drier. Figure 1.6 shows the results from a simulation with an aquitard boundary condition and a drain. The upper boundary has a no flux condition, thus the only supply of water is through the aquitard. As it can be observed, the matrix pressure potential in the drain is 0.

Figure 1.6: Matrix pressure potential in a drained soil. The drain is indicated with a dot. The lower boundary is formed by an aquitard condition.

1.3.8 Drip irrigation

1.3.9 Iteration scheme

Equation (1.26) describes how the matrix pressure potential in a given cell depends on the matrix pressure potential in the neighboring cells. By assembling equation (1.26) for $i = 1, 2 \cdots N$, the problem can be written as a ordinary differential equation (ODE) on the form:

$$\mathbf{Q}\frac{d\boldsymbol{\theta}}{dt} = \mathbf{E}(\boldsymbol{\psi})\boldsymbol{\psi} + \mathbf{F}(\boldsymbol{\psi}) \tag{1.48}$$

where $\mathbf{Q}$ is a diagonal matrix with $Q(i,i) = |Q_i|$ and $\theta_{\mathrm{m}} = \theta_{\mathrm{m}}(\psi)$. $\mathbf{E}(\boldsymbol{\psi})\boldsymbol{\psi}$ is the assembly of D_{ij} and $D_{ij'}^D$ and G_{ij}, $G_{ij'}^D$, $B_{ij'}^N$ and S_i are assembled in $\mathbf{F}(\boldsymbol{\psi})$. The equation is solved in the time domain using the backward Euler method:

$$\mathbf{Q}\frac{\boldsymbol{\theta}^{n+1,m+1} - \boldsymbol{\theta}^n}{\Delta t} = \mathbf{E}(\boldsymbol{\psi}^{n+1,m})\boldsymbol{\psi}^{n+1,m} + \mathbf{F}(\boldsymbol{\psi}^{n+1,m}) \tag{1.49}$$

In order to get rid of θ_{m} at iteration step $m + 1$, the mixed formulation by Celia *et al.* (1990) is applied. In the mixed formulation, the water content at time step $n+1$ and iteration step $m+1$ is approximated by a Taylor expansion:

$$
\begin{aligned}
\theta_{\mathrm{m}}^{n+1,m+1} &= \theta_{\mathrm{m}}^{n+1,m} + \frac{d\theta_{\mathrm{m}}}{d\psi}\mid^{n+1,m} (\psi^{n+1,m+1} - \psi^{n+1,m}) \\
&= \theta_{\mathrm{m}}^{n+1,m} + C^{n+1,m}(\psi^{n+1,m+1} - \psi^{n+1,m})
\end{aligned}
\tag{1.50}
$$

where $C = \partial\theta_{\mathrm{m}}/\partial\psi$ is the specific water capacity function. The time derivative of θ_{m} can then be approximated as:

$$
\begin{aligned}
\frac{\partial\theta_{\mathrm{m}}}{\partial t} &\approx \frac{\theta_{\mathrm{m}}^{n+1,m+1} - \theta_{\mathrm{m}}^n}{\Delta t} = \frac{\theta_{\mathrm{m}}^{n+1,m+1} - \theta_{\mathrm{m}}^{n+1,m}}{\Delta t} + \frac{\theta_{\mathrm{m}}^{n+1,m} - \theta_{\mathrm{m}}^n}{\Delta t} \\
&\approx C^{n+1,m}\frac{\psi^{n+1,m+1} - \psi^{n+1,m}}{\Delta t} + \frac{\theta_{\mathrm{m}}^{n+1,m} - \theta_{\mathrm{m}}^n}{\Delta t}
\end{aligned}
\tag{1.51}
$$

Thus, the iterative scheme is

$$\left(\frac{1}{\Delta t}\mathbf{Q}\mathbf{C}(\boldsymbol{\psi}^{n+1,m}) - \mathbf{E}(\boldsymbol{\psi}^{n+1,m})\right)\boldsymbol{\psi}^{n+1,m+1} =$$

$$\mathbf{F}(\boldsymbol{\psi}^{n+1,m}) + \frac{1}{\Delta t}\mathbf{Q}\mathbf{C}(\boldsymbol{\psi}^{n+1,m})\boldsymbol{\psi}^{n+1,m} + \frac{1}{\Delta t}\mathbf{Q}\left(\boldsymbol{\theta}^n - \boldsymbol{\theta}^{n+1,m}\right) \tag{1.52}$$

where $\mathbf{C}$ is a diagonal matrix with $C(i,i) = C_i$.

In the MATLAB-prototype it is possible to choose simulations with a constantly or dynamically size of the time steps, Δt. For the last choice, the size of Δt depends on how difficult it is to obtain a solution. A procedure based on same principles is described in detail in Mollerup (2001). In Daisy2D the current Daisy method will be applied.

1.3.10 Matrix solution technique

In the prototype, for solving the large matrix system of the type $\mathbf{Ax} = \mathbf{b}$ (see equation (1.52)), the MATLAB backslash operator (also called leftdivision) is used. For description of the applied sparse matrix solver is refereed to Mollerup (2001).

1.3.11 Hydraulic properties

In the Daisy2D it shall be possible to choose between the existing models for the soil hydraulic properties in Daisy. In the prototype, the retention characteristics are described with the model by van Genuchten (1980):

$$\theta_{\mathrm{m}} = \begin{cases} \theta_r + \frac{\theta_s - \theta_r}{[1+|\alpha\psi|^n]^m} & \text{for } \psi < 0 \\ \theta_s & \text{for } \psi \geq 0 \end{cases} \tag{1.53}$$

where α, n and m are empirical parameters, θ_s and θ_r are the saturated and the residual water content, respectively. By combination with the hydraulic conductivity model by Mualem (1976) and choosing $m = 1 - 1/n$, the hydraulic conductivity can be calculated as

$$K = K_s S_e^{1/2}[1 - (1 - S_e^{1/m})^m]^2 \tag{1.54}$$

where K_s is the hydraulic conductivity at saturation and S_e is the effective saturation defined as

$$S_e = \frac{\theta_{\mathrm{m}} - \theta_r}{\theta_s - \theta_r} \tag{1.55}$$

The retention model by van Genuchten has been adapted to a large class of soils.

1.4 Verification

The FVM-code is verified by comparing solutions obtained by FVM with quasi-analytical solutions for one-dimensional infiltration by Philip.

1.4.1 Infiltration Model of Philip

Philip (1957b) showed that the infiltration depth as function of time and saturation can be written as a power series in $t^{\frac{1}{2}}$. The coefficients are then functions of soil water content, θ_{m}. From the expression for the infiltration depth, as function of water content and time, it is relatively easy to derive that the cumulative infiltration, also can be written as a power series in $t^{\frac{1}{2}}$. The assumptions for the theory, is a one-dimensional vertical flow into a homogenous soil semi-infinite soil column, initially with uniform water content. The cumulative infiltration is expressed as

$$I = \sum_{n=1}^{+\infty} A_n t^{\frac{n}{2}} \tag{1.56}$$

16

where $A_1 = S$ is the often refereed sorptivity as defined in Philip (1969). The coefficients are found by solving a set of successive integro-differential equations. One drawback of the power series theory is that the theory only describes the infiltration process well for short to intermediate times. The power series is "practical convergent" for $t < t_{\text{grav}}$. Where t_{grav} is the characteristic time of the infiltration process

$$t_{\text{grav}} = \left(\frac{S}{K_0 - K_i} \right)^2 \tag{1.57}$$

where $K_i = K(\theta_i)$ and $K_0 = K(\theta_0)$ is the hydraulic conductivity corresponding to the initial water content, θ_i and the water content at the soil surface, θ_0. For ponded conditions at the soil surface we have $K_0 = K_s$.

The soil parametrization, which is applied for the test simulations, is the G.E. silt loam (van Genuchten, 1980) where $K_s = 4.96$ cm/day, $\theta_s = 0.396$ cm^3/cm^3, $\theta_r = 0.131$ cm^3/cm^3, $\alpha = 0.00423$ cm^{-1} and $n = 2.06$.

A constant size of Δt= 1/60 day has been applied in the FVM test simulations. For all simulations the initial condition is h_i=-200 cm, corresponding to $\theta_i = 0.332$ cm^3/cm^3 is chosen.

Vertical falling-head infiltration

Initially, it was shown that the power series solution can be applied for non-saturated or just saturated conditions at the soil surface (see Philip, 1955, 1957b,a). Philip (1958) later expanded the theory to cover ponding situations with constant positive pressure at the soil surface. Later it was shown (Mollerup and Hansen, 2007) that the power series solution also can be applied for a falling-head condition, where the ponding depth is dependent on the amount of infiltrated water. The pressure at the soil surface is then

$$H = H_0 - I \tag{1.58}$$

where H_0=20 cm is the initial ponding depth.

In the FVM simulations, both the vertical and horizontal discretisation, $\Delta z = \Delta x$ is 1 cm. The lower boundary was placed at $z = 600$ cm with a free drainage (gravity flow) condition. For the scenario is t_{grav}=3.34 days and the time at which the pond empties, t_p =2.6022 days is computed by applying the iteration procedure as proposed in Mollerup and Hansen (2007). In FVM-simulation, the pond empties at approximately t =2.5833 days. I.e. t_p is approximately 0.7% higher for the power series solution than for the similar FVM results obtained with a rather rough discretization in time. Minor errors can be expected in the power series solution as only the first 4 terms are calculated. For constant-head simulation the first 6 terms are calculated. Philip (1957b) found that normally only first two or three terms are necessary for a for practical use sufficient correct solutions.

17

Figure 1.7: Analytical and FVM solution for vertical falling head infiltration. The solution is shown for $t = 1/5$, $2/5$, $3/5$, $4/5$ and $1 \cdot t_{\mathrm{p}}$.

In figure 1.7, the wetting profiles as calculated by applying FVM and the power series theory are shown. The wetting profiles are shown for $t = 1/5$, $2/5$, $3/5$, $4/5$ and $1 \cdot t_{\mathrm{p}}$. As it can be observed, the solutions are almost identical except for $t = t_{\mathrm{p}}$ (2.6022 days) where the effects of the slightly earlier emptying ponded water in the FVM simulation instantly effects the water content profiles.

Horizontal constant-head infiltration

For also insuring that horizontal flows are simulated correctly a simulation with a horizontal oriented column is made. For the FVM simulation, the column has height of 1 cell and a width of 800 cells with $\Delta x = \Delta z = 1$ cm. The left boundary condition is $H = 20$ cm and the initial condition is $h_n = -200$ cm. Vertical constant-head infiltration can analytically be calculated as:

$$I = A_1 \sqrt{(t)} \tag{1.59}$$

where A_1 is identical to the A_1 calculated for vertical infiltration with constant-head (and falling-head) conditions. Contrary to vertical infiltration, equation (1.59) is applicable also for longer periods. Figure 1.8 shows the water content profiles at $t = 1/5$, $2/5$, $3/5$, $4/5$ and $1 \cdot t_{\mathrm{grav}}$. as calculated with FVM and the power series theory. As it can be seen are the solutions almost identical.

Figure 1.8: Analytical and FVM solution for horizontal infiltration. The solution is shown for $t = 1/5, 2/5, 3/5, 4/5$ and $1 \cdot t_{\mathrm{grav}}$.

1.4.2   Vertical constant-head infiltration in a wide column

Until now all the verification simulations are made for a grid consisting of only 1 cell in the direction perpendicular to the flow direction. Also the size of the cells was equal. In the wide column experiment the cell height varies with the depth. The soil column consists of 3 horizons (A, B and C). The A-horizon is 25 cm depth with $\Delta z =$ 1 cm, the B-horizon is 75 cm depth with $\Delta z =$ 3 cm, and the C-horizon is 400 cm depth with $\Delta z =$ 8 cm. The soil column have a width of 200 cm with $\Delta x =$ 20 cm. Figure 1.9 shows the mesh and figure 1.10 shows a upper part of the mesh.

In the simulation is the ponding depth constantly $H = 20$ cm. Figure 1.11 shows the water content after 1 day. As it can be observed, the water do not vary with the x-coordinate for a given depth, i.e. there is no indication of unintended exchange of water between internal vertical cell boundaries. Also here (not shown) comparisons with a power series solution shown fine agreement

1.4.3   Other simulations

Also a simulation with a Neumann (flux) condition at the upper boundary and a simulation with a non-zero sink term have been conducted. The simulations showed mass-balances with negligible errors.
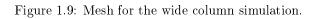
Figure 1.9: Mesh for the wide column simulation.

Figure 1.10: Upper left part of mesh used for the wide column simulation.

Figure 1.11: Water distribution after 1 day in the wide column simulation.

Chapter 2

Solute movement

2.1   2-domain matrix transport

For describing the solute movement, the soil matrix as solved by Richard' equation (equation (1.7)) the water is divided into two domains, a primary and a secondary domain:

$$\theta_{\mathrm{m}} = \theta_1 + \theta_2 \tag{2.1}$$

where θ_1 and θ_2 are the water content in the primary and the secondary domain, respectively. The primary part representing the flow in the smallest pores is always filled first. When the matrix water content, θ_{m} exceeds a certain limit, θ_{lim} the secondary domain start to be filled, i.e θ_{lim} is maximum value of θ_1. Thus, the primary part of θ_{m} can be expressed as

$$\theta_1 = \min(\theta_{\mathrm{m}}, \theta_{\mathrm{lim}}) \tag{2.2}$$

The secondary domain representing the flow in the largest pores is emptied first. The secondary part of θ_{m} can then expressed as

$$\theta_2 = \max(0, \theta_{\mathrm{m}} - \theta_{\mathrm{lim}}) \tag{2.3}$$

The fluxes as computed by Darcys equation are divided into two; a part representing the fluxes in the primary domain, $\mathbf{q}_1$ and a part representing the fluxes in the secondary domain, $\mathbf{q}_2$:

$$\mathbf{q}_{\mathrm{m}} = \mathbf{q}_1 + \mathbf{q}_2 \tag{2.4}$$

Similarly is the hydraulic conductivity matrix divided into a primary and a secondary part.

$$\mathbf{K}_{\mathrm{m}} = \mathbf{K}_1 + \mathbf{K}_2 \tag{2.5}$$

where $\mathbf{K}_1$ can be calculated

$$\mathbf{K}_1 = \mathbf{K}(\theta_1) \tag{2.6}$$

i.e. using the hydraulic conductivity function as used for the water movement computations, but with θ_1 instead of θ. Thus $\mathbf{K}_2$ can be calculated as:

$$\mathbf{K}_2 = \begin{cases} \mathbf{0} & \text{for } \theta_2 = 0 \\ \mathbf{K}(\theta_{\mathrm{m}}) - \mathbf{K}(\theta_{\mathrm{lim}}) & \text{for } \theta_2 \geq 0 \end{cases} \qquad (2.7)$$

As a consequence of Darcy's equation can the fluxes $\mathbf{q}_1$ and $\mathbf{q}_2$ be calculated as

$$\mathbf{q}_1 = \frac{\|\mathbf{K}_1\|_2}{\|\mathbf{K}\|_2}\mathbf{q} \qquad (2.8)$$

$$\mathbf{q}_2 = \frac{\|\mathbf{K}_2\|_2}{\|\mathbf{K}\|_2}\mathbf{q} \qquad (2.9)$$

The associated Darcy velocity can be calculated as $\mathbf{v}_1 = \mathbf{q}_1/\theta_1$ and $\mathbf{v}_2 = \mathbf{q}_2/\theta_2$. It should be remarked that when there is water in the secondary domain is the associated velocity, $\mathbf{v}_2$ often considerably larger than $\mathbf{v}_1$.

The solute concentration is similarly divided into a part associated with the primary water, C_1 and a part associated with the secondary water, C_2. The exchange of solutes between the primary and the secondary domain is driven by the concentration differences. The transfer of solutes from the primary domain to the secondary domain can be regarded as a sink in the primary domain, $\Gamma_{\mathrm{s1 \to s2}}$ or a source in the secondary domain, $-\Gamma_{\mathrm{s2 \to s1}}$

$$\Gamma_{\mathrm{s1 \to s2}} = -\Gamma_{\mathrm{s2 \to s1}} = \begin{cases} \alpha_{1 \to 2}(C_1 - C_2) & \text{for } C_1 \geq C_2 \\ \alpha_{2 \to 1}(C_1 - C_2) & \text{for } C_1 < C_2 \end{cases} \qquad (2.10)$$

The rates for moving solutes from C_1 to C_2, $\alpha_{1 \to 2}$ is not necessarily equal to the rate for moving solute from C_2 to C_1, $\alpha_{2 \to 1}$.

The mass balance for the solute can be expressed as:

$$\frac{\partial(\rho_b C_a)}{\partial t} + \frac{\partial(\theta_1 C_1)}{\partial t} + \frac{\partial(\theta_2 C_2)}{\partial t} + \frac{\partial(\theta_{\mathrm{mp}} C_{\mathrm{mp}})}{\partial t}$$
$$= -\nabla \cdot \mathbf{j}_1 - \nabla \cdot \mathbf{j}_2 - \nabla \cdot \mathbf{j}_{\mathrm{mp}} - \Gamma_{\mathrm{s}} \qquad (2.11)$$

where ρ_b is the soil bulk density and C_a is the concentration in the adsorbed phase. θ_{mp} is the volumetric water content in the macropore domain and C_{mp} is the concentration. $\mathbf{j}_1$, $\mathbf{j}_2$ and $\mathbf{j}_{\mathrm{mp}}$ are the fluxes in the the primary, secondary and macroporous domain. Γ_{s} is the net sink term of the solute.

2.2 Solute movement in the primary domain: Advection-dispersion equation

Three physical processes can contribute to movement of solutes in the primary part of the soil water:

- advection

- molecular diffusion

- hydrodynamic dispersion (only in connection with advection)

Advection (or bulk flow) is the process where the dissolved chemical moves with the soil solution. The flux of solute can be described as:

$$\mathbf{j}_1 = \mathbf{q}_1 C_1 \tag{2.12}$$

The Molecular diffusion is a result of the Brownian motion (random walk) of the molecules. A process related to the movement of the water is the hydrodynamic dispersion, which is a consequence of the fact that flow is not uniform, because the flow paths move around obstacles and air, but also because of variation in pore size and the non-uniform velocity distribution inside the pores. Mathematically the hydrodynamical dispersion process can be described as a diffusion process. The movement by diffusion like processes can be expressed as:

$$\mathbf{j}_1 = -\theta_1 \mathbf{D} \nabla C_1, \quad \mathbf{D} = \begin{bmatrix} D_{xx} & D_{xz} \\ D_{zx} & D_{zz} \end{bmatrix} \tag{2.13}$$

where $\mathbf{D}$ is the dispersion tensor (or matrix). The consequence is that the solute tries to move from areas with high concentration to areas with lower concentration. The elements in $\mathbf{D}$ are often calculated as:

$$
\begin{aligned}
D_{xx} &= \alpha_L \frac{v_{1,x}^2}{|\mathbf{v}_1|} + \alpha_T \frac{v_{1,z}^2}{|\mathbf{v}_1|} + D^* \\
D_{zz} &= \alpha_L \frac{v_{1,z}^2}{|\mathbf{v}_1|} + \alpha_T \frac{v_{1,x}^2}{|\mathbf{v}_1|} + D^* \\
D_{xz} &= D_{zx} = (\alpha_L - \alpha_T) \frac{v_{1,x} v_{1,z}}{|\mathbf{v}_1|}
\end{aligned}
\tag{2.14}
$$

where D^* is the molecular diffusion. The rest of the terms are arising from the hydrodynamic dispersion. α_L is called the longitudinal dispersion and α_T the transversal dispersion. The calculation of the dispersion tensor is based on $\mathbf{v}_1$ and not $\mathbf{v} = \mathbf{q}/\theta$.

The molecular diffusion can be calculated as:

$$D^* = \tau D_0 \tag{2.15}$$

where D_0 is the diffusion coefficient for the solute in free water and τ is the tortuosity factor. As an example Millington and Quirk (1961) suggested:

$$\tau = \frac{\theta_1^{7/3}}{\theta_s} \tag{2.16}$$

Also here, the value is based on θ_1 and not the the total matrix water content θ_{m}. If we are using equation (2.16) and expressing the mean velocity in the

pores associated with solute movement by $\mathbf{q}_1$ and θ_1, the elements of $\theta_1 \mathbf{D}$ can be expressed as:

$$\theta_1 D_{xx} = \alpha_L \frac{q_{1,x}^2}{|\mathbf{q}_1|} + \alpha_T \frac{q_{1,z}^2}{|\mathbf{q}_1|} + D_0 \frac{\theta_1^{10/3}}{\theta_s}$$

$$\theta_1 D_{zz} = \alpha_L \frac{q_{1,z}^2}{|\mathbf{q}_1|} + \alpha_T \frac{q_{1,x}^2}{|\mathbf{q}_1|} + D_0 \frac{\theta_1^{10/3}}{\theta_s} \qquad (2.17)$$

$$\theta_1 D_{xz} = \theta_1 D_{zx} = (\alpha_L - \alpha_T) \frac{q_{1,x} q_{1,z}}{|\mathbf{q}_1|}$$

The solute movement can be expressed as a sum of the advection and the diffusion process:

$$\mathbf{j}_1 = \theta_1 C_1 \mathbf{v}_1 - \theta_1 \mathbf{D} \nabla C_1 = C_1 \mathbf{q}_1 - \theta_1 \mathbf{D} \nabla C_1 \qquad (2.18)$$

The mass balance of dissolved solutes in the primary domain yields:

$$\frac{\partial(\theta_1 C_1)}{\partial t} = -\nabla \cdot \mathbf{j}_1 - \Gamma_{s1} \qquad (2.19)$$

where Γ_{s1} is the sink term which remove solutes from the primary water domain. The removed (or added) solute can be absorbed, moved to the secondary domain (Γ_{s1} as expressed by equation (2.10) or the macropore domain or be subject to chemical or biological reduction.

The boundary conditions to the equation specifies a combination of C_1 and its derivative on the boundary. Also here, the *Dirichlet* boundary condition (specified concentration) and the *Neumann boundary condition*, where the flux through the boundary is specified, are common. The Dirichlet boundary condition is:

$$C_1 = C_{1,0} \qquad (2.20)$$

where $C_{1,0}$ is the predescribed concentration. The Neumann boundary condition is:

$$\bar{\mathbf{n}} \cdot (C_1 \mathbf{q}_1 - \theta_1 \mathbf{D} \nabla C_1) = \bar{\mathbf{n}} \cdot \mathbf{j}_1 = j_1 \qquad (2.21)$$

where j_1 is the size of the flux, positive for outward flux. As boundary condition for the ingoing flow $j_1 = \bar{\mathbf{n}} \cdot \mathbf{q}_1 C_{1,0} = q_1 C_{1,0}$ is often used where $C_{1,0}$ is the concentration of the flow. As lower boundary condition is $j_1 = \bar{\mathbf{n}} \cdot \mathbf{q}_1 C_1 = q_1 C_1$ often used. In both cases it is assumed that the diffusion crossing the border is zero.

Summarized, the problem to be solved for determination of the concentration of solute in Ω is:

$$\begin{cases} \theta_1 \frac{\partial C_1}{\partial t} + C_1 \frac{\partial \theta_1}{\partial t} = -\nabla \cdot (C_1 \mathbf{q}_1 - \theta_1 \mathbf{D} \nabla C_1) - \Gamma_{s1} & \text{in } \Omega \\ \bar{\mathbf{n}} \cdot (C_1 \mathbf{q}_1 - \theta_1 \mathbf{D} \nabla C_1) = j_1 & \text{on } \partial\Omega_1 \\ C_1 = C_{1,0} & \text{on } \partial\Omega_2 \end{cases} \qquad (2.22)$$

where $\partial\Omega_1$ is the part of the boundary with Neumann condition, and $\partial\Omega_2$ is the part of the boundary with Dirichlet boundary conditions. Also here it is not necessary that $\partial\Omega_1$ and $\partial\Omega_2$, respectively, are coherent.

2.3   Numerical solution

The basic principles behind the finite volume modeling of the solute transport are very similar to the numerical solution of the water movement equation (Richards' equation). But there are some differences. One of the major differences is that the advection diffusion equation is considered as linear inside each timestep. I.e. the coefficients in the equation in each of the timesteps are independent of the concentrations in the current timestep. This simplification can be done when the size of the sources are dependent only on the concentrations in the previous timstep. Similar are adsorption added as sink/source term. Thus different from the water movement simulation, the Picard iterations loop used inside each timestep can be avoided.

2.3.1   Stabilization methods

There are often a lot of numerical problems involved with the solving of the convection diffusion problem especially when the problems are dominated by convection. The numerical solutions have often unexpected oscillations in that situation. There have been developed a lot of more or less complicated methods to reduce the problems. Three of the methods are *upstream weighting, streamline diffusion*, and *timestep reduction*.

Upstream weighting

When steep concentration fronts occur, numerical oscillations can raise. A method to stabilize the system is to apply upstream weighting for the advective solute movement. For advective transport between two cells is the concentration at the face between the cells normally calculated as the average concentration of the two cells. For fully upstream weighting is the concentration at the cell face equal to the upstream concentration.

In Daisy2D it is possible to set a parameter, $0 \leq \alpha \leq 1$ where $\alpha = 1$ corresponds to a fully upstream weighting and $\alpha = 0.5$ corresponds to setting the cell face concentration to the average concentration of the two cells. It is not recommended to apply an $\alpha < 0.5$.

P_e and C_r numbers

There are two different numbers which are important for the stability. The *Peclet number*:

$$P_e = v_1 \Delta x / D \tag{2.23}$$

where v is the velocity, Δx is the space increment and D is the diffusion coefficient (including molecular diffusion and hydrodynamic dispersion). The *Courant number* is defined as:

$$\mathrm{C_r} = v_1 \Delta t / \Delta x \tag{2.24}$$

Theoretical stability investigations are rather complicated, especially in a two or three-dimensional space with heterogeneous soil. Most of the theoretical stability considerations are done for one-dimensional flow with uniform velocity. The classical constraints for stability for the standard Crank-Nicholson-Galerkin (Finite Element Method) is $\mathrm{P_e} \leq 2$ and $\mathrm{C_r} \leq 1$, Perrochet and Bérod (1993).

It can easily be concluded that keeping the Courant number lower than one is just a question of sufficiently small timesteps. But is it possible to make a mesh which under all circumstances prevents that the Peclet number raises over 2?. The Peclet number for the flow in the x-direction is:

$$
\begin{aligned}
(\mathrm{P_e})_x = \frac{q_{1,x}\Delta x}{\theta_1 D_{xx}} &= \frac{q_{1,x}\Delta x}{\alpha_L \frac{q_{1,x}q_{1,x}}{|\mathbf{q}|} + \alpha_T \frac{q_{1,z}q_{1,z}}{|\mathbf{q}|} + D_0 \frac{\theta_1^{10/3}}{\theta_s}} \\
&< \frac{q_{1,x}\Delta x}{\alpha_T \frac{q_{1,x}q_{1,x}+q_{1,z}q_{1,z}}{|\mathbf{q}_1|}} = \frac{q_{1,x}\Delta x}{\alpha_T |\mathbf{q}_1|} \leq \frac{\Delta x}{\alpha_T}
\end{aligned} \tag{2.25}
$$

where it is assumed that $\alpha_L \geq \alpha_T$. The same procedure can of course be used to evaluate $(\mathrm{P_e})_z$. It can then be concluded that the maximum Peclet number is lower than $\Delta x/\alpha_T$. If the longitudinal dispersivity is 5 cm and the transversal is $1/10$ of the longitudinal and the maximum allowed $\mathrm{P_e}$ is 2 can it be concluded that the maximum side length of the elements shall be approximately 1 cm. This will result in a very fine mesh.

Besides the upstream weighting method it is possible to choose between 2 stabilizing methods:

1. Introducing extra diffusion in the streamline direction so $\mathrm{P_e}\mathrm{C_r} \leq \gamma$. Where γ is the performance index.

2. Varying the size of Δt so $\mathrm{P_e}\mathrm{C_r} \leq \gamma$.

It is of course also possible to deselect any stabilizing methods. The last stabilizing method is straight forward, but the first deserves its own subsection:

Streamline diffusion

For practical situation are there often stability so long $\mathrm{P_e}\mathrm{C_r} \leq \gamma$ where $2 \leq \gamma \leq 10$ (Perrochet and Bérod, 1993) which under all circumstances is less restrictive than keeping both $\mathrm{P_e} \leq 2$ and $\mathrm{C_r} \leq 1$. γ is called the *performance index*.

In the streamline diffusion is according to Perrochet and Bérod (1993) added

some additional longitudinal dispersion to prevent that P_eC_r raises over the chosen performance index. The additional longitudinal dispersion, $\bar{\alpha}_L$ is calculated as:

$$\bar{\alpha}_L = \begin{cases} \frac{|\mathbf{v}_1|\Delta t}{\gamma} - \alpha_L - \frac{D^*}{|\mathbf{v}_1|}, & \text{for } \alpha_L + \frac{D^*}{|\mathbf{v}_1|} < \frac{|\mathbf{v}_1|\Delta t}{\gamma} \\ 0, & \text{for } \alpha_L + \frac{D^*}{|\mathbf{v}_1|} \geq \frac{|\mathbf{v}_1|\Delta t}{\gamma} \end{cases} \tag{2.26}$$

Stability tests

To investigate the stability of the numerical model is a simple system modeled. The situation here is a one-dimensional column, horizontal column with steady-state water flow with pore velocity v. There is no secondary water, θ_2 thus $v_1 = v$. The diffusion, D (both molecular diffusion and hydrodynamic dispersion) is given. For the present test is the solute non-adsorping. For a scenario with linear adsorping (constant retardation factor) the adsorption have a stabilizing effect, thus for linear adsorping solutes the model is expected to be more stable. The advection-dispersion equation in one dimension can be written as:

$$\frac{\partial C_1}{\partial t} = D\frac{\partial^2 C_1}{\partial x^2} - v\frac{\partial C_1}{\partial x} \tag{2.27}$$

where $v = q/\theta$. The initial condition is a zero concentration in the whole column:

$$C_1(x,0) = 0 \tag{2.28}$$

At the left boundary is the solute flux constant.

$$\left(-D\frac{\partial C_1}{\partial x} + vC_1\right)\bigg|_{x=0} = vC_{1,0} \tag{2.29}$$

The solution can then according to van Genuchten and Alves (1982) be written as:

$$\begin{aligned} C1(x,t) = &\frac{1}{2}C_{1,0}\,\text{erfc}\left[\frac{x-vt}{2(Dt)^{1/2}}\right] + \left(\frac{v^2t}{\pi D}\right)\exp\left[-\frac{(x-vt)^2}{4Dt}\right] \\ &- \frac{1}{2}C_{1,0}(1 + \frac{vx}{D} + \frac{v^2t}{D})\exp(vx/D)\,\text{erfc}\left[\frac{x+vt}{2(Dt)^{1/2}}\right] \end{aligned} \tag{2.30}$$

For the simulations is made a water flow situation with steady state flow with the chosen pore water velocity $v = 1$ cm/hour. $C_{1,0}$ is for the simplicity chosen to 1 and $D = 0.05$ cm^2/hour. For the Daisy simulations is the virtual soil column 10 cm wide and 1 cm high. On the domain is generated a regular mesh with 100 equally large elements, each with $\Delta x = 0.1$ cm. With Δt chosen to 1 hour are $C_r = 10$ and $P_e = 2$, i.e. $P_eC_r = 20$. The numerical parameter, ω is set to $1/2$, i.e. a Crank-Nicholson scheme.

In figure 2.1 is the analytical solution compared with numerical solutions with and without upstream weighting corresponding to $\alpha = 1$ and $\alpha = 0.5$, respectively. Streamline diffusion and timestep reduction has not been applied. As it

can be observed, are the wiggles significantly smaller when applying upstream weighting. The only drawback seems to be slightly more numerical diffusion compared with the numerical solution for $\alpha = 0.5$. In the following cases up-

Figure 2.1: Analytical solution compared with numerical solution with regular weighting ($\alpha = 0.5$) and upstream weighting ($\alpha = 1.0$). The solutions are shown as concentration as function of x after simulation of 4 hours. For the actual case are $v = 1$ cm/hour, $D = 0.05$ cm^2/hour. For the numerical simulations are $\Delta t = 1$ hour and $\Delta x = 0.1$ cm, i.e. $P_e = 2$ and $C_r = 10$.

stream weighting has not been applied ($\alpha = 0.5$).

In figure 2.2 is the analytical solution shown. Besides is the numerical solution shown for the cases: no stabilization, timestep reduction and streamline diffusion. For both the timestep reduction and streamline diffusion method is the performance index, $\gamma = 10$ chosen. For the simulation without any stabilization are the wiggles significant. The wiggles are smaller for the simulation with streamline diffusion. By comparing with the analytical solution can the additional diffusion be observed. The additional diffusions effectively reduced the P_e-number from 20 to 10. The remaining graph shows the simulation with the timestep reduction stabilizing method where the size of Δt is changed so $P_e C_r \leq 10$. Here the size of the timestep is reduced from $\Delta t = 1$ hour (no stabilization) to 0.5 hour. Effectively the C_r-number is reduced from 10 to 5,

i.e. for the computation is used 2 times so many timesteps (or approximately 2 times so long CPU-time). Compared with the numerical solution without stabilization are wiggles reduced, but have also smaller wavelength.

Figure 2.2: Analytical solution compared with different numerical solutions. The solutions are shown as concentration as function of x after simulation of 4 hours. For the numerical solution without stabilization are $P_e = 2$ and $C_r = 10$.

In figure 2.3 is the analytical and numerical solutions shown. The numerical solutions are computed using different performances indexes. The performance index, γ is changed applying timestep reduction. It can be observed that the wiggles are significant for $\gamma = 10$, but for $\gamma = 5$ (and lower) the size of the wiggles seems to be acceptable for most purposes.

In figure 2.4 is the analytical solution shown. Also the numerical solutions for varying performances indexes are shown. The performance index, γ is changed applying streamline diffusion. The wiggles are reduced when using a low value of γ, but compared with the analytical solution, the steepness of the front is reduced dramatically.

Figure 2.3: Analytical solution compared with different numerical solutions obtained using timestep reduction with varying performance index. For the simulation are $v = 1$ cm/hour and $D = 0.05$ cm^2/hour. For the numerical simulations is $\Delta x = 0.1$ cm and Δt is ranging from $1/10$ hour ($\gamma = 2$) to $1/2$ hour ($\gamma = 10$).

Figure 2.4: Analytical solution compared with numerical solutions obtained using streamline diffusion with varying performance index. For the analytical solution are $v = 1$ cm/hour and $D = 0.05$ cm^2/hour. For the numerical simulations are $\Delta x = 0.1$ cm and $\Delta t = 1$ hour.

2.3.2 Upper boundary condition

The upper boundary condition describes the movement of solute applied at the surface that moves into the soil. It also describes the movement of solutes out from the domain if the water is flowing out at the upper boundary.

If the soil surface not is ponded, the flux into the soil is moved by advection into the domain, with the water. If the surface is ponded with water containing a solute, the solute is also moved solely by advection. A little more accurate description could have been obtained if the boundary condition was implemented as a Dirichlet condition which also allowed the solute to be moved by dispersion and diffusion into the soil. It is assessed that the error is insignificant.

When the water moving out of the domain. No solute is following the water. This is for most solutes a good boundary condition - at least for evaporation processes. For a situation with liquid water is leaving the soil through the upper boundary the description is not appropriate, but at present Daisy is not intended for that kind of scenarios. Summarized the upper boundary condition can be expressed as:

$$\left(-\theta_1 D_{zz} \frac{\partial C_1}{\partial z} + q_z C_1 \right) \bigg|_{z=z_{\text{surf}}} = \begin{cases} q_{\text{out}} C_{\text{surf}}, & q_{\text{out}} < 0 \\ 0, & q_{\text{out}} > 0 \end{cases} \tag{2.31}$$

where z_{surf} is the z-coordinate of the soil surface, q_z is the flow (positive upwards) and q_{out} is the flux out of the domain (negative for flux into the domain). C_{surf} is the concentration of the surface water.

Numerically all the types of upper boundary conditions is implemented as explicit Neumann conditions, i.e. the solute movement over the boundary is independent of the solute concentrations in the domain.

2.3.3 Lower boundary condition

The lower boundary condition describes the movement of solute through the lower boundary. If the water has a free drainage condition, there is a flux condition for the solute when the solute is moved out of the domain by advection. If there is specified a groundwater table or aquitard boundary condition, i.e. pressure (Dirichlet) conditions for the water flow, also the solute movement have a Dirichlet condition with a specified concentration at the boundary. For a specified steady-state water flux (mostly used for testing purposes), it is possible both to chose specified concentration (Dirichlet) and flux boundary conditions.

Numerically the Neumann boundary conditions is implemented, either implicit or explicit - implicit when the water flux is outwarded from the lower boundary and the concentration associated with the flux is given by the concentration inside the domain - explicit when the water moves into the domain and the

associated concentration is the concentration outside the domain.

The Dirichlet boundary condition is implemented as an explicit Neumann boundary condition. Based on the solution in the previous timestep the flux is calculated with the given concentration on the boundary. The method is different from the method used for the water movement, but prevents an extra iteration loop inside each timestep with following increased computational times. In the water movement simulations, the iteration loop was under all circumstances necessary since the equation is non-linear. But with very large concentration gradients (and following large movement by diffusion and dispersion processes) and small cells at the boundary some numerically problems can occur. To prevent this kind of instability, the size of the timesteps can be lowered if also timestep reduction is chosen as stabilizing method to prevent to high $P_e C_r$-numbers. If timestep reduction is chosen, the timestep is reduced so:

- For diffusion into the cell, only half of the volume of the concentration difference between border and cell can be transported by diffusion over the boundary into the cell in a timestep.

- For diffusion out from the cell, only half of the volume of the concentration difference between cell and border can be transported by diffusion over the boundary out from the cell in a timestep.

To prevent to large computational times, the timestep can not be lower than a chosen a minimum value. If the instabilities are high and produce negative concentrations at the cells at the boundary, the computations in the timestep are repeated and the solute is only moved by advection over the boundary. This can happen if the minimum value of the size of the timesteps is chosen to high.

2.3.4 Verification: One-dimensional flow with retardation and degradation

There are developed a lot of analytical solutions for the one-dimensional convective-dispersive equation, see for example van Genuchten and Alves (1982). The equations are developed for situations where the diffusion is constant and the water flow is steady state (i.e. $\partial \theta_1 / \partial t = 0$ and constant $\mathbf{q}$). The secondary water content, $\theta_2 = 0$ and all the adsorption processes are going through the primary water to the sorped phase. These conditions are seldom fulfilled in the 'real life' where both the water content and the flux are time-dependent. For testing the solute transport model is a situations with steady state water movement simulated.

If the adsorption process is very fast, the amount of adsorbed solute can be expressed with a adsorption isotherm which is a relationship between adsorbed (C_a) and dissolved concentration, C_1. The bulk density is assumed to be constant through time. The two first terms of the left hand side of equation (2.11)

can be rewritten as:

$$\rho_b \frac{\partial C_a}{\partial t} + \frac{\partial (C_1 \theta_1)}{\partial t} = \rho_b \frac{\partial C_a}{\partial C_1} \cdot \frac{\partial C_1}{\partial t} + \theta_1 \frac{\partial C_1}{\partial t} + C_1 \frac{\partial \theta_1}{\partial t} =$$
$$\theta_1 R \frac{\partial C_1}{\partial t} + C_1 \frac{\partial \theta_1}{\partial t} \qquad (2.32)$$

where R often in the literature is called the *Retardation factor*:

$$R = \frac{\rho_b}{\theta_1} \cdot \frac{\partial C_a}{\partial C_1} + 1 \qquad (2.33)$$

The most simple adsorption isotherm is the linear adsorption where $C_a = K_d C_1$ and as a consequence $R = 1 + \frac{\rho_b K_d}{\theta_1}$.

Zero or first order kinetics are included in the model. In zero order kinetics, the velocity of the reaction is independent of the concentration and in 1.st order kinetics the reaction velocity is proportional to the concentration. Thus the advection dispersion equation yields:

$$R \theta_1 \frac{\partial C_1}{\partial t} + C_1 \frac{\partial \theta_1}{\partial t} = -\nabla \cdot (C_1 \mathbf{q}_1 - \theta_1 \mathbf{D} \nabla C_1) - \theta_1 \mu_l C_1 + \rho_b \mu_s \qquad (2.34)$$

where the second last term represents a first order production in the liquid phase. μ_l is the rate constant. An often-used term is the half-life. In a batch experiment the half-life is the time required for the mass of reacting materiel to decrease to half the original mass. The reaction half-life can be calculated as $t_{1/2} = ln(2)/\mu_l$. The equation can be used for many chemical processes, and for radioactive decay. The last term on the right hand side of equation (2.34) represents a zero order removal from the solid to the liquid phase. μ_s is the rate constant for the zero order process. In van Genuchten and Alves (1982) is considered a one-dimensional case with degradation of both zero and first order. The governing differential equation can then be expressed as:

$$R \frac{\partial C_1}{\partial t} = D \frac{\partial^2 C_1}{\partial x^2} - v \frac{\partial C_1}{\partial x} - \mu C_1 + \gamma \qquad (2.35)$$

where μ is the rate constant for first order decay in the liquid and γ represents the similar rate constant for zero-order production in the liquid phase. For the simulation, the initial condition is

$$C1(x, 0) = C_{1,i} \qquad (2.36)$$

and the upper boundary condition is

$$\left(-D \frac{\partial C_1}{\partial x} + v C_1 \right) \Big|_{x=0} = \begin{cases} v C_{1,0}, & 0 < t \le t_0 \\ 0, & t > t_0 \end{cases} \qquad (2.37)$$

The solution is

$$C_1 = \begin{cases} \frac{\gamma}{\mu} + (C_{1,\mathrm{i}} - \frac{\gamma}{\mu})A(x,t) + (C_{1,0} - \frac{\gamma}{\mu})B(x,t), 0 < t \le t_0 \\ \frac{\gamma}{\mu} + (C_{1,\mathrm{i}} - \frac{\gamma}{\mu})A(x,t) + (C_{1,0} - \frac{\gamma}{\mu})B(x,t) - C_{1,0}B(x,t-t_0), t > t_0 \end{cases}$$

(2.38)

where $A(x,t)$ and $B(x,t)$ can be calculated as

$$A(x,t) = \exp(-\mu t/R)\cdot$$

$$\left\{ 1 - \frac{1}{2}\operatorname{erfc}\left[\frac{Rx - vt}{2(DRt)^{1/2}}\right] - \left(\frac{v^2 t}{\pi DR}\right)^{1/2}\exp\left[-\frac{(Rx - vt)^2}{4DRt}\right]\right.$$

$$\left. + \frac{1}{2}\left(1 + \frac{vx}{D} + \frac{v^2 t}{DR}\right)\exp(vx/D)\operatorname{erfc}\left[\frac{Rx + vt}{2(DRt)^{1/2}}\right]\right\}$$

(2.39)

$$B(x,t) = \frac{v}{v+u}\exp\left[\frac{(v-u)x}{2D}\right]\operatorname{erfc}\left[\frac{Rx - ut}{2(DRt)^{1/2}}\right] +$$

$$\frac{v}{v-u}\exp\left[\frac{(v+u)x}{2D}\right]\operatorname{erfc}\left[\frac{Rx + ut}{2(DRt)^{1/2}}\right] +$$

$$\frac{v^2}{2\mu D}\exp\left[\frac{vx}{D} - \frac{\mu t}{R}\right]\operatorname{erfc}\left[\frac{Rx + vt}{2(DRt)^{1/2}}\right]$$

(2.40)

with

$$u = v\sqrt{1 + \frac{4\mu D}{v^2}}$$

(2.41)

For comparing the analytical solution with the Daisy solution is chosen an situation with $v = 10$ cm/day, $D = 5$ cm^2/hour, $\gamma = 0.2$ hour^{-1} and $\mu = 0.5$ hour^{-1}. For the simulation is $\Delta x = 1$ cm. The length of the timesteps, Δt is $1/10$ day. In figure 2.5 is the Daisy solution compared with the above described analytical solution. As it can be seen are the solutions in practise coincident.

2.4 Solute movement in the secondary domain: Advection

The heterogeneities are normally relatively small in the secondary domain. And when $\theta_2 \ne 0$ are the movement by advection relatively large compared to the movement by molecular diffusion. As a consequence is diffusion-like processes (diffusion and dispersion) in the secondary domain negligible. Thus the solute movement is modeled as a purely advection process:

$$\mathbf{j}_2 = \mathbf{q}_2 C_2$$

(2.42)

The mass balance of dissolved solutes in the secondary domain yields:

$$\frac{\partial(\theta_2 C_2)}{\partial t} = -\nabla \cdot \mathbf{j}_2 - \Gamma_{\mathrm{s}2}$$

(2.43)

36

Figure 2.5: Comparison between analytical and Daisy simulation of a process with adsorption, zero order production and first order degradation.

where Γ_{s2} is the sink term which remove solutes from the secondary water domain. The removed (or added) solute can be absorbed, moved to the primary domain (Γ_{s1} as expressed by equation (2.10) or to the macropore domain or be subject to chemical or biological reduction.

Chapter 3

Heat transfer

3.1 Theory

Two physical processes can contribute to heat transfer movement in the matrix part (non macroporous part) of soil:

- conduction

- convection

Mathematical the transport can be expressed as:

$$\mathbf{q}_h = -K_H \nabla T + C_w \rho_w T \mathbf{q}_\mathrm{m} \tag{3.1}$$

where T is the temperature, K_H is the thermal conductivity, C_w the specific heat storage of water, ρ_w the density of water and $\mathbf{q}_\mathrm{m}$ is the water flux vector for matrix flux. Both K_H and C_w are calculated as in the existing 1D Daisy

The conservation of heat can be expressed as

$$\frac{\partial H}{\partial t} = -\nabla \cdot \mathbf{q}_h + S_h = \nabla \cdot (K_H \nabla T - C_w \rho_w T \mathbf{q}_\mathrm{m}) + S_h \tag{3.2}$$

where H is the heat content of the soil, t the time and S_h is a heat source. If water is added to the domain by drip irrigation with an amount of S_i the associated added heat is $C_w \rho_w S_i T_i$ where T_i is the temperature of the irrigation water which at the moment is the same as T, i.e. $T_i = T$. For water removed from the soil with a rate of S_w, the corresponding amount of heat is $C_w \rho_w S_w T$. In the model heat can also be added directly without exchanging water.

The left hand side of equation (3.2) can be written as

$$\frac{\partial H}{\partial t} = \frac{\partial (C_s T)}{\partial t} = C_s \frac{\partial T}{\partial t} + T \frac{\partial C_s}{\partial t} \tag{3.3}$$

where C_s is the specific heat capacity of the soil (including water).

The boundary conditions specifies a combination of the the temperature and its derivative on the boundaries. Actual in form of specified heat flux $K_H \nabla T - C_w \rho_w T \mathbf{q}_m$ (*Neumann boundary condition*) or specified temperature T_0 (*Dirichlet boundary condition*). Summarized, the problem to be solved for determination of the temperatures in Ω is:

$$
\begin{cases}
C_s \frac{\partial T}{\partial t} + T \frac{\partial C_s}{\partial t} = \nabla \cdot (K_H \nabla T - C_w \rho_w T \mathbf{q}_m) + S_h & \text{in } \Omega \\
\bar{\mathbf{n}} \cdot (C_w \rho_w T \mathbf{q}_m - K_H \nabla T) = q_h & \text{on } \partial \Omega_1 \\
T = T_0 & \text{on } \partial \Omega_2
\end{cases}
\tag{3.4}
$$

where $\partial \Omega_1$ is the part of the boundary with Neumann condition, and $\partial \Omega_2$ is the part of the boundary with Dirichlet boundary conditions. Also here it is not necessary that $\partial \Omega_1$ and $\partial \Omega_2$, respectively, are coherent. $\bar{\mathbf{n}}$ is the outwarded unit normal to the boundary and q_h is the size of the heat movement out of the boundary.

3.2 Numerical solution of Heat transfer equation

The basic principles behind the finite volume modeling of the heat transfer is almost equal to the numerical solution of the equation for the solute movement (advection-dispersion equation).

But there are some differences. In Dasisy2D is made the assumption that the thermal conductivity is equal in all direction. This makes the implementation easier than the advection-dispersion equation where the hydrodynamic dispersion are dependent on the direction. Furthermore the heat movement by conduction is so dominant over the movement caused by convection that numerical instabilities under normal conditions not is expected. Thus different from the implementation of the advection-dispersion equation there is not implemented made any stabilizing methods, except for the boundary conditions (see later).

The partial differential equation (PDE) describing the heat transfer (see equation (3.4)) is considered linear inside each timestep where the values from the beginning of the timestep is used. For frost/thaw processes both the conductivity for heat and the heat capacity is temperature dependent. But the changes during a typical timestep very small. Following are the errors, considering the PDE linear for each timestep small. As a consequence of the quasi linearity are Picard iterations inside each timestep avoided.

3.2.1 Upper boundary conditions

The upper boundary condition describes the transfer of heat energy between the atmosphere and the soil.

In Daisy is the upper boundary condition of Dirichlet type, i.e the surface is forced to have a specific temperature. Except when snow is covering the surface the soil temperature is approximated with the air temperature. For snow covered surfaces are the soil temperature estimated as in the existing Daisy.

At first the upper boundary was implemented in a similar manner as used for the Dirichlet boundary conditions for solute movement. For the heat simulations, the temperature is with the method given exactly at the boundaries. But with the relatively large timesteps intended to be used in the heat simulations the method is sometimes unstable.

Instead a more simple and stable method is applied. In the implementation, the boundary temperature is set in the the nodal points in the upper boundary cells. Then the specified temperature is given a half cell lower than wanted. The tradeoff compared to the other method is small since the uncertainties in the heat simulations are under all circumstances large. And reducing the CPU-costs by using large timesteps when possible is important.

3.2.2 Lower boundary condition

The lower boundary condition describes the transfer of heat energy through the lower boundary. It is possible to choose between 2 boundary conditions: A flux conditions where no energy is transferred through the boundary or as in Daisy 1D a forced temperature with a annual oscillation.

The forced temperature is implemented in a similar manner as for the upper boundary condition. For a no fluxcondition nothing has to be done.

Bibliography

Celia, M. A., Bouloutas, E. T., and Zarba, R. L. (1990). A general mass-conservative numerical solution for the unsaturated flow equation. *Water Resour. Res.*, **26**(7), 1483–1496.

Chow, V. T., Maidment, D. R., and Mays, L. W. (1988). *Applied Hydrology*. McGraw-Hill.

Hillel, D. (1998). *Environmental Soil Physics*. Academic Press, London, UK.

Millington, R. J. and Quirk, J. P. (1961). Permeability of porous solids. *Trans. Faraday Soc.*, **57**, 1200–1207.

Mollerup, M. (2001). *Numerical Modelling of Water and Solute Movement in Tilled Topsoil*. Ph.D. thesis, The Royal Veterinary and Agricultural University, Tåstrup, Denmark.

Mollerup, M. and Hansen, S. (2007). Power series solution for falling-head ponded infiltration. *Water Resour. Res.*, **43**. W03425, doi:10.1029/2006WR004928.

Mualem, Y. (1976). A new model for predicting the hydraulic conductivity of unsaturated porous media. *Water Resour. Res.*, **12**(3), 513–522.

Perrochet, P. and Bérod, D. (1993). Stability of the standard Crank-Nicolson scheme applied to the diffuion-convection equation: some new insights. *Water Resources Research*, **29**(9), 3291–3297.

Philip, J. R. (1955). Numerical solution of equations of the diffusion type with diffusivity concentration-dependent. *Transaction of the Faraday Society*, **51**(7), 885–892.

Philip, J. R. (1957a). Numerical solution of equations of the diffusion type with diffusivity concentration-dependent. II. *Australian Journal of Physics*, **10**, 29–42.

Philip, J. R. (1957b). The theory of infiltration: 1. the infiltration equation and its solution. *Soil Sci.*, **83**, 345–357.

Philip, J. R. (1958). The theory of infiltration: 6. effect of water depth over soil. *Soil Sci.*, **85**, 278–286.

Philip, J. R. (1969). Theory of infiltration. *Advances in hydroscience*, pages 215–296.

van Genuchten, M. T. (1980). A closed-form equation for predicting the hydraulic conductivity of unsaturated soils. *Soil Sci.*, **44**.

van Genuchten, M. T. and Alves, W.-J. (1982). *Analytical Solutions of the one-dimensional convective-dispersive solute transport equation*. United States Department of Agriculture.