

### 3. MATHEMATICAL MODEL

**AQUA3D** solves the three-dimensional groundwater flow and transport equations. In the flow equation the top layer can be treated as confined or unconfined. In the case of an unconfined aquifer the equations become nonlinear and are then solved by iteration. The aquifer is divided into vertical layers as many as necessary and they can be separated by semi-permeable layers.

The transport equation can both be solved for the transport of contaminants and heat and by proper selection of the parameters the user decides which one he wants to solve.

#### 3.1 Flow model

The three-dimensional movement of groundwater with constant density may be described by the following partial differential equation.

$$\frac{\partial}{\partial x} \left( k_{xx} \frac{\partial h}{\partial x} \right) + \frac{\partial}{\partial y} \left( k_{yy} \frac{\partial h}{\partial y} \right) + \frac{\partial}{\partial z} \left( k_{zz} \frac{\partial h}{\partial z} \right) = S_s \frac{\partial h}{\partial t} - Q \quad (3.1)$$

where:

$k_{xx}, k_{yy}, k_{zz}$	are values of the permeabilities along the principal axis, m/s
$h$	is the piezometric head, m
$Q$	is a volumetric flux per unit volume, $m^3/s/m^3$
$S_s$	is the specific storage coefficient, $m^{-1}$
$t$	is time, s

In the above equation it is assumed that the principal axis are in horizontal and vertical planes. The above equation applies to a local coordinate system within each element, so anisotropy can vary from element to element. The aquifer is now discretized into layers as many as necessary and the above equation is integrated vertically across each layer. The results are given below for n layers counted from top to bottom.

Top layer:

$$\begin{aligned} & \frac{\partial}{\partial x} \left( k_{xx} (h_1 - Zb_1) \frac{\partial h_1}{\partial x} \right) + \frac{\partial}{\partial y} \left( k_{yy} (h_1 - Zb_1) \frac{\partial h_1}{\partial y} \right) \\ & + R + g_2 (h_2 - h_1) + Q = S_s (h_1 - Zb_1) \frac{\partial h_1}{\partial t} + S_y \frac{\partial h_1}{\partial t} \end{aligned} \quad (3.2)$$

Layer i:

$$\begin{aligned} & \frac{\partial}{\partial x} \left( k_{xx} (Zt_i - Zb_i) \frac{\partial h_i}{\partial x} \right) + \frac{\partial}{\partial y} \left( k_{yy} (Zt_i - Zb_i) \frac{\partial h_i}{\partial y} \right) \\ & + g_i (h_{i-1} - h_i) + g_{i+1} (h_{i+1} - h_i) + Q = S_s (Zt_i - Zb_i) \frac{\partial h_i}{\partial t} \end{aligned} \quad (3.3)$$

Bottom layer:

$$\frac{\partial}{\partial x} (k_{xx} (Zt_n - Zb_n) \frac{\partial h_n}{\partial x}) + \frac{\partial}{\partial y} (k_{yy} (Zt_n - Zb_n) \frac{\partial h_n}{\partial y}) + g_n (h_{n-1} - h_n) + Q = S_s (Zt_n - Zb_n) \frac{\partial h_n}{\partial t} \quad (3.4)$$

where the average vertical conductance,  $\gamma_i$  is defined as:

$$g_i = \frac{l}{\frac{Zt_{i-1} - Zb_{i-1}}{2k_{zi-1}} + \frac{Zt_i - Zb_i}{2k_{zi}} + \frac{Zb_{i-1} - Zt_i}{k_i}} \quad (3.5)$$

where we have taken into account both the vertical permeability in each layer as well as vertical permeability in possible semi-permeable layers between the layers where:

$k_{xx}, k_{yy}$ ,	are values of the permeabilities along the principal axis in each layer, m/s
$S_s$	is the specific storage coefficient in each layer, $m^{-1}$
$h_i$	is the piezometric head in layer i, m
$Zb_i$	is the bottom elevation of layer i, m
$Zt_i$	is the top elevation of layer i, m
$k_{zi}$	is the vertical permeability in layer i, m/s
$k_i$	is the vertical permeability in a semi-permeable layer between layers i and i-1, m/s
$R$	is the infiltration rate, mm/year
$Q$	is the pumping/injection rate in each layer, $m^3/s$
$S_y$	is the specific yield

The following three kinds of boundary conditions are allowed in each layer.

- 1) Dirichlet boundary condition
- 2) Von Neumann boundary condition
- 3) Cauchy boundary condition

In the Dirichlet boundary condition the groundwater level, the piezometric head or the potential function is prescribed at the boundary as a function of time. In the von Neumann boundary condition the flow at the boundary is prescribed. Any given flow at a boundary can be modelled by defining source nodes (recharge or pumping) at the no flow boundary nodes. The Cauchy boundary condition is a head-dependant condition. The flow rate is related to both the normal boundary derivative and the head according to the following equation.

$$q = \begin{cases} a (h - h_o) & h \geq h_r \\ a (h_o - h_r) & h < h_r \end{cases} \quad (3.6)$$

where

$\alpha$	is head constant, $m^2/s$
$h_o$	surface elevation of river
$h_r$	bottom elevation of river

$q$  is boundary inflow,  $m^3/s$

### 3.2 Transport model

The three-dimensional transport of mass/heat in groundwater may be described by the following partial differential equation.

$$\begin{aligned} \frac{\partial}{\partial x} \left( D_{xx} \frac{\partial c}{\partial x} \right) + \frac{\partial}{\partial y} \left( D_{yy} \frac{\partial c}{\partial y} \right) + \frac{\partial}{\partial z} \left( D_{zz} \frac{\partial c}{\partial z} \right) + \\ + Q(c_w - c) - \left( V_x \frac{\partial c}{\partial x} + V_y \frac{\partial c}{\partial y} + V_z \frac{\partial c}{\partial z} \right) \\ = j R_d \frac{\partial c}{\partial t} + j R_d | c \end{aligned} \quad (3.7)$$

The above equation applies to a local coordinate system within each element having the main axis along the flow direction. We have assumed for the dispersion transport that vertical flow velocities are smaller than the horizontal ones.

The dispersion coefficients,  $D_{xx}$ ,  $D_{yy}$ ,  $D_{zz}$  are then defined by:

$$\begin{aligned} j D_{xx} &= a_L V^n + D_m j \\ j D_{yy} &= a_T V^n + D_m j \\ j D_{zz} &= a_T V^n + D_m j \end{aligned} \quad (3.8)$$

The retardation coefficients  $R_d$  are given by:

$$R_d = 1 + b(1 - j) r_s / (j r_l) \quad (3.9)$$

where  $\beta$ , the retardation constants are defined as:

$$b = K_d r_l \quad (3.10)$$

for mass transport, and for heat transport it is defined as

$$\text{where: } b = C_s / C_l \quad (3.11)$$

$c$	is solute concentration/temperature
$V_x, V_y, V_z$	is the velocity vector taken from the solution, m/s of the flow problem, m/s
$\alpha_L$	is longitudinal dispersivity, m
$\alpha_T$	is transversal dispersivity, m
$V$	is velocity, m/s
$D_m$	is molecular diffusivity, m <sup>2</sup> /s
$\phi$	is porosity
$c_w$	is concentration/temperature of injected water
$Q$	is pumping/injection rate, m <sup>3</sup> /s
$\lambda$	is exponential decay constant, s <sup>-1</sup>
$K_d$	is the distribution coefficient
$\rho_l$	is density of the liquid, kg/m <sup>3</sup> , 1000 kg/m <sup>3</sup>
$\rho_s$	is density of the porous medium, kg/m <sup>3</sup> , 2500 kg/m <sup>3</sup>
$C_l$	is specific heat capacity of the liquid
$C_s$	is specific heat capacity of the porous medium

Equation 3.7 is now integrated vertically across each layer. The results are given below for  $n$  layers counted from top to bottom. For convenience we define the thickness of each layer as:

$$b_i = Z_{t_i} - Z_{b_i} \quad (3.12)$$

and the thickness of possible semi-permeable layers as:

$$m_i = Z_{b_{i-1}} - Z_{t_i} \quad (3.13)$$

Top layer:

$$\begin{aligned} \frac{\partial}{\partial x} (j b_1 D_{xx} \frac{\partial c_1}{\partial x}) + \frac{\partial}{\partial y} (j b_1 D_{yy} \frac{\partial c_1}{\partial y}) - V_x b_1 \frac{\partial c_1}{\partial x} - V_y b_1 \frac{\partial c_1}{\partial y} = j b_1 R_d \frac{\partial c_1}{\partial t} + j b_1 R_d | c_1 \\ -(c_o - c_1)R - Q(c_w - c_1) - g_2(h_2 - h_1)(c_{1,2} - c_1) - d_2(c_2 - c_1) \end{aligned} \quad (3.14)$$

Layer i:

$$\begin{aligned} \frac{\partial}{\partial x} (j b_i D_{xx} \frac{\partial c_i}{\partial x}) + \frac{\partial}{\partial y} (j b_i D_{yy} \frac{\partial c_i}{\partial y}) - V_x b_i \frac{\partial c_i}{\partial x} - V_y b_i \frac{\partial c_i}{\partial y} \\ = j b_i R_d \frac{\partial c_i}{\partial t} + j b_i R_d | c_i - Q(c_w - c_i) - d_i(c_{i-1} - c_i) - d_{i+1}(c_{i+1} - c_i) \\ - g_i(h_{i-1} - h_i)(c_{i,i-1} - c_i) - g_{i+1}(h_{i+1} - h_i)(c_{i,i+1} - c_i) \end{aligned} \quad (3.15)$$

Bottom layer:

$$\begin{aligned} \frac{\partial}{\partial x} (j b_n D_{xx} \frac{\partial c_n}{\partial x}) + \frac{\partial}{\partial y} (j b_n D_{yy} \frac{\partial c_n}{\partial y}) - V_x b_n \frac{\partial c_n}{\partial x} - V_y b_n \frac{\partial c_n}{\partial y} = j b_n R_d \frac{\partial c_n}{\partial t} + j b_n R_d^1 c_n \\ - Q(c_W - c_n) - d_n (c_{n-1} - c_n) - g_n (h_{n-1} - h_n)(c_{n,n-1} - c_n) \end{aligned} \quad (3.16)$$

where

$$c_{i,i-1} = \begin{cases} c_i & \text{for outflow} \\ c_{i-1} & \text{for inflow} \end{cases} \quad (3.17)$$

The average vertical dispersion coefficient,  $\delta_i$  is defined as:

$$d_i = \frac{l}{\frac{b_{i-1}}{2(j D_{zz})_{i-1}} + \frac{b_i}{2(j D_{zz})_i} + \frac{m_i}{(j D_z)_i}} \quad (3.18)$$

where we have taken into account both vertical dispersion in each layer as well as vertical dispersion in possible semi-permeable layers between the layers.

where:

- $D_{xx}, D_{yy}, D_{zz}$  are values of dispersion coefficients along the principal axis in each layer,  $m^2/s$
- $D_z$  is the value of the vertical dispersion coefficient in a semi-permeable layer between layers,  $m^2/s$
- $c_i$  is concentration/temperature in layer  $i$

The following two kinds of boundary conditions are allowed in each layer.

- 1) Dirichlet boundary condition
- 2) Von Neumann boundary condition

In the Dirichlet boundary condition the concentration or temperature is specified at the boundary. In the von Neumann boundary condition the concentration gradient or the temperature gradient is set to zero indicating just convective transport of mass or heat through the boundary.

## 4 NUMERICAL MODEL

The equations for each layer in the flow model as well as the transport model are all special cases of the following equation.

$$a \frac{\partial u_i}{\partial t} + b_x \frac{\partial u_i}{\partial x} + b_y \frac{\partial u_i}{\partial y} + (e_{xx} \frac{\partial u_i}{\partial x}) + \frac{\partial}{\partial y} (e_{yy} \frac{\partial u_i}{\partial y}) + f_o u_i + f_{-1} u_{i-1} + f_{+1} u_{i+1} + g = 0 \quad (4.1)$$

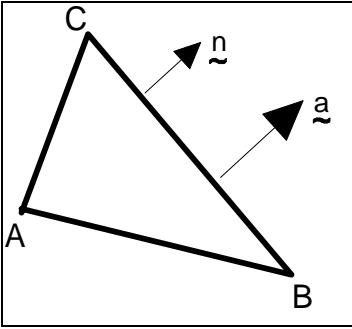
where  $u_i$  denotes the unknown head, concentration or temperature in the  $i$ -th layer and the  $a$ -,  $b$ -,  $e$ -,  $f$ - and  $g$ -coefficients are spatially dependent prescribed coefficients within the layer. In the top-layer equation (3.2) in the flow model the  $a$ - and  $e$ - coefficients may be dependent on the

unknown,  $u_i$ . The treatment of this special case is described at the end of section 4.4.

In **AQUA3D** each layer equation in the flow and transport model is approximated spatially by a Galerkin finite element method using linear approximations on each triangle.

### 4.1 Finite element notation

The approximation is most readily described in terms of a general triangular element ABC (see figure 4.1). We introduce the following notation,  $\Delta$  the triangular element,  $\partial\Delta$  the boundary of  $\Delta$ ,  $\partial\Delta_a$  the boundary edge BC, opposite A,  $|\Delta|$  the area of  $\Delta$ ,  $\mathbf{n}$  a general outward unit normal vector on  $\partial\Delta$ ,  $\mathbf{a}=(a_x, a_y)$  the outward normal vector on  $\partial\Delta_a$ , whose length is that of the edge, i.e. if  $(x_B, y_B)$  and  $(x_C, y_C)$  denote the coordinates of B and C resp.  $\mathbf{a} = (y_C - y_B, x_B - x_C)$ ,  $\mathbf{b}$  and  $\mathbf{c}$  are defined similarly,  $\mathbf{k}$  unit vector in the  $z$ -direction (vertically upwards)



We also define the following element matrices:

The element divergence matrix:

$$N = 1/2 \begin{bmatrix} a_x & a_y \\ b_x & b_y \\ c_x & c_y \end{bmatrix}$$

Figure 4.1.

The element mass matrix:

$$M = \frac{|\Delta|}{12} \begin{bmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 2 \end{bmatrix}$$

The lumped element mass matrix:

$$M_i = \frac{|\Delta|}{3} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

The element x-stiffness matrix:

$$K_x = \frac{1}{4|\Delta|} \begin{bmatrix} a_x^2 & a_x b_x & a_x c_x \\ a_x b_x & b_x^2 & b_x c_x \\ a_x c_x & b_x c_x & c_x^2 \end{bmatrix}$$

$K_y$  is defined similarly

The element stiffness matrix:

$$K = K_x + K_y = \frac{1}{|\Delta|} NN^T$$

The element x-derivative matrix

$$L_x = -\frac{1}{6} \begin{bmatrix} a_x & b_x & c_x \\ a_x & b_x & c_x \\ a_x & b_x & c_x \end{bmatrix}$$

$L_y$  is defined similarly.

## 4.2 Finite element approximations

The approximation of equation 4.1 is based on the following weak reformulation of that equation.

$$\begin{aligned} & \int_{\Delta} \left( a \frac{\partial u}{\partial t} + b_x \frac{\partial u}{\partial x} + b_y \frac{\partial u}{\partial y} + fu + g \right) dx dy + \int_{\partial\Delta} \left( e_{xx} \frac{\partial u}{\partial x}, e_{yy} \frac{\partial u}{\partial y} \right) \cdot ny ds \\ & - \int_{\Delta} \left( e_{xx} \frac{\nabla u \cdot \nabla y}{\nabla x \cdot \nabla x} + e_{yy} \frac{\nabla u \cdot \nabla y}{\nabla y \cdot \nabla y} \right) dx dy = 0 \end{aligned}$$

where we have adopted a local coordinate system which is such that the x-axis is the principal axis of the e-tensor so that  $e_{xy}=0$ .

$\psi$  denotes an appropriate test (weight) function. In **AQUA3D** the test functions on an element are  $\psi_i$ ,  $i = A, B, C$  where  $\psi_i$  is a linear function taking the value 1 at node  $i$  and the value 0 at the opposite edge. The resulting finite element approximation is:

$$aM \frac{d}{dt} u + (b_x L_x + b_y L_y + fM)u + Mg + P^d - (e_{xx} K_x + e_{yy} K_y)u = 0$$

where:

$$u = (u_A, u_B, u_C), \quad g = (g_A, g_B, g_C)$$

$u_i, g_i, i=A, B, C$  are the values of  $u$  and  $g$  at the given nodes and:

$$P^d = (P_A^d, P_B^d, P_C^d) \quad P_i^d = \int_{\Gamma_A} (e_{xx} \frac{\partial u}{\partial x}, e_{yy} \frac{\partial u}{\partial y}) \cdot n y_i ds \quad i = A, B, C$$

and we approximate  $a, b_x, b_y, e_{xx}$  and  $e_{yy}$  by constant values within each element. The  $P_i^d$  values do in fact not enter into the global system except on outer boundaries with a specified non-zero Neumann boundary condition or a Cauchy boundary condition, since they cancel out on all inner boundaries, thus implicitly ensuring continuity in flow. For the sake of simplicity and flexibility the approach taken in **AQUA3D** is to enter flow boundary conditions on a nodal basis. The boundary values for  $P_i^d$  in the assembled system can then be taken care of by entering non-zero flow values as point sources (or sinks) through the flow vector  $\mathbf{g}$  and in the case of Cauchy boundary conditions by also adding the head constant  $a$  (in eqn. 3.6) to the corresponding diagonal entry in the assembled stiffness matrix.

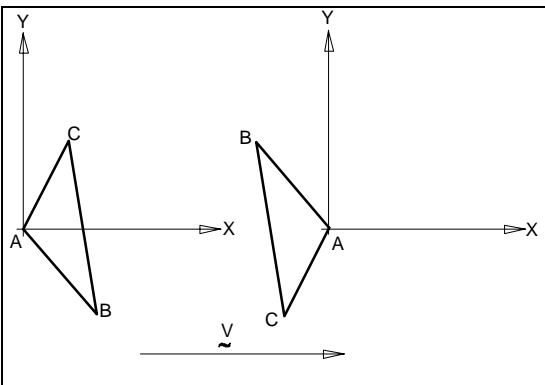
For the sake of simplicity  $M$  is further replaced by its lumped counterpart  $M_i$  in **AQUA3D**.

### 4.3 Upstream modification

Optionally in **AQUA3D**, when the terms  $b_x \partial u / \partial x$  and  $b_y \partial u / \partial y$  are present in equation 4.1 the spatial approximation can be stabilized by modifying the test functions within the element as where the coordinates within the element have been chosen so that  $b_x > 0$  and  $b_y = 0$ ,

$$y_i + m \frac{2 |\Delta|}{3 |a_x|} \frac{\partial y_i}{\partial x}, \quad 0 \leq m \leq 1, \quad i = A, B, C$$

the element is supposed to have one of the two configurations shown in figure 4.2 and  $\mu$  is a free parameter. Such a modification will in general damp out spurious oscillations that may arise in the case of large Peclet numbers (typically  $> 5$ ).



For the configuration in figure 4.2 the Peclet number may be defined as (cf. definition in section 4.7):

Figure 4.2.

(Note that  $e_{xx} < 0$  for transport models, cf.section 4.1 Also note that  $b_x$  and  $e_{xx}$  refer to the coefficients in equation 4.1 whereas  $a_x$  refers to the element edge opposite A, cf.section 4.1)

The modification amounts to a so-called Streamline Upstream Petrov Galerkin method or Localized Adjoint method, the test functions in the extreme case when  $\mu = 1$  being local solutions to the "adjoint" equation, in the limit as  $-b_x/e \rightarrow \infty$  :

The resulting modification to equation 4.2 is that:

The first modification amounts to adding the term

to the fourth term in equation 4.1 ie. replacing  $e_{xx}$  by  $e_{xx} (1 + \mu/3 Pe)$  where  $Pe$  is the Peclet number defined above. The modifications of the remaining terms ensure, however, that the resulting approximations remain consistent with the original equation. In **AQUA3D** the value of  $m$  is chosen by the user and only one global

value can be chosen.

#### **4.4 Time integration and solution of systems**

A fully implicit approximation is used for the time-derivative term  $\partial u / \partial t$  in equation 4.1 i.e. the derivative at time  $t$  is approximated by

$$\frac{1}{\Delta t} [u(t) - u(t - \Delta t)]$$

where  $\Delta t$  denotes the timestep. Thus a linear system whose matrix is the global counterpart of

$$\frac{a}{\Delta t} M_l - e_{xx} K_x - e_{yy} K_y + b_x L_x + b_y L_y + fM_l$$

has to be solved for each layer at each timestep with the head coefficient  $\alpha$  being added to the appropriate diagonal entry for nodes with specified Cauchy boundary conditions. With  $b_x = b_y = 0$  and  $e_{xx}$  and  $e_{yy} < 0$  as in the case of flow models this is a positive definite matrix and the system is solved by a band Choleski factorization, the bandwidth of the matrix having been minimized by ordering the nodes using the Gibbs-Stockmeyer-Poole algorithm. When  $b_x \neq 0$  and  $b_y \neq 0$  as in the case of transport models the matrix is no longer symmetric and the system is solved by LU-factorization using the same ordering of nodes as before. In this case we may get numerical instability unless pivoting is introduced. Resulting row exchanges in the global matrix will on the other hand increase the bandwidth of the matrix and thus add to the solution time and memory requirements. In [AQUA3D](#) the approach is taken not to introduce row exchanges but to print a warning if the size of a multiplier during the factorization exceeds 10. This may happen in the case of large Peclet numbers when no modification of the test function is introduced (cf. section 4.3) but such a modification will in general remedy the situation.

These linear systems for each layer are coupled together through the terms  $f_{-1}u_{i-1}$  and  $f_{+1}u_{i+1}$  in equation 4.1. This coupling is taken care of by the following iterative procedure:

The linear system for each layer is solved in succession starting with the top layer and going down to the bottom layer and then returning up again to the top through the layers in reverse order. When solving for the unknowns  $u_i$  in the  $i$ -th layer the most recently available approximations to the values  $u_{i-1}$  and  $u_{i+1}$  in the layers immediately above and below are used and these treated as knowns.

The sweeping procedure down and up through the layers is then iteratively repeated until a termination criterion is satisfied. The criterion is that in each layer either the maximum absolute change in  $u_i$  values between two iterations over all the nodes has to be less than a specified tolerance (tola) or the maximum relative change in  $u_i$  values between two iterations over all the nodes has to be less than a specified tolerance (tolr). If more than 100 iterations were needed to satisfy such a criterion the timestep  $\Delta t$  is halved (and all layer matrices thus refactored) before carrying on with the timeintegration. If on the other hand 100 iterations have sufficed for 100 consecutive timesteps the timestep is doubled before carrying on.

As soon as new  $u_i$ -values have been calculated for a given layer within an iteration these values are replaced by updated values using extrapolation from the previous iteration with a specified overrelaxation parameter,  $\omega$ , which is the same for all layers and all timesteps. The whole procedure thus amounts to using a fully implicit procedure for the whole coupled system at each timestep and in turn solving the coupled system with a block symmetric successive overrelaxation (block SSOR) method, the blocks corresponding to the layers.

For flow models changes in  $a$ - and  $e$ - coefficients may take place when these coefficients are dependent on the head. The approach taken in [AQUA3D](#) is to only refactor the matrix when the corresponding layer coefficients

have changed by a certain percentage specified by the user. These changes are otherwise taken care of by an appropriate modification of the right hand side in the system. For transport models the layer matrices are updated and factored again at each timepoint that corresponds to a specified velocity output and results in changes to the b-coefficients.

#### 4.5 Calculation of flow

In the absence of the terms  $b_x \partial u / \partial x$  and  $b_y \partial u / \partial y$  in equation 4.1 calculation of flow across an outer or inner boundary within each layer amounts to having to integrate  $(e_{xx} \partial u / \partial x, e_{yy} \partial u / \partial y) \cdot \mathbf{n}$  along the boundary where  $\mathbf{n}$  denotes the unit normal on the boundary and  $e_{xy} = 0$ . In **AQUA3D** approximations of this flow integral are based on the relationship between  $\mathbf{P}^d$  and the remaining terms in equation 4.2. This approach ensures mass balance in the calculated flow, provided equation 4.1 is based on a mass balance. In the presence of the terms  $b_x \partial u / \partial x$  and  $b_y \partial u / \partial y$  in equation 4.1, as in the case of the transport models, calculation of flow across an outer or inner boundary amounts to having to integrate  $(e_{xx} \frac{\partial u}{\partial x} + b_x u, e_{yy} \frac{\partial u}{\partial y} + b_y u) \cdot \mathbf{n}$

along the boundary. In **AQUA3D** the approach in this case is to replace the term  $b_x \partial u / \partial x$  in equation by the term  $(\partial / \partial x (u b_x) - u \partial b_x / \partial x)$ .

Similarly for  $b_y \partial u / \partial y$

The terms  $u \partial b_x / \partial x$  and  $u \partial b_y / \partial y$  may be incorporated into the remaining terms and the approximation of the term  $\partial / \partial x (u b_x) + \partial / \partial y (u b_y)$  may be based on the following weak reformulation:

$$\int_{\partial \Delta} (b_x u, b_y u) \cdot \mathbf{n} y ds - \iint_{\Delta} (b_x \frac{\partial y}{\partial x} + b_y \frac{\partial y}{\partial y}) u dx dy$$

which amounts to replacing  $(b_x L_x + b_y L_y) \mathbf{u}$  in equation 4.2 by

$$P^a - (b_x L_x^T + b_y L_y^T) u$$

$$P^a = (P_A^a, P_B^a, P_C^a) \quad P_i^a = \int_{\partial \Delta} (b_x u, b_y u) \cdot \mathbf{n} y_i ds, \quad i = A, B, C$$

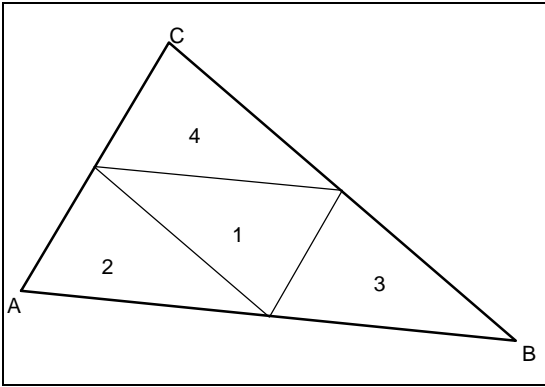
In **AQUA3D** the approximations of the flow integral in this case are based on the relationship between  $\mathbf{P}^d + \mathbf{P}^a$  and the remaining terms in the modified form of equation 4.2. Again a mass balance in the calculated flow is ensured provided the transport model is based on a mass balance equation.

#### 4.6 Tracing of pathlines

Pathlines may be traced in **AQUA3D** for a flow that may be considered to be in a steady state on a timescale comparable with the time it takes to travel along the line. Pathlines can just be traced in each layer, because no vertical connections between pathlines in different layers are calculated. This means that the actual pathlines are projected on the different layers. Flow values, calculated in **AQUA3D**, are constant within each triangular element, but continuity in normal flow between elements is implicitly retained through the fact that the  $\mathbf{P}^d$  term in equation 4.2 cancels out between elements in the global assembly process. Before tracing pathlines from the calculated flow in **AQUA3D** the calculated flow within the element is adjusted to the normal flow across the

element boundary, implicitly specified by the  $\mathbf{P}^d$  term by the following procedure, adopted from Cordes and Kinzelbach:

Each element is subdivided into four congruent subelements as shown in figure 4.3. The flow value in subelement I is taken to be the calculated flow of the element, but the flow values in the remaining subelements are recalculated from the following conditions that determine them uniquely:



1) Normal flow across the outer boundary of the element calculated from the new flow values is to be continuous.

2) When these normal flow values are substituted in for  $\mathbf{P}^d$  in equation 4.1 the equation has to hold true.

3) Circulation of the flow around each node following a path that joins together the midpoints of the edges adjacent to the

node is to be same whether it is calculated from the new flow values in the subelements adjacent to the node or from the original flow values of the element.

These new flow values can in fact be calculated locally by considering simultaneously all subelements surrounding a given node. Also note that in the absence of source terms other than point sources the normal flow will also be continuous across the edges between the subelements, but in the presence of such sources, these are effectively concentrated onto the edges between subelements causing jumps in normal flow across them.

#### 4.7 Grid size and time step requirements.

It is well known that the advective-dispersive transport equation is more difficult to solve numerically than the flow equation. The problems are particularly severe when advection dominates over dispersion. In this situation, the Galerkin finite element solution usually exhibits numerical spatial oscillations - overshoot and undershoot - near the concentration (temperature) front. Overshoot describes the erroneously high values of concentration encountered upstream of the moving front. The analogous behaviour on the downstream side is called undershoot.

These numerical oscillations tend to be more severe as advection becomes more dominant. General experience indicates that in a case where the dispersion coefficient,  $D$ , is greater than zero, numerical oscillations in the Galerkin finite element solution using linear basis functions can be virtually eliminated if the element size is selected so that its local Peclet number does not exceed 2. The Peclet number is defined as:

$$Pe = \frac{V\Delta l}{D} = \frac{V\Delta l}{a_L V^n + D_m}$$

where  $\Delta l$  is a characteristic length of the finite element grid. In most cases involving nonuniform flow, acceptable numerical solutions with very mild oscillations are achieved even when the local Peclet number is as high as 10. A simple guideline for selecting the finite element mesh and time step size can be given:

*The Peclet number  $< 10$*

$$\text{The Courant number} = \frac{V\Delta t}{\Delta l} < 1$$

The characteristic grid size is then given by:

$$\Delta l \leq 10(a_L V^n + D_m) / V$$

In case of  $n = 1$  and  $a_L V \gg D_m$  we have:

$$\Delta l \leq 10 a_L$$

The characteristic time step size is given by:  $\Delta t \leq \frac{\Delta l}{V}$

It should be borne in mind that in most cases it is acceptable to keep the grid size small locally around contamination or heat sources where the numerical results are of importance but ignoring the numerical oscillations in the far field, but instead being able to have large grid sizes far away from the sources.

It should also be noted that for low Peclet numbers ( $< 2$ ) a better solution may be obtained without the upstream correction mentioned in section 4.3.

For flow calculations the aquifer (start of pumping) is

$$\Delta t \leq \frac{\Delta l^2 S}{T}$$

appropriate time step after sudden stresses on the given by:

**AQUA3D** automatically checks the timestep, so if the number of vertical iteration becomes greater than 100 the timestep is automatically halved and is increased again after 100 timesteps if the number of iterations has been less 100 for that period of time. It should be borne in mind that sometimes it saves time to increase the timestep but at the same time increase the accuracy by selecting smaller values for the termination criterion see page 4.7. Also if oscillations occur in the solution it is necessary to decrease the timestep or increase the accuracy.