R-04-78

Rock-matrix diffusion in transport of salinity

Implementation in CONNECTFLOW

A R Hoch, C P Jackson
Serco Assurance

July 2004

Svensk Kärnbränslehantering AB
Swedish Nuclear Fuel
and Waste Management Co
Box 5864
SE-102 40 Stockholm Sweden
Tel 08-459 84 00
+46 8 459 84 00
Fax 08-661 57 19
+46 8 661 57 19

ISSN 1402-3091

SKB Rapport R-04-78

Rock-matrix diffusion in transport of salinity

Implementation in CONNECTFLOW

A R Hoch, C P Jackson

Serco Assurance

July 2004

This report concerns a study which was conducted for SKB. The conclusions and viewpoints presented in the report are those of the authors and do not necessarily coincide with those of the client.

A pdf version of this document can be downloaded from www.skb.se

Executive summary

In modelling groundwater flow in fractured rocks in Sweden, it has been found that in calibrating models against experiments over short time periods, the porosity that gives the best match to the observations tends to be very small, whereas in order to match the natural undisturbed flows, which are associated with much longer time scales, the porosity accessed needs to be much larger. A possible explanation for this is that in the experiments on short times, the porosity accessed is only the porosity associated with channels carrying the bulk of the flow within the fractures, whereas on longer time scales, the salinity has time to diffuse into less mobile water within the rock matrix, or within the fractures. That is, the salinity is effectively undergoing “rock-matrix diffusion”. It is important that this process is represented in order to model accurately transient groundwater flow and transport.

One of the programs used for modelling groundwater flow in Swedish rocks for SKB is CONNECTFLOW, which combines the facilities of the programs NAMMU for modelling continuum porous-medium models and NAPSAC for modelling discrete fracture-networks. Continuum porous-medium models can also be used to represent groundwater flow on a sufficiently large scale in fractured rocks, using suitable effective hydrogeological properties that represent the behaviour of the underlying fracture network.

The version of CONNECTFLOW current at the start of the work described here (Release 2.0) did not have a capability to model rock-matrix diffusion for saline flows in continuum porous-medium models of fractured rocks. Therefore, possible approaches for implementing such an option were evaluated and then the approach that was considered to be the most suitable was implemented and tested and then used in calculations for a realistic example.

Three main approaches for representing diffusion in the rock matrix were considered: the use of a numerical finite-difference scheme, an approach based on the use of Laplace transforms, and a so-called “hybrid” approach (based on an approach developed by Carrera et al. [1] which combines a series solution that gives a good representation at long times with an inverse square-root form that gives a good representation at small times. The details of these approaches are described. The ease of implementation, computational cost, I/O and storage requirements, flexibility and accuracy of the methods were compared.

Ease of implementation

All three approaches lead to similar changes to the discretised form of the equation for the transport of salinity. This is modified to include two additional terms: one proportional to the salinity at the end of the time step, and one that is a linear combination of certain intermediate quantities that depend on the salinities at previous time steps. Such a modification can be very readily implemented in CONNECTFLOW, and so there are no significant differences in the ease of implementation of the methods.

[1] J Carrera, X Sanchez-Vila, I Benet, A Medina, G Galarza and J Guimera. On Matrix Diffusion: Formulations, Solution Methods and Quantitative Effects, Hydrogeology Journal, 6, 1, p 178–190, 1998.

Computational cost

The corrections to the equation for the transport of salinity are effectively determined by carrying out 1D diffusion calculations using suitable techniques. For each time step, the 1D diffusion calculations are carried out for each element (or node) in the finite-element model, prior to carrying out the calculation of the transport of salinity for the 3D porous medium model of the fracture flow system. In general, the cost of all the 1D diffusion calculations is much less than the cost of solving the numerical system for the 3D model. Thus computational cost does not distinguish between the different approaches.

It should also be noted that, were it possible to use the same time steps as in a model without rock-matrix diffusion and with the porosity taken to be all the accessible porosity, then the overall computational cost of a simulation with rock-matrix diffusion would be only slightly more than the computational cost of a simulation without rock-matrix diffusion. In order that the time step sizes would not be unnecessarily restricted, modified versions of the Laplace transform and hybrid approaches were developed that would allow large time steps to be taken, and would not unnecessarily constrain the time step size by the need to model parts of the system where the response time is rapid, even though these parts are effectively in equilibrium. Nevertheless, it is likely that the time step sizes will have to be smaller if rock-matrix diffusion is modelled, increasing the overall computational time. The time-step sizes may have to be smaller in order to handle the faster flow in the fracture system, and may also have to be smaller for convergence of the method used to solve the numerical system for the flow and salinity at each time step.

Storage and I/O

As noted, the calculations for each time step involve a number of intermediate quantities for each element (or node). For the finite-difference approach and the hybrid approach, the numbers of intermediate quantities (for each time step and element) depend on the rock-matrix diffusion parameters, and are similar for both methods and might typically be of order fifty. For the Laplace transform approach, the intermediate quantities required are the values of the salinity in the fractures at all previous time steps. On average, therefore, the number of intermediate quantities (for each time step and element) is about half the total number of time steps, and would therefore be about 250 in a typical run with about 500 time steps. It would either be necessary to store the intermediate quantities in main memory or in disk files, if there is insufficient memory. This could distinguish between the different methods. The Laplace transform approach would require the use of files, rather than memory, for models with fewer elements, and the I/O cost of transferring information between files and memory would be higher. However, it should be noted that, even for the Laplace transform method, the I/O time would be a small fraction of the computational time for models with many elements.

Flexibility

The finite-difference approach is the most flexible of the approaches considered. In particular, it is the only approach that can deal with the case in which, at each location in the continuum representing the fractures, the groundwater density varies within the rock matrix. This is because diffusion within the rock matrix is non-linear in this case. The Laplace transform approach and the hybrid approach can only handle linear problems.

Accuracy

The Laplace transform approach and the hybrid approach treat the diffusion in the rock matrix exactly, whereas the finite-difference approach involves discretisation errors. In particular, the unit response function (the response to a step change in the concentration in the fractures) is significantly in error for the first few time steps. The overall effects of the errors could be reduced by reducing the time step size and the size of the grid blocks, but this could significantly increase the computational cost. It might be thought possible to attempt to choose an optimum time step for which the spatial and temporal discretisation errors effectively cancel. However, this is not a practicable approach, because the time step is generally determined by considerations relating to modelling the flow in the fractures.

Evaluation

In summary, there is little to choose between the methods in terms of ease of implementation or computational cost; the Laplace transform requires more storage or I/O time (although the I/O time is likely to be a small fraction of the overall computational time); the finite-difference method is less accurate than the Laplace transform or hybrid methods, which accurately represent diffusion into the rock matrix; and the finite-difference method is the most flexible approach and the only one that can handle the case in which, at each location in the continuum representing the fractures, the groundwater density varies within the rock matrix. In view of this, the method that was considered to be the most appropriate to implement initially in CONNECTFLOW for SKB was the hybrid method (modified as indicated to allow large time steps to be taken). This was mainly on the basis that the method provides a highly accurate representation of diffusion in the rock matrix. The method was preferred to the Laplace transform method because the latter method involves more intermediate quantities and greater I/O time. It is noted that, in order to use the method, it is necessary to make the approximation that, at each location within the continuum representing the fractures, the groundwater density is independent of position within the rock matrix. However, it is considered likely that this will be an acceptable approximation for many of the calculations that might be carried out for SKB.

The finite-difference approach is the only one that is sufficiently flexible to handle the non-linear case in which variations in groundwater density within the rock matrix at each location within the continuum representing the fractures are taken into account. For this reason, the method was initially preferred. However, the hybrid approach was ultimately chosen to be implemented because the finite-difference approach gives a less accurate representation of the diffusion in the rock matrix. However, there are many common features in the implementation of the various methods, and it would not require much work to implement the finite-difference approach in the future if this was considered desirable in order to model cases with large variations in salinity and hence groundwater density within the rock matrix at each location.

Implementation

The implementation of the hybrid algorithm is outlined and the required input data are described. The verification of the algorithm using a simple test case for which a semi-analytical solution is available is described.

Example

The algorithm was used to carry out calculations for a large site-scale model, based on the version 1.1 model of the Forsmark site in Sweden. Calculations of the evolution of the groundwater flow system from conditions 10,000 years ago to the present were carried out. The calculations took into account the changing elevation of the land surface relative to sea level resulting from post-glacial rebound and the changing salinity in the Baltic sea. The calculations were carried out for a Base Case without rock-matrix diffusion, but with the porosity taken to be all of the accessible porosity in the rock matrix, and for nine variants with different values of the rock-matrix diffusion parameters.

It was initially found that the calculations with rock-matrix diffusion required smaller time steps than the Base Case calculation in order to avoid instabilities in the solution of the numerical equation for the transport of salinity. This is a consequence of the porewater velocity being greater because the porosity has been reduced from that of the matrix to that of the fractures. The smaller time steps would have led to prohibitively long computational times. In order to deal with this, the algorithm for solving the numerical equations for the transport of salinity was enhanced to use mixed interpolation, nodal quadrature and a sequential-iteration time-stepping scheme. Mixed interpolation helps to avoid element-scale ripples in the vertical groundwater velocity in regions where the velocity is small. In the calculations, piecewise linear interpolation was used for the residual pressure and the salinity, and piecewise constant interpolation was used for the groundwater density. Nodal quadrature helps to avoid so-called "mass-matrix" ripples in the salinity resulting from the finite-element discretisation of the time derivative terms in the flow and transport equations. These changes make the numerical equations less non-linear and therefore easier to solve.

In the sequential iteration scheme, at each time step:

(i) the average density in each element is calculated from a suitable equation;

(ii) the residual pressure at the end of the time step is calculated (using the calculated average density) from a discretised version of the steady-state flow equations;

(iii) the salinity at the end of the time step is calculated from a discretised version of the salinity transport equation using the calculated average density and calculated flow.

There are different variants of this scheme, in which a single cycle of the above calculations is carried out for each time step, or a fixed number of cycles is carried out, or the cycles are repeated until convergence is obtained for the non-linear equations at each time step. It was found that, for the realistic example, repeating the cycles until convergence was obtained required a small time step, leading to prohibitively long calculations. However, if the time-stepping scheme that involved only a single cycle for each time step was used, it was found to be possible to use much larger time steps, similar to those used in calculations for the model without rock-matrix diffusion. This scheme was therefore adopted for the calculations.

With these changes, it proved to be possible to carry out calculations for the variants with rock matrix diffusion. The results are compared with those for the Base Case (without rock matrix diffusion).

Comments

This has demonstrated that the algorithm for rock-matrix-diffusion in transport of salinity can be used in some models of the sort that are likely to be used in practical modelling. The algorithm allows models to be set up in which the effective porosity for the continuum representing a network of fractures is taken to be a realistic porosity corresponding to the fractures. At the same time, the models can represent the possibility that there is a significantly greater porosity available within the rock matrix which may be accessed by diffusion on sufficiently long time scales. The fracture porosity could be obtained from short-term measurements or derived from fracture-network modelling (using either detailed numerical models or simple approximations).

The algorithm appears to be reasonably robust and allow calculations for a range of parameters. However, it should be noted that in general, smaller time steps will be required to model transport with rock-matrix diffusion than would be necessary for a model without rock matrix diffusion and with the porosity taken to be equal to the accessible porosity in the rock matrix (because the groundwater velocity will be faster and modelling the details of the diffusion into the rock matrix will be of interest).

The rock-matrix diffusion algorithm has currently been implemented in CONNECTFLOW only for the transport of salinity. However, it would be possible to use the current implementation to model migration of other solutes, which might be sorbing. It would also be possible to use the implementation to model the migration of a radionuclide, although it would be necessary for the user to post-process the calculated concentrations to take into account the effects of radioactive decay.

Contents

1 Introduction

The standard mathematical model for modelling saline groundwater flows in a porous medium is a single-porosity model that couples groundwater flow and transport of salinity. A variant of this model is implemented in the finite-element groundwater flow and transport program CONNECTFLOW, which combines the facilities of the programs NAMMU /1, 2, 3/ for modelling continuum porous-medium models and NAPSAC /4/ for modelling discrete fracture-networks. As well as representing saline flows in porous rocks, continuum porous-medium models can also be used to represent flows (on suitably large scales) in fractured rocks.

However, in modelling groundwater flow in fractured rocks in Sweden, it has been found /5/ that in calibrating models against experiments over short time periods, the porosity that gives the best match to the observations (i.e. the porosity apparently accessed by the salinity) tends to be very small (perhaps of order 10^{-4}), whereas in order to match the natural undisturbed flows, which are associated with much longer time scales, the porosity accessed needs to be much larger (perhaps of order 10^{-2}). A possible explanation for this is that in the experiments on short times, the porosity accessed is only the porosity associated with channels carrying the bulk of the flow within the fractures, whereas on longer time scales, the salinity has time to diffuse into less mobile water within the fractures or within the rock matrix. That is, the salinity is effectively undergoing "rock-matrix diffusion". Indeed, this would be expected in a fractured rock. It should be stressed that the large porosity accessed at long times is not necessarily distributed throughout the rock matrix, but may correspond to less mobile water within the fractures themselves or may be associated with zones immediately adjacent to the fractures.

Similarly, in the Finnish radioactive waste disposal programme, it has been found that the observed high salinities at the Olkiluoto site cannot readily be reproduced in an equivalent continuum model without rock matrix diffusion, whereas inclusion of rock–matrix diffusion in the model leads to salinities similar to those observed /6/.

The version of CONNECTFLOW current at the start of the work described here (Release 2.0) did not have a capability to model rock-matrix diffusion for saline flows in continuum porous-medium models of fractured rocks. Therefore, possible approaches for implementing such an option were evaluated and then the approach that was considered to be the most suitable was implemented. This work is described in this report.

In Section 2, a simple mathematical model for rock-matrix diffusion, which formed the basis of the work, is discussed. In particular, analytical solutions to the equation for diffusion into the rock matrix are presented. In Section 3, possible approaches for solving the rock-matrix diffusion equation in CONNECTFLOW are described. Most of these methods exploit the analytic solutions discussed in Section 2. The relative advantages and disadvantages of the various methods are compared in Section 4. The implementation in CONNECTFLOW of the method that was considered the most suitable is outlined in Section 5. In Section 6, the testing of the new option using a simple test case is described. In Section 7, the application of the option in a large site-scale model, based on the version 1.1 model of the Forsmark site in Sweden is presented. Finally, in Section 8 some concluding remarks are made.

2 Mathematical model

As noted in the Introduction, groundwater flow though a fractured rock on a large scale can be represented using a porous medium model. A basic model that represents flow of groundwater of variable salinity and also allows for rock-matrix diffusion can be expressed using the following equations:

$$\mathbf{q} = -\frac{k}{\mu} \cdot (\nabla P - \rho \mathbf{g}) \ , \tag{1}$$

$$\frac{\partial(\phi_f \rho)}{\partial t} + \nabla \cdot (\rho \mathbf{q}) = 0 \ , \tag{2}$$

$$\frac{\partial(\phi_f \rho c)}{\partial t} + \nabla \cdot (\rho \mathbf{q} c) = \nabla \cdot (\phi_f \rho D \cdot \nabla c) + \sigma \rho D_i \left. \frac{\partial c'}{\partial w} \right|_{w=0} \ , \tag{3}$$

$$\frac{\alpha \partial(\rho c')}{\partial t} = \frac{\partial}{\partial w}(\rho D_i \frac{\partial c'}{\partial w}) \ , \tag{4}$$

where

q is the specific discharge (or Darcy velocity) [LT^{-1}];

k is the effective permeability tensor due to the fractures carrying the flow [L^2];

μ is the groundwater viscosity (which depends on the salinity) [ML^{-1}T^{-1}];

P is the (total) pressure in the groundwater [ML^{-1}T^{-2}];

ρ is the groundwater density (which depends on the salinity) [ML^{-3}];

g is the gravitational acceleration [LT^{-2}];

t is the time [T];

ϕ_f is the effective porosity due to the fractures carrying the flow (which is sometimes referred to as the kinematic porosity) [–];

c is the salinity in the groundwater flowing through the fractures (expressed as a mass fraction) [–];

D is the (effective) dispersion tensor [L^{2}T^{-1}];

D_i is the intrinsic diffusion coefficient for diffusion into the rock matrix, which is referred to as the effective diffusion coefficient in the Swedish radioactive waste disposal programme [L^{2}T^{-1}];

σ is the specific fracture surface area, that is the average surface area of the matrix per unit volume [L^{-1}], which is called the specific flow-wetted surface area in the Swedish radioactive waste disposal programme. It is based on the volume of rock. (For smooth planar fractures, σ is given by $2P_{32}$, where P_{32} is the fracture area per unit volume, which is a measure of fracture intensity);

w is a coordinate in the rock matrix [L^{-1}];

c′ is the salinity of the groundwater in the rock matrix (expressed as a mass fraction) [–];

α is the capacity factor of the rock matrix [–].

It should be noted that, for clarity, the dependences of the salinities *c* and *c′* on position within the continuum representing the fractures are left implicit throughout this report.

In the above, Equation (1) is equivalent to Darcy's law, Equation (2) corresponds to conservation of groundwater mass, Equation (3) corresponds to conservation of groundwater salinity in the fractures (allowing for diffusion into the rock matrix) and Equation (4) is the basic equation for diffusion into the rock matrix.

In modelling the migration of salinity, the capacity factor in Equation (4) would normally be taken to be equal to the accessible porosity in the rock matrix, ϕ_m. However, it is envisaged that the development of CONNECTFLOW that is described here might also be used to model migration of other solutes, which might be sorbing. In order to allow for this, Equation (4) was written in the more general form using the capacity factor rather than the rock-matrix porosity. For a sorbing solute, the capacity factor would be given by

$$\alpha = R\phi_m \tag{5}$$

where R is the retardation due to sorption of the solute to the rock matrix [–]. The description in terms of the capacity factor also facilitates modelling possible cases in which salinity is excluded from part of the matrix porosity because of ionic effects.

The model above corresponds to a simple model of rock-matrix diffusion in terms of 1D diffusion into blocks of rock between parallel, equally spaced, planar fractures of constant equal aperture. Equation (4) can be modified in various ways. For example, variants of Equation (4) can be used to represent diffusion into blocks of rock between a network of intersecting fractures, or diffusion away from cylindrical flow channels. Equation (4) can also be modified to deal with diffusion into a layered medium with different properties for the different layers or to deal with diffusion in parallel into several different media. Either of these models could be used to represent the case in which there is a combination of diffusion from the water in the channels carrying flow into immobile water in the fractures together with diffusion into immobile water in the rock matrix between the fractures.

It should be stressed that the description of the model above should not be taken too literally. Although the model is described in terms of diffusion into the "rock matrix" this could represent, for example porous regions immediately adjacent to the fractures, rather than the full rock matrix.

The equations given above have to be supplemented by appropriate boundary and initial conditions. Suitable boundary conditions for the groundwater flow equations (Equations (1) and (2)) are prescriptions of either the groundwater pressure or the groundwater flux around the boundary of the domain modelled. Suitable boundary conditions for the equation for the transport of salinity (Equation (3)) are prescriptions of the salinity in the fractures at the domain boundary or the flux of salinity into the groundwater in the fractures. The boundary conditions for Equation (4) are that the salinity in the groundwater in the matrix at the fracture surface is equal to the salinity in the groundwater in the fractures locally:

$$c'(w = 0) = c \ ; \tag{6}$$

and that the flux of salinity in the matrix is zero at the maximum penetration depth d into the matrix:

$$-D_i \frac{\partial c'}{\partial w}(w = d) = 0 \ . \tag{7}$$

For example, if the model is used to represent diffusion into all of the rock matrix between the fractures,d would be taken to be equal to half the fracture spacing (because salinity could diffuse into the block from the fractures on either side of the block). The model could also be used to represent cases in which the distance that salinity can diffuse into the rock matrix is more limited.

As indicated, the density and the viscosity of the groundwater depend on the salinity, and Equations (1) to (4) have been written in a form that is valid when such dependences are taken into account. The forms of the dependences are generally given by empirical fits to data. For example, Herbert et al. /7/ suggested that the inverse of the density should be taken to be linearly dependent on salinity, and the viscosity could be taken to be exponential. If the salinity variations are small, such empirical fits may be approximated by linear relationships, and linear relationships are often used in modelling saline groundwater flows.

If the variation of the groundwater density in the rock matrix is taken into account, Equation (4) is non-linear, and can only be solved numerically (see subsection 3.1). However, if at a location in the continuum representing the fractures, the density in the rock matrix is independent of position within the rock matrix, Equation (4) for that location is then linear and can be solved analytically using Laplace transforms (see subsection 2.1) or by a series solution (see subsection 2.2). In many cases, it may be a reasonable approximation to take the groundwater density in the rock matrix at each location to be independent of the position within the rock matrix. This will be the case if the salinity is small everywhere, for example, which may be appropriate for at least some sites in Sweden. Even if the overall range of salinity is large at a site, the variations in salinity at each location may be small. For example, the salinity may generally increase with depth, with the variation in salinity for a particular elevation being relatively small. In such cases, it may be reasonable to make the approximation that the salinity in the rock matrix at a particular location is independent of position within the rock matrix. However, if the range of salinity at a site is large, the variations in salinity at a location may be large, because of the salinity migrating in the flow field. For example, there may be up-coning of salinity to an underground facility. Large variations in salinity at a location will probably lead to large variations in the salinity within the rock matrix.

It should be noted that neglecting the variation in the groundwater density within the rock matrix corresponds to approximating the diffusion process within the rock matrix. In fact, it corresponds to neglect of terms proportional to the temporal and spatial gradients of the density. However, the diffusion process is probably still represented in a plausible manner. Further, the effects of the density variations are probably smaller than the effects of the variation in the intrinsic diffusion coefficient that occur physically.

2.1 Laplace transform solution

If the groundwater density (at a particular location in the continuum representing the fractures) is independent of position within the rock matrix, Equation (4) with the boundary conditions (6) and (7) and initial condition

$$c'(t=0) = c'_0 \ , \tag{8}$$

(where c'_0 is a constant) can be readily solved in terms of Laplace transforms. Taking the Laplace transform of Equation (4) yields

$$\alpha\,(p\hat{c}' - c'_0) = D_i \frac{\partial^2 \hat{c}'}{\partial w^2} \ , \tag{9}$$

where ^ denotes the Laplace transform and p is the Laplace transform variable.

The general solution to Equation (9) can be expressed as

$$\hat{c}' = \frac{c'_0}{p} + Ae^{mw} + Be^{-mw} \ , \tag{10}$$

where

$$m = \sqrt{\frac{\alpha p}{D_i}} \ . \tag{11}$$

The solution that satisfies the boundary conditions is

$$\hat{c}' = \frac{c_0'}{p} + \left(\hat{c} - \frac{c_0'}{p} \right) \frac{\cosh[m(d-w)]}{\cosh[md]} \ . \tag{12}$$

It is convenient to introduce the response $f(w,t)$ to a unit step change in the concentration in the groundwater in the fractures. That is, $f(w,t)$ is the solution to the problem in which c' and c are initially zero, and then $c = 1$ for $t > 0$. In this case, the Laplace transform of c is

$$\frac{1}{p} \ , \tag{13}$$

and so from Equation (12)

$$\hat{f} = \frac{1}{p} \frac{\cosh[m(d-w)]}{\cosh[md]} \ . \tag{14}$$

Then, Equation (12) can be expressed as

$$\hat{c}' = c_0' \left(\frac{1}{p} - \hat{f} \right) + p\hat{c}\hat{f} \ . \tag{15}$$

Now, the Laplace transform of $\frac{\partial c}{\partial t}$ is

$$p\hat{c} - c(t=0) \ . \tag{16}$$

Therefore, the Laplace transforms in Equation (15) can be inverted to give

$$c' = c_0'(1-f) + c(t=0)f + \int_0^t \frac{\partial c(\tau)}{\partial \tau} f(t-\tau)d\tau \ , \tag{17}$$

using the result that the inverse of a product of Laplace transforms is a convolution (as in the last term on the right-hand side of Equation (17)).

The second and third terms on the right-hand side of Equation (17) can be derived directly using Duhamel's principle.

Further, from the above, the flux of salinity into the rock matrix is given by

$$-D_i \frac{\partial c'}{\partial w}(w=0) = -c_0' g + c(t=0)g + \int_0^t \frac{\partial c(\tau)}{\partial \tau} g(t-\tau)d\tau \ , \tag{18}$$

where

$$g = -D_i \frac{\partial f}{\partial w}(w=0) \ , \tag{19}$$

which has Laplace transform

$$\hat{g} = -D_i \frac{\partial \hat{f}}{\partial w}(w=0) = \frac{D_i m}{p} \tanh[md] = \sqrt{\frac{\alpha D_i}{p}} \tanh[md] \ . \tag{20}$$

The analysis above provides an explicit representation of the flux into the rock matrix (which is required in Equation (3)) in terms of a function g with a known Laplace transform. In general, this Laplace transform cannot be inverted in terms of a simple combination of elementary functions. However, in the case in which d tends to infinity, that is the times considered are so small that only a limited part of the accessible matrix is accessed,

$$\hat{g} = \sqrt{\frac{\alpha D_i}{p}} \quad , \tag{21}$$

and this can be readily inverted to give

$$g = \sqrt{\frac{\alpha D_i}{\pi t}} \quad . \tag{22}$$

The analysis above can be readily extended to deal with a different geometry, such as a cylindrical or spherical geometry, representing diffusion into blocks of rock between a network of intersecting fractures, or diffusion away from cylindrical flow channels. It can also be extended to deal with a layered medium.

2.2 Series solution

In the analysis in the previous subsection, which used a Laplace transform approach, an expression for *g* was developed that is particularly useful for small values of time. Using a series approach, it is possible to develop an alternative expression for *g* that is particularly useful at long times.

The series solution is developed using the method of separation of variables. It can readily be shown that a function of the form $W(w)\ T(t)$ is a solution of Equation (4) (with constant groundwater density) provided that

$$\frac{\partial^2 W}{\partial w^2} = -k_n^2 W \quad , \tag{23}$$

$$\frac{\partial T}{\partial t} = -\frac{D_i k_n^2}{\alpha} T \quad . \tag{24}$$

For non-zero k_n, the general solution to Equation (23) is

$$W_n = A_n \sin k_n x + B_n \cos k_n x \quad , \tag{25}$$

where A_n and B_n are constants. The solution that satisfies the homogeneous boundary condition

$$W(w = 0) = 0 \quad , \tag{26}$$

at $w = 0$ is

$$W_n = A_n \sin k_n x \quad . \tag{27}$$

In order to satisfy the homogeneous boundary condition

$$-D_i \frac{\partial W}{\partial w} = 0 \tag{28}$$

at $w = d$ it is necessary for k_n to be of the form

$$k_n = \frac{(n - \frac{1}{2})\pi}{d} \tag{29}$$

where *n* is an integer greater than or equal to 1.

The corresponding function T has the form

$$T_n = \exp\left\{-\frac{(n-\frac{1}{2})^2\pi^2 D_i t}{\alpha d^2}\right\} = \exp(-\omega_n t) \quad , \tag{30}$$

say.

For $k_n = 0$, the general solution to Equation (23) is

$$W_0 = A + Bw \ , \tag{31}$$

where A and B are constants. The solution that satisfies the (homogeneous) boundary condition at $w = d$ is

$$W_0 = A \ . \tag{32}$$

The corresponding function T has the form

$$T_0 = 1 \ . \tag{33}$$

Therefore, the function $c' = 1$ satisfies the boundary conditions and the governing equation. However, it does not satisfy the initial condition. It is therefore appropriate to seek the solution in the form

$$f(w,t) = 1 + \sum_{n=1}^{\infty} A_n \sin\frac{(n-\frac{1}{2})\pi w}{d} \exp\left\{-\frac{(n-\frac{1}{2})^2\pi^2 D_i t}{\alpha d^2}\right\} \ . \tag{34}$$

From the above, such a function satisfies the governing equation, and the boundary conditions, and can satisfy the initial condition ($f = 0$ for $w > 0$) for appropriate choice of the coefficients A_n. This requires

$$A_n = -\frac{2}{d}\int_0^d \sin\frac{(n-\frac{1}{2})\pi w}{d}\,dw = -\frac{2}{(n-\frac{1}{2})\pi} \ . \tag{35}$$

Equation (34) therefore gives the desired solution.

Then,

$$g(t) = -\sum_{n=1}^{\infty} \frac{(n-\frac{1}{2})\pi D_i A_n}{d} \exp\left\{-\frac{(n-\frac{1}{2})^2\pi^2 D_i t}{\alpha d^2}\right\} \ , \tag{36}$$

or

$$g(t) = \sum_{n=0}^{\infty} b_n \exp[-\omega_n t] \ , \tag{37}$$

say, where

$$b_n = -\frac{(n-\frac{1}{2})\pi D_i A_n}{d} = \frac{2D_i}{d} \ , \tag{38}$$

$$\omega_n = \frac{(n-\frac{1}{2})^2\pi^2 D_i}{\alpha d^2} \ . \tag{39}$$

The analysis above can be readily extended to the case of a different geometry, such as a spherical or cylindrical geometry, or to the case of a layered medium. The end result is that f is given by an expansion similar to Equation (34), but involving eigenfunctions appropriate to the problem being solved rather than sines. Then g is given by an expression similar to Equation (37). It should be noted, however, that the appropriate eigenvalue equation is generally much more complicated than Equation (29).

3 Possible approaches

In this Section, various approaches are outlined that might be used in CONNECTFLOW /1/ to model flow of groundwater with variable salinity through fractured rock using a continuum representation and allowing for diffusion into the rock matrix. All of the approaches are based on the use of standard finite-element discretisations to represent the (residual) groundwater pressure and the salinity in the groundwater in the (continuum representation of the) fractures (see Reference /1/). Thus, for example, in three dimensions, the residual pressure and the salinity might be represented by tri-linear functions on eight-node cuboid elements. The numerical representation of the continuum approximation for the flow in the fractures (Equations (1), (2) and (3) apart from its last term) is obtained as usual.

However, the various approaches represent rock-matrix diffusion in different ways. That is, effectively, they treat the last term in Equation (3) differently. It should be emphasised that all the approaches are based on Equation (4). It is worth noting that there are effectively two variants for each approach, in which the rock matrix is taken to be associated with respectively, the elements or the nodes of the flow model.

It is also worth noting, that although the various approaches are presented here in the context of finite-element models for the flow, similar approaches could be adopted in the context of finite-difference approaches for the flow.

3.1 Finite-difference (or finite-element) approach

The approach that is perhaps conceptually the simplest for dealing with rock-matrix diffusion in the context of a numerical flow model is to use numerical discretisation to solve Equation (4) for diffusion into the rock matrix. A simple finite-difference version of the approach is illustrated here. It should be stressed that in CONNECTFLOW, the flow equations are still being solved using a finite-element approach.

For each element (or each node, see above), in the finite-element flow model, the associated rock matrix is discretised into a number of 1D grid blocks. In a simple finite-difference (strictly finite-volume) discretisation, a discretised value of the salinity is associated with each grid block and time step. Then a simple fully implicit discretisation of Equation (4) is

$$\alpha \frac{\Delta w_j (\rho_j^l c_j'^l - \rho_j^{l-1} c_j'^{l-1})}{\Delta t^l} = \rho_{j-1}^l D_i \frac{(c_j'^l - c_{j-1}'^l)}{\frac{1}{2}(\Delta w_{j-1} + \Delta w_j)} - \rho_j^l D_i \frac{(c_{j+1}'^l - c_j'^l)}{\frac{1}{2}(\Delta w_j + \Delta w_{j+1})} \ , \qquad (40)$$

where the subscript denotes the grid block and the superscript denotes the time level, and

Δt^l is the size of the l-th time step;

Δw_j is the size of the j-th grid block.

Modified versions of Equation (40) hold at the ends of the domain. For the last (N-th) grid block

$$\alpha \frac{\Delta w_N (\rho_N^l c_N'^l - \rho_N^{l-1} c_N'^{l-1})}{\Delta t^l} = \rho_{N-1}^l D_i \frac{(c_N'^l - c_{N-1}'^l)}{\frac{1}{2}(\Delta w_{N-1} + \Delta w_N)} \ . \qquad (41)$$

This modifies the general Equation (40) to take into account the absence of a flux beyond the grid block, that is the boundary condition of Equation (7). For the first grid block

$$\alpha \frac{\Delta w_1 (\rho_1^l c_1'^l - \rho_1^{l-1} c_1'^{l-1})}{\Delta t^l} = \rho(c^l) D_i \frac{(c_1'^l - c^l)}{\frac{1}{2}\Delta w_1} - \rho_j^l D_i \frac{(c_2'^l - c_1'^l)}{\frac{1}{2}(\Delta w_1 + \Delta w_2)} \ . \qquad (42)$$

This modifies the general Equation (40) to take into account the boundary condition of Equation (6), that is, salinity diffuses into the matrix from the fracture, where the discretised salinity in (the continuum representation of) the fractures c'. (c' would therefore have additional indices to identify the element in the flow model that it is associated with. However, for clarity, such indices have been dropped here.)

The flux of salinity from the fractures into the rock matrix, which determines the last term in Equation (3), is given by

$$-D_i \frac{(c_1'^l - c^l)}{\frac{1}{2}\Delta w_j} \ . \qquad (43)$$

The discretisation presented above allows for the possibility that the sizes of the grid blocks may vary. For efficiency, it is likely to be necessary to exploit this, taking the grid blocks smaller near to the interface with the fractures. The discretisation also allows for the possibility that the time steps might vary, either in a manner specified by the user, or determined automatically by the program. This allows, in particular, the possibility that the time steps might be controlled by aspects of the flow in the fractures.

It might, at first sight, seem that in order to calculate the discretised groundwater pressure and salinity and the salinity in the rock matrix at the end of a time step, Equations (40), (41) and (42) have to be solved numerically together with the discretised versions of Equations (1), (2) and (3) to which they are coupled via Equation (43). It should be noted that several tens of discretised values may be necessary to represent the rock matrix for each element in the flow model (see below). Therefore, the numerical system for the model incorporating rock-matrix diffusion would be one to two orders of magnitude larger than the model for the flow system, and the computational cost of solving the numerical equations would apparently be several orders of magnitude larger than the computational cost of solving the equations without rock-matrix diffusion, even if it were possible to use the same time-step sizes. This would make the approach prohibitively expensive computationally for large models of the flow system.

However, it is not necessary to solve the discretised equations in this way, at least in the case in which, at each location in the continuum representing the fractures, the groundwater density is taken to be independent of position within the rock matrix. It is then possible to solve the equations as follows. The numerical equations for the rock-matrix associated with each element in the model of flow in the fractures can be solved in terms of the unknown salinity in the groundwater in the fractures at the end of the time step (see below for more detail). From this, the diffusive flux into the rock matrix can be expressed in terms of the unknown salinity in the fractures. This can then be included in the discretised version of Equation (3). This can be solved for the salinity in the fractures. Then, once the salinity in the fractures has been determined, the salinity in the rock matrix can be calculated. The numerical system for the salinity in the fractures has the same size as the numerical system for the model without rock-matrix diffusion. Therefore, provided that it is possible to use the same time-step sizes for the model with rock-matrix diffusion, the computational cost of solving the model with rock-matrix diffusion will be similar to that of solving the model without rock-matrix diffusion. It should be noted that (see below) the computational cost of dealing with the salinity in the rock matrix is much less than the computational cost of solving the discretised equations for flow in the fractures.

In detail, the approach is as follows. If the density is assumed to be constant, then the solution to Equations (40), (41) and (42) can be expressed as

$$c_j'^l = A_j c^l + B_j^l \ , \tag{44}$$

where A_j is the solution in the case in which $c_j'^{l-1} = 0$ and B_j^l is the solution in the case in which c^l is zero. B_j^l depends on $c_j'^{l-1}$, but not on c^l. A_j does not depend on $c_j'^{l-1}$, but is actually independent of l (as indicated by the absence of the superscript l).

Then the flux into the rock matrix is given by

$$-D_i \frac{(A_1 c^l + B_1^l - c^l)}{\frac{1}{2}\Delta w_1} \ . \tag{45}$$

In the l-th time step, the term in Equation (45) in B_1^l involves quantities that are known. It is only the terms in c^l that are not known. Therefore, as indicated above, if the flux into the rock matrix is represented as in Equation (45), its inclusion in the discretised version of Equation (3) does not increase the size of the numerical system, but simply modifies the coefficient of c^l in this system, and adds in a contribution to the right-hand side of the system. The modified system can then be solved for c^l. Then the salinity in the rock matrix at the end of the time step can be obtained simply by substituting the calculated value of c^l into Equation (44). The process can then be repeated for subsequent time steps. As indicated, this is a very efficient way to implement the algorithm.

3.2 Laplace transform approach

The approach outlined in the previous subsection uses a numerical approach to solve the equation for diffusion into the rock matrix (in conjunction with a standard numerical method for solving the equations for flow in the fractures). Another possible approach exploits the analytic solution for diffusion into the rock matrix in terms of Laplace transforms given in subsection 2.1, for the case in which, at each location in the continuum representing the fractures, the groundwater density is taken to be independent of position within the rock matrix and independent of time. The basic approach is described in subsection 3.2.1, and a refinement of the approach is described in subsection 3.2.2.

3.2.1 Basic approach

As shown in Equation (17), the flux of salinity into the rock matrix can be expressed in terms of a convolution of the time derivative of the salinity in the fractures and the function g (and in addition there are terms arising from the initial conditions in the matrix and the fractures). In most transient calculations with CONNECTFLOW, a fully implicit (Backward Euler) time-integration scheme is used. For this time-integration schemes (as for the fully explicit (Forward Euler) and Crank-Nicholson time integration schemes), the salinity in the fractures is effectively taken to vary in a piecewise linear manner with time. The flux into the rock matrix at time t^l is therefore given by

$$-D_i \frac{\partial c'}{\partial w}(w=0) = J^l(t^l) \ . \tag{46}$$

where

$$J^l(t) = -c_0' g(t) + c^0 g(t) + \sum_{m=1}^{l} \frac{(c^m - c^{m-1})}{(t^m - t^{m-1})} \int_{t^{m-1}}^{t^m} g(t-\tau)\,d\tau \quad \text{for } t^{l-1} \le t \le t^l \ . \tag{47}$$

This is applied for each element (or each node). However, as in subsection 3.1, the indices specifying the element (or node) have been dropped in Equation (47) for clarity.

The flux obtained from Equation (47) can be substituted into the discretised version of Equation (3) to give a fully implicit approach for the flux into the rock matrix. In the calculation for the l-th time step (from t^{l-1} to tl), only c^l in Equation (47) is unknown. All of the other quantities are known. Therefore, as in the finite-difference approach discussed in subsection 3.1, incorporating the flux into the rock matrix into the discretised equations does not increase the size of the system, but simply modifies the coefficient of c^l in this system, and adds in a contribution to the right-hand side of the system. The modified system can then be solved for c^l. The process can then be repeated for subsequent time steps. Again this is a very efficient way to implement rock matrix diffusion.

In order to implement the algorithm outlined above, values of g and certain definite integrals of this are needed. As discussed in subsection 2.1, in the general case, g cannot be expressed in terms of a simple combination of elementary functions. However, a simple expression for the Laplace transform of g has been developed and the required values of g and its integrals can be calculated by numerical inversion. Various techniques such as Talbot's algorithm /8/ can be used for this. If this has to be done for each element the computational effort involved might be significant, although for large problems, it would be much less than the effort in solving the numerical flow system. However, in many cases, the physical system of interest is represented in terms of a relatively small number of rock types with different properties. Then it is only necessary to carry out the calculations for these rock types and the computational effort involved is insignificant compared to the rest of the calculation.

Various aspects of the above algorithm should be noted. First, the distribution of salinity within the matrix is not actually calculated, in contrast to the finite-difference approach discussed in subsection 3.1. However, it would be possible to reconstruct the distribution of salinity at a specified time using the equivalent of Equation (17).

Second, the expression for the flux into the rock matrix gives the flux in terms of the salinity in the fractures at all previous time levels. It does not involve any additional quantities. Therefore, provided that the salinities in the fractures at all previous time levels have been kept in computer memory, the approach does not require any additional computer storage. In contrast to this, the finite-difference approach (see subsection 3.1) explicitly involves the salinities in the rock matrix, and additional storage has to be allocated for these. However, for a large problem, it might not be possible to keep the salinities in the fractures at all previous time levels in memory, and it would then be necessary to save these on backing files and read them in for each time step when needed.

3.2.2 Refined approach

The basic idea of the Laplace Transform approach has been outlined in the previous subsection. The basic method can be refined by using the average flux into the rock matrix over a time step rather than the flux at the end of the time step. The potential benefit of this approach is that it may allow much larger time steps to be taken. In particular, it may allow a time step to be taken that is so large that, within the time step, the salinity diffuses into all of the accessible porosity within the rock matrix. Over such a time step, the distribution of salinity within the rock matrix associated with a particular element would tend to a constant value, equal to the value in the fractures (which may vary over the time step). The flux into the rock matrix would therefore tend to zero over the time step. If the value of the flux at the end of the time step were used in the numerical version of Equation (3), then this would not take account of all of the salinity that had diffused into the rock matrix over the time step.

Using the average flux over the time step ensures that all of the salinity that has diffused into the rock matrix during the time step is taken into account properly. In fact, it ensures that for large time steps, the model behaves as if the effective porosity is given by the total accessible porosity (in the rock matrix and the fractures). A particular benefit of the refined approach is that it means that in a model with many different rock types and widely varying time scales, the time step is not unnecessarily constrained by the need to model diffusion into the rock matrix in part of the model that has a rapid response time, but where the system has in fact reached an effective equilibrium.

In the refined approach, the flux into the matrix in the discretised version of Equation (3) is modelled by

$$-D_i \frac{\partial c'}{\partial w}(w=0) = \frac{1}{(t^l - t^{l-1})} \int_{t^{l-1}}^{t^l} J^l(t')dt' \ , \qquad (48)$$

where $J^l\,(t^l)$ is given by Equation (47). Therefore

$$\begin{aligned}
-D_i \frac{\partial c'}{\partial w}(w=0) = &-\frac{c'_0}{(t^l - t^{l-1})} \int_{t^{l-1}}^{t^l} g(t')dt' + \frac{c^0}{(t^l - t^{l-1})} \int_{t^{l-1}}^{t^l} g(t')dt' + \\
&\sum_{m=1}^{l} \frac{(c^m - c^{m-1})}{(t^m - t^{m-1})} \frac{1}{(t^l - t^{l-1})} \int_0^{t^l} dt' \int_{t^{m-1}}^{t^m} g(t'-\tau)d\tau - \qquad (49) \\
&\sum_{m=1}^{l-1} \frac{(c^m - c^{m-1})}{(t^m - t^{m-1})} \frac{1}{(t^l - t^{l-1})} \int_0^{t^{l-1}} dt' \int_{t^{m-1}}^{t^m} g(t'-\tau)d\tau \ .
\end{aligned}$$

Noting that $g(t) = 0$ if $t < 0$, the order of the integrations in Equation (49) can be interchanged, leading to

$$\begin{aligned}
-D_i \frac{\partial c'}{\partial w}(w=0) = &-c'_0 \frac{(h(t^l) - h(t^{l-1}))}{(t^l - t^{l-1})} + c^0 \frac{(h(t^l) - h(t^{l-1}))}{(t^l - t^{l-1})} + \\
&\sum_{m=1}^{l} \frac{(c^m - c^{m-1})}{(t^m - t^{m-1})} \frac{1}{(t^l - t^{l-1})} \int_{t^{m-1}}^{t^m} h(t^l - \tau)d\tau - \qquad (50) \\
&\sum_{m=1}^{l-1} \frac{(c^m - c^{m-1})}{(t^m - t^{m-1})} \frac{1}{(t^l - t^{l-1})} \int_{t^{m-1}}^{t^m} h(t^{l-1} - \tau)d\tau \ ,
\end{aligned}$$

where

$$h(t) = \int_0^t g(t')dt' \ . \qquad (51)$$

The Laplace transform of $h(t) = \int_0^t g(t')dt'$ is given by

$$\hat{h} = \frac{\hat{g}}{p} = \frac{\sqrt{\alpha D_i}}{p^{\frac{3}{2}}} \tanh[md] \ , \qquad (52)$$

(see Equation (20)). Thus the values of $h(t)$ and its integrals that are required by the refined approach can be readily calculated by numerical inversion of Laplace transforms as for the basic Laplace transform approach (see subsection 3.2.1).

3.3 Hybrid approach

For the case in which at each location in the continuum representing the fractures the groundwater density is taken to be independent of position within the rock matrix, Carrera et al. /9/ have developed an approach that they call the hybrid approach. The basic idea of this approach is to use a combination of the analytic solutions for diffusion into the rock matrix discussed in subsections 2.1 and 2.2. The inverse square-root form of Equation (22) is used for small values of time and the series solution of Equation (37) is used for large values of time. It should be noted that the inverse square-root expression breaks down for large values of time, and the series solution requires many terms to give accurate results for small values of the time. Using the two solutions in combination offers the possibility of an accurate representation of diffusion into the rock matrix in terms of a relatively small number of quantities. The basic approach of Carrera et al. is outlined in subsection 3.3.1 to introduce the key ideas. In the original paper of Carrera et al., the approach is only presented for a constant step. The approach is generalised here to deal with a varying time step. Also certain aspects of the algorithm are clarified. Then, in subsection 3.3.2, a refinement of the basic hybrid algorithm along similar lines to the refinement of the basic Laplace transform approach discussed in subsection 3.2.2 is presented.

3.3.1 Basic hybrid approach

As indicated above, the basic idea of the hybrid algorithm is that g is approximated by

$$g(t) = \begin{cases} \sqrt{\dfrac{\alpha D_i}{\pi t}} & if \quad t < t^* \\ \sum\limits_{n=0}^{N} b_n \exp[-\omega_n t] & if \quad t > t^* \end{cases} \quad . \tag{53}$$

Key points that are not well explained in the original paper by Carrera et al. are that the matching time t^* has to be less than the time step size (the smallest time step size if the time steps are varying), and that N is determined by the requirement that the two expressions for g agree to within a certain specified accuracy at t^*. (In fact, Carrera et al. imply that t^* is determined by the requirement that the two expressions for g are equal at t^*, but this cannot be achieved.)

Substituting Equation (53) into Equation (47), the flux into the rock matrix is given by

$$\begin{aligned} -D_i \frac{\partial c'}{\partial w}(w=0) = J^l(t^l) = -c'_0 \sum_{n=1}^{N} b_n \exp[-\omega_n t^l] + c^0 \sum_{n=1}^{N} b_n \exp[-\omega_n t^l] + \\ \sum_{m=1}^{l-1} \frac{(c^m - c^{m-1})}{(t^m - t^{m-1})} \int_{t^{m-1}}^{t^m} \sum_{n=1}^{N} b_n \exp[-\omega_n (t^l - \tau)] d\tau + \\ \frac{(c^l - c^{l-1})}{(t^l - t^{l-1})} \int_{t^{l-1}}^{t^l - t^*} \sum_{n=1}^{N} b_n \exp[-\omega_n (t^l - \tau)] d\tau + \\ \frac{(c^l - c^{l-1})}{(t^l - t^{l-1})} \int_{t^l - t^*}^{t^l} \sqrt{\frac{\alpha D_i}{\pi (t^l - \tau)}} d\tau \quad . \end{aligned} \tag{54}$$

The integrals that appear in Equation (54) can be readily calculated, in terms of exponentials and powers. At first sight, it might appear that this is all that has been achieved by the use of the approximation of Equation (53), because Equation (54), like Equation (47) involves the salinities in the fractures at all previous time levels. However, Equation (54) has a very important property. If the quantities

$$I_n^l = \int_0^{t^l} \frac{\partial c(\tau)}{\partial \tau} \exp[-\omega_n (t^l - \tau)] d\tau = \sum_{m=1}^{l} \frac{(c^m - c^{m-1})}{(t^m - t^{m-1})} \int_{t^{m-1}}^{t^m} \exp[-\omega_n (t^l - \tau)] d\tau \ , \tag{55}$$

are introduced, then Equation (52) can be written as

$$\begin{aligned} -D_i \frac{\partial c'}{\partial w}(w=0) = -c_0' \sum_{n=1}^{N} b_n \exp[-\omega_n t^l] + c^0 \sum_{n=1}^{N} b_n \exp[-\omega_n t^l] + \\ \sum_{n=1}^{N} b_n \exp[-\omega_n (t^l - t^{l-1})] I_n^{l-1} + \\ \frac{(c^l - c^{l-1})}{(t^l - t^{l-1})} \sum_{n=1}^{N} b_n \frac{\exp[-\omega_n t^*] - \exp[-\omega_n (t^l - t^{l-1})]}{\omega_n} + \\ 2 \frac{(c^l - c^{l-1})}{(t^l - t^{l-1})} \sqrt{\frac{\alpha D_i t^*}{\pi}} \ , \end{aligned} \tag{56}$$

so that the flux can be expressed in terms of I_n^{l-1} and $\frac{(c^l - c^{l-1})}{(t^l - t^{l-1})}$ (rather than the salinities at all previous time levels).

Further, it can readily be seen that

$$\begin{aligned} I_n^l &= \int_0^{t^{l-1}} \frac{\partial c(\tau)}{\partial \tau} \exp[-\omega_n (t^l - \tau)] d\tau + \int_{t^{l-1}}^{t^l} \frac{\partial c(\tau)}{\partial \tau} \exp[-\omega_n (t^l - \tau)] d\tau \\ &= \exp[-\omega_n (t^l - t^{l-1})] \int_0^{t^{l-1}} \frac{\partial c(\tau)}{\partial \tau} \exp[-\omega_n (t^{l-1} - \tau)] d\tau + \\ &\int_{t^{l-1}}^{t^l} \frac{\partial c(\tau)}{\partial \tau} \exp[-\omega_n (t^l - \tau)] d\tau \\ &= \exp[-\omega_n (t^l - t^{l-1})] I_n^{l-1} + \frac{(c^l - c^{l-1})}{(t^l - t^{l-1})} \int_{t^{l-1}}^{t^l} \exp[-\omega_n (t^l - \tau)] d\tau \\ &= \exp[-\omega_n (t^l - t^{l-1})] I_n^{l-1} + \frac{(c^l - c^{l-1})}{(t^l - t^{l-1})} \frac{1 - \exp[-\omega_n (t^l - t^{l-1})]}{\omega_n} \ . \end{aligned} \tag{57}$$

Thus it is only necessary to use the latest values of the intermediate quantities I_n^{l-1} and $\frac{(c^l - c^{l-1})}{(t^l - t^{l-1})}$ in the calculation for a time step. Then once the salinities at the end of the time step have been calculated, the intermediate quantities can be updated from their current values and the calculated value of $\frac{(c^l - c^{l-1})}{(t^l - t^{l-1})}$ using Equation (57). Thus, rather than using the values of the salinities at all previous time levels, the algorithm only uses the latest values of the *N* intermediate quantities I_n^{l-1}. As indicated, this property, which is a consequence of the eigenfunctions in the expansion of Equation (34) having an exponential time dependence, is very important. If the number of time steps is large, the property could give a very substantial reduction in the amount of computer memory or I/O required compared to the requirements for the Laplace transform algorithm.

It should be noted that the intermediate quantities introduced above are not quite the same as those introduced by Carrera et al. /9/, although they are closely related.

It is possible to generalise the algorithm so that the matching time t^* can be chosen greater than the time step size. However, this may require the values of more quantities to be stored than is the case for the variant of the algorithm described above. The number N of intermediate quantities I_n^{l-1} that have to be kept is proportional to $\frac{1}{\sqrt{t^*}}$, which decreases as t^* increases. However, if t^* is greater than the time step size, all past salinities within an interval t^* of the current time have also to be kept. Often the latter more than compensates for the saving due to the reduction in N. Since the additional complexity associated with being able to choose t^* independent of the time step size does not bring substantial savings in the amount of computer memory or I/O required, this generalisation of the algorithm is not considered further.

3.3.2 Refined hybrid approach

As for the Laplace Transform approach, and for the same reasons (see subsection 3.2.2), the basic method outlined in the previous subsection can be refined by using the average flux into the rock matrix over a time step rather than the flux at the end of the time step. This leads to the following expression for the flux into the rock matrix

$$
\begin{aligned}
& -D_i \frac{\partial c'}{\partial w}(w=0) = -c'_0 \sum_{n=1}^{N} b_n \frac{\exp[-\omega_n t^{l-1}] - \exp[-\omega_n t^l]}{\omega_n (t^l - t^{l-1})} + \\
& c^0 \sum_{n=1}^{N} b_n \frac{\exp[-\omega_n t^{l-1}] - \exp[-\omega_n t^l]}{\omega_n (t^l - t^{l-1})} - \\
& \sum_{n=1}^{N} b_n \frac{\exp[-\omega_n (t^l - t^{l-1})] - \exp[-\omega_n t^*]}{\omega_n (t^l - t^{l-1})} I_n^{l-1} + \\
& \frac{(c^l - c^{l-1})}{(t^l - t^{l-1})} \sum_{n=1}^{N} \frac{b_n}{\omega_n (t^l - t^{l-1})} \left\{ \exp[-\omega_n t^*](t^l - t^{l-1} - t^*) - \frac{\exp[-\omega_n t^*] - \exp[-\omega_n (t^l - t^{l-1})]}{\omega_n} \right\} + \\
& 2 \frac{(c^l - c^{l-1})}{(t^l - t^{l-1})} \sqrt{\frac{\alpha D_i t^*}{\pi}} \frac{(t^l - t^{l-1} - t^*)}{(t^l - t^{l-1})} + \\
& 2 \frac{(c^l - c^{l-1})}{(t^l - t^{l-1})} \sqrt{\frac{\alpha D_i t^*}{\pi}} \frac{\frac{2}{3} t^*}{(t^l - t^{l-1})} - \\
& \sum_{n=1}^{N} b_n \frac{\exp[-\omega_n (t^{l-1} - t^{l-2} + t^*)] - \exp[-\omega_n (t^{l-1} - t^{l-2})]}{\omega_n (t^l - t^{l-1})} I_n^{l-2} + \\
& \frac{(c^{l-1} - c^{l-2})}{(t^l - t^{l-1})} \sum_{n=1}^{N} \frac{b_n}{\omega_n (t^l - t^{l-1})} \left\{ \exp[-\omega_n t^*] t^* - \frac{\exp[-\omega_n (t^{l-1} - t^{l-2})] - \exp[-\omega_n (t^{l-1} - t^{l-2} + t^*)]}{\omega_n} \right\} + \\
& 2 \frac{(c^{l-1} - c^{l-2})}{(t^l - t^{l-1})} \sqrt{\frac{\alpha D_i t^*}{\pi}} \frac{\frac{1}{3} t^*}{(t^l - t^{l-1})} \; .
\end{aligned}
\tag{58}
$$

For the first time step, the last three terms in Equation (58) are not present, and the first two terms are modified so that

$$-D_i\frac{\partial c'}{\partial w}(w=0) = -2c'_0\sqrt{\frac{\alpha D_i t^*}{\pi}}\frac{1}{t^1} + c'_0\sum_{n=1}^{N} b_n\frac{\exp[-\omega_n t^1]-\exp[-\omega_n t^*]}{\omega_n t^1} +$$

$$2c^0\sqrt{\frac{\alpha D_i t^*}{\pi}}\frac{1}{t^1} - c^0\sum_{n=1}^{N} b_n\frac{\exp[-\omega_n t^1]-\exp[-\omega_n t^*]}{\omega_n t^1} +$$

$$\frac{(c^1-c^0)}{t^1}\sum_{n=1}^{N}\frac{b_n}{\omega_n t^1}\left\{\exp[-\omega_n t^*](t^1-t^*) - \frac{\exp[-\omega_n t^*]-\exp[-\omega_n t^1]}{\omega_n}\right\} + \qquad (59)$$

$$2\frac{(c^1-c^0)}{t^1}\sqrt{\frac{\alpha D_i t^*}{\pi}}\frac{(t^1-t^*)}{t^1} +$$

$$2\frac{(c^1-c^0)}{t^1}\sqrt{\frac{\alpha D_i t^*}{\pi}}\frac{\frac{2}{3}t^*}{t^1} \;.$$

Again it is possible to organise the calculations for a time step so that it is only necessary to keep the latest values of the intermediate quantities I_n^l (in memory or on backing files).

3.4 Possible approximate Laplace transform approaches

The Laplace transform approach for treating diffusion into the rock matrix that is outlined in subsection 3.2 is accurate, easy to implement, and efficient. The only potential problem with the approach is that the calculation of the flux into the rock matrix involves the salinities in the fractures at all previous time levels. It is therefore necessary to store these values either in memory or on backing files. The latter is likely to be necessary for large models. In this case, the I/O time required to read the salinities into memory for each time step may be significant. It would not be expected that the time required for the steps in the calculation dealing with rock-matrix diffusion would be significant compared to the time required to solve the numerical system for the flow.

In order to reduce the potential memory or I/O requirement, consideration might therefore be given to the possibility of developing an approximation to the Laplace transform approach that involves fewer quantities. The basic idea would be to attempt to exploit the fact that the effect of the salinities in the fractures at a given time on the salinity in the rock matrix falls off over time. Various approaches might be used. One approach would be to combine the early time steps into groups and replace the detailed time variation of the salinity in the fractures for a group with a suitable average value. The detailed representation of the variation over time steps would only be used for the most recent time steps. Another possible approach might be to approximate the early time variation as an expansion in terms of a small number of suitable functions, such as polynomials.

However, all of the possible approaches are to some extent ad hoc. They probably would require significant work to determine suitable values for the parameters of the approaches, such as the size of the groups for the first suggested approach, or the functions and their number for the second suggested approach. None of the approaches would seem to offer any significant benefits compared to the hybrid approach developed by Carrera et al. /9/.

This approach is also an approximation, but the approximation is likely to be very accurate. Therefore, such approaches were not considered further.

3.5 Galerkin approach

Another approach that might be used to model the diffusion into the rock matrix is a Galerkin approach for solving the diffusion equation for each time step. In such an approach, the distribution of salinity in the rock matrix would be represented in terms of an expansion in terms of a number of suitable basis functions, which would generally be smooth. The coefficients in the expansion would be determined by requiring the expansion to satisfy, in some approximate sense, the equation representing rock-matrix diffusion for each time step.

Such an approach is closely related to a finite-element approach to solving the equation for diffusion into the rock matrix. In fact, a standard finite-element approach is basically a Galerkin approach that uses piecewise-smooth basis functions. The potential benefit of using a Galerkin approach with smooth basis functions is that, for a problem that is in a sense smooth, such a method would generally be expected to converge much more rapidly with increasing numbers of basis functions than a finite-element approach. Thus it might be possible to represent the diffusion into the rock matrix in terms of fewer quantities (the coefficients in the expansion) than would be required for a finite-element (or finite-difference) approach. The price that would be paid for this is that the equations that would be solved for the coefficients in the expansion for each time step would generally not be as sparse as the equations for a finite-element method, and might even be full. The equations would therefore require significantly greater computational effort to solve, although the work involved would probably still be significantly less than the work involved in solving the numerical system for the flow.

Further, the potentially faster convergence of the Galerkin approach only applies to smooth problems. However, one case that is likely to be of particular interest is the response to a step change in the concentration in the fractures (see subsection 2.1). For such a case, a Galerkin method may not converge more rapidly than a standard finite-element method, and may not even converge as rapidly.

Overall, it was considered that a Galerkin approach would not offer any significant benefits compared to the hybrid approach of Carrera et al. The latter approach essentially adopts a representation of the distribution of salinity in the rock matrix in terms of a number of suitable basis functions (sines, or suitable eigenfunctions in more general cases) combined with an alternative representation (the inverse square-root) to give a more accurate representation at early times than can be achieved using a relatively small number of the basis functions. A Galerkin approach would still involve a time-discretisation error. For these reasons, Galerkin approaches were not considered further.

3.6 The approach of Hopkirk and Gilby

Another approach for dealing with diffusion into the rock matrix has been proposed by Hopkirk and Gilby /10/. The approach was developed to deal with radionuclides, but could readily be modified to deal with salinity. The essence of their approach is as follows. The distribution of the concentration in the rock matrix associated with a grid block in a finite-difference discretisation of the flow system is represented as a finite sum of certain functions (cosines and hyperbolic cosines). The way in which this would evolve over a time step can be calculated analytically (using an approach based on Laplace transforms, in fact) for a linear variation over time of the concentration in the fractures. Then the solution at the end of the time step is approximated in terms of the chosen functions again. This is done by finding the best fit of a linear combination of these functions to the solution in the sense of minimising the square of the difference between the solution and the expansion. The fit is constrained in order to ensure conservation of mass.

Overall, it was considered that this approach would not offer any significant benefits compared to the hybrid approach of Carrera et al. Although the analytic calculation of the solution over a time step is exact, the process of approximating the solution in terms of the chosen functions at each time step introduces an error, analogous to the discretisation error in a finite-element, finite-difference or Galerkin approach. The hybrid approach is likely to be more accurate. It is also considered to be somewhat simpler to implement. For these reasons, the approach of Hopkirk and Gilbey was not considered further.

4 Comparison of approaches

In the previous Section, various methods that might be used for modelling diffusion into the rock matrix are outlined. There are many similarities between the different methods. All of the methods exploit the fact that the equation for diffusion into the rock matrix can to a large extent be solved separately from the equations for the flow through the fractures. The equation for diffusion into the rock matrix associated with an element in the finite-element model for flow in the fractures only involves the salinity of that element. In the case in which the groundwater density at each location is taken to be independent of position within the rock matrix, the solution to the rock-matrix diffusion equation for a time step can therefore be expressed as a linear combination of the salinity in the fractures at the end of the time step and other known quantities. The flux into the rock matrix can therefore be expressed in a similar manner. Such a term can be readily included in the equation for transport of salinity. This does not increase the size of the numerical system for the flow compared to that for flow without rock-matrix diffusion. It simply modifies the coefficient of the salinity in the numerical system and adds a contribution to the right-hand side of the system. In many cases, these changes do not significantly increase the computational effort required to solve the numerical system for the flow.

The differences between the approaches are therefore associated with the different ways in which the distribution of salinity in the rock matrix is represented and diffusion into the rock matrix is handled, and the implications of these. The potential key issues are the ease of implementation, computational cost, storage, flexibility and accuracy of the methods. A detailed comparison of these issues for the three main methods discussed in Section 3 (the finite-difference approach, the refined Laplace transform approach and the refined hybrid approach) is given below.

4.1 Ease of implementation

All of the main approaches discussed in Section 3 can be readily implemented in the finite–element groundwater flow and transport program CONNECTFLOW (or other groundwater flow and transport programs). The code in CONNECTFLOW needs to be modified in two places (see also Section 5). The subroutines that describe the flow equations that are being modelled need to be modified to include the extra term representing the flux into the rock matrix. As indicated above, and as discussed in Section 3, for each time step this term has the form

$$Ac^l + B^l \ . \qquad (60)$$

where c^l is the salinity at the end of the time step and A and B^l do not depend on c^l. (In practice (see Section 5), it may be convenient to implement this as a term of the form

$$A\frac{(c^l - c^{l-1})}{(t^l - t^{l-1})} + B'^l \ , \qquad (61)$$

where B'^l like B^l does not depend on c^l.) Such a term can be very easily implemented, particularly in CONNECTFLOW, which was originally designed to allow the equations that are being modelled to be specified or changed through FORTRAN subroutines that are closely related to the mathematical equations.

The second place where CONNECTFLOW needs to be changed is in the subroutine that loops over the time steps. This needs to be modified to calculate the quantities A and B^l (or A and B'^l) for each element and arrange for these to be passed to the subroutines that specify the flow equations. This change can be implemented in a very modular way via a call to a suitable subroutine. Given the current version of CONNECTFLOW (which is written in FORTRAN 77), it is easiest to use backing files to pass the quantities A and B^l (or A and B'^l) to the subroutines that specify the equations. (However, were a version of CONNECTFLOW to be implemented using a later version of FORTRAN, such as FORTRAN 90, it would be possible to use an alternative approach such as a suitable MODULE.) Again this is easy to implement.

The only differences between the various approaches therefore lie in the implementation of the calculation of A and B^l (or A and B'^l) for each element and each time step. The nature of these calculations is different for the different approaches. For the finite-difference approach, it is necessary to solve a number of matrix equations (see subsection 3.1). The matrix in question corresponds to the 1D discretisation of the diffusion into the rock matrix and is tri-diagonal. It is easy to implement ab initio an algorithm for the solution of such systems using, for example, the Thomas algorithm (which is a variant of Gaussian elimination). Indeed this was done in the testing of the accuracy of the various algorithms (see subsection 4.4). Versions of suitable algorithms can also be readily found in many subroutine libraries.

For the Laplace transform approach, it is necessary to invert numerically a number of Laplace transforms in order to calculate and B^l (or A and B'^l) for each element and each time step (see subsection 3.2). Versions of suitable subroutines for numerical inversion of Laplace transforms can be found in many subroutine libraries. At Serco Assurance, a variant of Talbot's algorithm /8/ has previously been used for numerical inversion of Laplace transforms in, for example, the MASCOT PSA program. If the Laplace transform approach were to be adopted for modelling rock-matrix diffusion in transport of salinity, then the subroutines used could be readily adapted.

For the hybrid approach, in order to calculate A and B^l (or A and B'^l) for each element and each time step, it is only necessary to carry out calculations involving exponentials and powers (see subsection 3.2). This is straightforward to implement.

Therefore, all the main approaches considered could be readily implemented.

It is perhaps worth adding a note here about the implementation of the program MATDIF /11/, based on NAMMU, which was previously developed to model rock matrix diffusion of radionuclides. MATDIF used a finite-element approach to model diffusion into the rock matrix. Such an approach is very similar to the finite-difference approach outlined in subsection 3.1. However, in MATDIF, the approach was implemented in a very different and more complicated way than that proposed above for the finite-difference approach. The basic idea was that NAMMU was used to solve the equations for diffusion into the rock matrix as well as the transport equations. However, the way in which this was implemented meant that NAMMU had to be modified to deal with many more variables than the standard version of NAMMU allowed at the time MATDIF was implemented and NAMMU also had to be modified to deal with many different models. The first of these required many changes to the code.

For the implementation proposed above for the finite-difference approach outlined in subsection 3.1, the variables associated with the distribution of salinity (concentration) in the rock matrix are separate from the variables characterising the flow and transport and are not handled by CONNECTFLOW. The implementation would therefore require few changes to CONNECTFLOW and would be much more modular than the implementation of MATDIF.

4.2 Computational cost

The overall computational cost of the various approaches depends on the number of time steps and on the cost of the calculations for each time step. The latter has two components: the cost of the calculation of the groundwater flow and transport in the fractures and the cost of the calculation for diffusion into the rock matrix.

First, it should be noted that all of the approaches outlined in Section 3 lead to a similar numerical system for flow and transport in the fracture provided that the time-step size is the same. The computational cost of solving this system is likely to be essentially the same for all of the approaches.

Further, it should be noted that the numerical system for flow and transport in the fractures is the same size as the numerical system in the case without rock matrix diffusion. For some techniques that might be used for solving the numerical system for flow and transport in the fractures, the cost of the calculation of groundwater flow and transport in the fractures with rock-matrix diffusion might be the same as the cost of the calculation in the case without rock-matrix diffusion. This might be the case if the direct frontal solver in CONNECTFLOW were to be used to solve the equations for example.

However, if the preconditioned conjugate gradient solver were to be used, then the cost of the calculations in the case with rock-matrix diffusion might be different from the cost of the calculations in the case without rock-matrix diffusion. It might be argued that the cost of the calculations would lie between the costs of two different calculations without rock matrix diffusion. In one of these the effective porosity would be taken to be the fracture porosity and in the other the effective porosity would be taken to be the total porosity of the matrix and the fractures. However, it is considered that this would not be a reliable guide to the cost. This is because of the way in which the pre-conditioned conjugate gradient solver converges.

It should also be noted that taking diffusion into the rock matrix into account will effectively change the non-linearity of the system. If the non-linearity is handled by iterations at each time step, then the number of iterations could be different in the cases in which rock-matrix diffusion is modelled and in which it is not. This would affect the cost of the calculations. Again, it is difficult to predict the likely effect of taking rock matrix diffusion into account in a particular case.

For each time step, all of the approaches for dealing with diffusion into the rock matrix outlined in Section 3 involve the calculation of the quantities A and B^l (or A and B'^l) for each element from a number of intermediate quantities. The overall number of floating point operations involved will be equal to the number of elements times a number that might be of the order of several hundred. For large models, this computational cost will be a small fraction of the computational cost of solving the numerical system for flow and transport in the fractures. Thus this contribution to the overall computational cost is not likely to differentiate between the various approaches. The worst case would be that of the Laplace transform approach for a heterogeneous model in which each element has different rock matrix diffusion properties. For each time step, it would then be necessary to carry out a large number of numerical Laplace transform inversions: one for each element and each previous time step. On average, the number of inversions required for each time step would be about half the total number of time steps times the number of elements. The cost of this many inversions might be significant for a simple implementation of the inversion algorithm. However, this case is unlikely to arise in practice because the variation of the rock-matrix diffusion properties is likely to be specified in terms of a number of rock types. Further, it would be possible to attempt to develop a variant of the inversion algorithm tailored to the approach, with much reduced computational cost, although this would increase the difficulty of implementing the rock-matrix diffusion algorithm.

The key issue in comparing the overall computational costs of each approach is then the extent to which each approach may constrain the allowable time steps. This leads to a potentially important distinction between the various methods. In the case of the basic Laplace transform approach and the basic hybrid approach, in a model with many different rock types and widely varying time scales, the time step may be constrained by the need to model rock-matrix diffusion in part of the model that has a rapid response time, even though this part of the model is effectively in equilibrium. The refined variants of the basic approaches were specifically designed to ensure that the model for rock-matrix diffusion would be reasonable for a very large time step. This would also be the case for the finite-difference approach, provided that a fully implicit time-integration method is used for this. Therefore, the refined variants of the basic approach and the fully implicit finite-difference approach might allow much larger time steps for some models. This would lead to a much lower overall computational cost. Therefore, these methods are preferred.

4.3 I/O and storage

All the approaches considered utilise certain intermediate quantities for each element in the calculation of the quantities A and B^l (or A and B'^l) for each element and each time step. In the case of the finite-difference approach, the intermediate quantities are the salinities of the grid blocks in the finite-difference representation of the rock matrix associated with the element. In the case of the Laplace transform approach, the intermediate quantities are the salinity in the fractures for the element for all previous time steps. In the case of the hybrid approach, the intermediate quantities are the quantities I_n^l defined in Equation (54).

For each approach, the intermediate quantities have either to be stored in memory or on backing file during the computation. For a small model, it will be possible to keep all of the intermediate quantities in memory, but for a large model it will be necessary to use a backing file. The I/O time to transfer the information into memory for each time step could then be significant.

The I/O time is effectively proportional to the number of intermediate quantities. Therefore, the number of intermediate quantities required for each element was estimated for each approach. In the case of the finite-difference approach, a preliminary estimate of the number of intermediate quantities for each element was obtained on the basis that the grid blocks in the finite-difference model should be small enough to be capable of giving a reasonably accurate representation of the flux into the rock matrix during a time step. For a uniform grid, this requires the size of the grid blocks to be similar to the distance salinity would diffuse into the matrix in a time step:

$$\Delta x = \sqrt{\frac{D_i \Delta t}{\alpha}} \quad . \tag{62}$$

The total number of grid blocks would then be

$$N = \frac{d}{\Delta x} = \sqrt{\frac{d^2 \alpha}{D_i \Delta t}} = \sqrt{\frac{t_d}{\Delta t}} \quad , \tag{63}$$

where t_d is the time to diffuse fully across the rock matrix.

Using the above analysis for realistic parameters, it is estimated that several tens of intermediate quantities would be required for each element in a typical case. This analysis is a very simple order-of-magnitude estimate and the actual number of grid blocks required to give acceptable accuracy might be somewhat larger (see also subsection 4.4). It is also possible that somewhat fewer grid blocks might be necessary if a variable grid spacing is used, with the smallest spacing adjacent to the fractures. (For comparison, in calculations with MATDIF, it was found that for each finite element in the representation of the fractures, between about twenty and fifty elements with refinement close to the fractures, were necessary to give a sufficiently accurate representation of the rock matrix /12/.) It should be noted that, for a simple implementation of the approach with the same value of N used throughout, in a model with widely varying properties, the value of N might be determined by part of the model.

For the Laplace transform approach, the number of intermediate quantities for each element is equal to the number of previous time steps. On average, therefore, the number of intermediate quantities for each element is half the total number of time steps. In a typical case, this could be several hundred.

The number of intermediate quantities required for the hybrid approach was also estimated. As discussed in subsection 3.3, the matching time t^* has to be smaller than the time step size and then N is determined by the requirement that the series in Equation (50) gives a good match to the inverse square-root at t^*. Interestingly, this leads to the requirement that N is of the same order as the value given by Equation (63). Again, it should be noted that, for a simple implementation of the approach with the same value of N used throughout, in a model with widely varying properties, the value of N might be determined by part of the model.

Overall therefore, the finite-difference approach and the hybrid approach might require fewer intermediate quantities than the Laplace transform approach. The I/O time required to read the intermediate quantities into memory for each time step would therefore probably be less for the finite-difference approach and the hybrid approach than for the Laplace transform approach.

It is of interest to estimate the I/O times that might be involved. For example, in the version 1.1 of the Forsmark model that has recently been developed for SKB there are approximately 400,000 nodes. For a typical run with about 500 time steps, for each time step, it would on average be necessary to read about 100 Mwords (800 Mb) into memory for the Laplace transform method, if the intermediate quantities are kept on backing file. For the computer currently in use at Harwell, the maximum speed for reading from disk is about 100 Mb per sec. The time to read the intermediate quantities would therefore be about 10 sec for each time step and the total time for reading the intermediate quantities during the run would be about 1hr.

For comparison, the overall time required on the Harwell computer for one time step using the preconditioned conjugate gradient solver is about 175 secs (3 mins) and the overall run time is about 24 hours. The time for each step is divided as follows:

(i) about 25 secs to assemble the equations for each variable (pressure and concentration);

(ii) about 25 secs to calculate the preconditioner for each variable (pressure and concentration);

(iii) about 75 secs to solve the equations.

Overall, therefore, the I/O time for the Laplace transform approach would be a small but significant fraction of the total run time. The I/O times for the finite-difference approach and the hybrid approach are likely to be significantly less than the I/O time for the Laplace transform approach.

4.4 Accuracy

The accuracy of the various approaches is potentially a key factor that might distinguish between them. The Laplace transform approach essentially treats the diffusion into the rock matrix exactly. For suitable choice of the parameters (t^* and N) the hybrid approach will give an extremely accurate representation of diffusion into the rock matrix. Careful consideration was therefore given to the accuracy of the finite-difference method. This had been considered to some extent in estimating the number of grid blocks required, which had been done on the basis that the grid blocks should be sufficiently small as to be capable of giving an accurate representation of the flux into the rock matrix during a time step. However, the analysis was crude. A more detailed study of the accuracy of the method was therefore undertaken. A simple implementation of the 1D finite-difference method as a FORTAN program was made and tested. Then the accuracy of the method was examined in the case in which the salinity in the rock matrix is initially zero and the salinity in the fractures is immediately changed to a value of 1. (This is the case used to derive the response function g of subsection 2.1.)

The accuracy of the flux into the rock matrix at the end of the first time step was considered for a wide range of possible grid-block sizes. It was found that, for small grid block sizes, the flux at the end of the first time step was too large by a factor of about 1.77, even though the finite-difference grid blocks were small enough to represent the correct value of the flux. Conversely, for large grid-block sizes the flux at the end of the first time step is too small.

These results are not surprising. The finite-difference approach includes spatial and temporal discretisation errors. In the case in which the grid blocks are very small, so that the spatial discretisation error is negligible the temporal discretisation error is still present, and in the case in which the grid blocks are large, the spatial discretisation error will dominate.

The results were checked by solving analytically the numerical equations for the finite-difference approach. It was found that, for small grid-block sizes, the flux at the end of the first time step should be overestimated by a factor of $\sqrt{\pi}$ =1.7725, and that, for large grid-block sizes, the flux at the end of the first time step should be underestimated by a factor of

$$4\sqrt{\frac{\pi D_i \Delta t}{\alpha \Delta x^2}} \quad , \tag{64}$$

exactly in accord with the observations. It was also possible to estimate a value of the grid block size, or equivalently the number of grid blocks, that would give an accurate result for the flux at the end of the first time step. The optimum value of the number of grid blocks was found to be

$$N = \sqrt{\frac{\alpha d^2}{4(\pi - 1) D_i \Delta t}} \quad . \tag{65}$$

(Effectively the spatial and temporal discretisation errors cancel for this choice of parameters.)

(A similar result was obtained using a finite-element approach to determine the flux into the rock matrix, although the optimum value of the number of grid blocks was slightly different in this case.)

These results have important implications for the method. If the rock-matrix diffusion model were to be solved independently of the equations for flow in the fractures and with separate time steps, then it would be possible to choose the time-step and grid-block sizes for the rock-matrix diffusion model sufficiently small to achieve any specified accuracy, although this could mean that the computational cost for the rock-matrix diffusion model would be significant. This would also be significantly more complicated to implement than the approach that is proposed for the initial development of an option in CONNECTFLOW to model rock-matrix diffusion during transport of salinity. In the proposed approach, the same time step would be used for modelling diffusion into the rock matrix and flow in the fractures.

It might at first be envisaged that it would be possible to choose the grid-block size (or equivalently the number of grid blocks) to ensure that the number of grid blocks is close to optimum. However, it is expected that it will be necessary to vary the time step throughout the run, so that it would not be possible for the grid-block size to be optimum throughout the run. Also, the number of grid blocks could be different for different elements, which would complicate the implementation slightly.

It might also be hoped that the error in the flux into the matrix would fall off rapidly with time, or that the accumulated error would become small over time. However, this is not the case as shown by Figures 4-1 and 4-2. In Figure 4-1, the variation of the flux into the rock matrix over time is shown for a case in which the grid block size is small. As can be seen there is always a significant error in the flux, although this error does decrease over time. In Figure 4-2, the integrated flux (i.e. the total salinity that has diffused into the rock matrix) is shown. Again there is always a significant error, although this decreases over time.

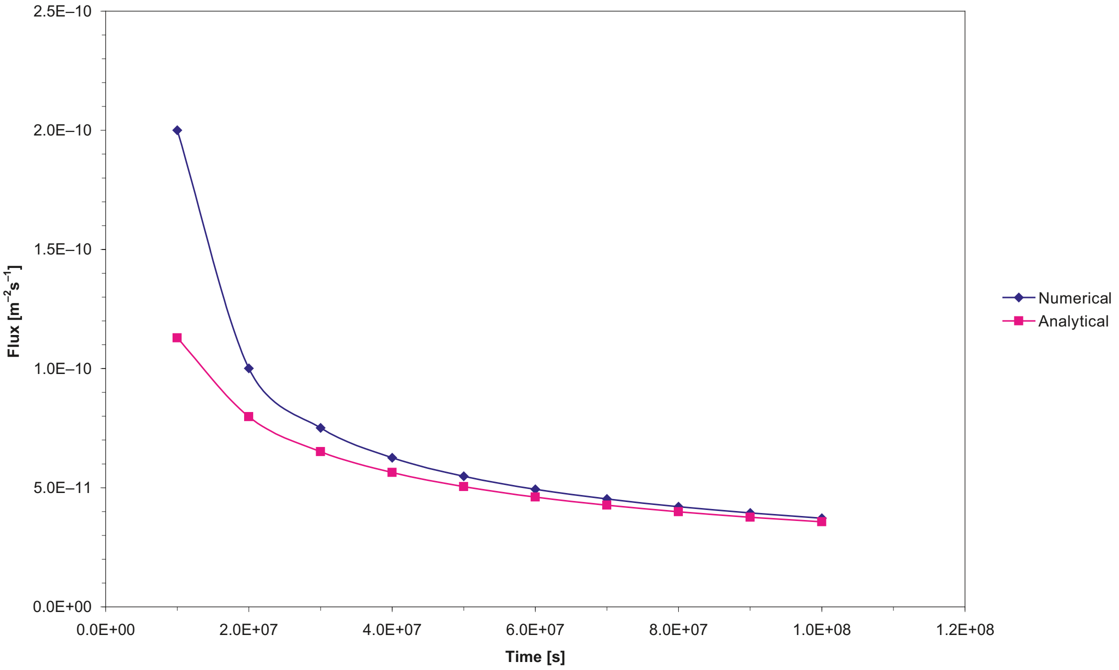

Figure 4-1. *Comparison of the flux into the rock matrix calculated using a simple finite-difference approach with that calculated analytically.*

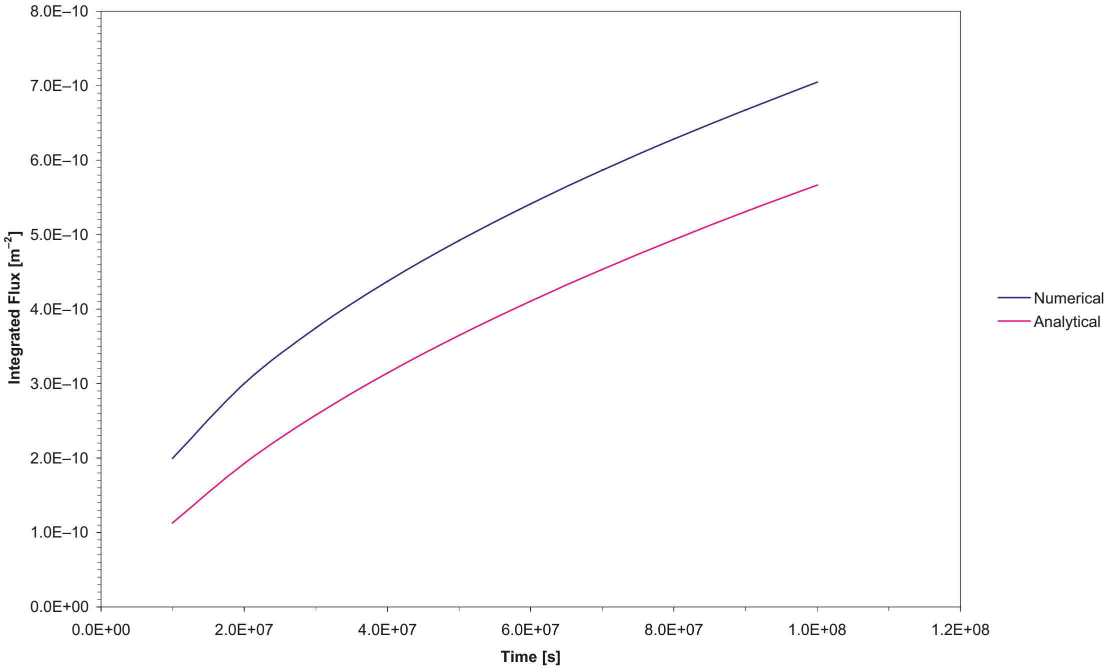

Figure 4-2. *Comparison of the integrated flux into the rock matrix calculated using a simple finite-difference approach with that calculated analytically.*

In conclusion, it would appear that in a simple implementation of the finite-difference approach that uses the same time step as the discretisation of the equations for flow in the fractures, the finite-difference approach is likely to have a significant error in the flux into the matrix. It is possible that the effect of this error might be small compared to the effect of the discretisation errors associated with the representation of flow in the fractures. However, recognising that the extent to which salinity diffuses into the rock matrix effectively determines the speed at which salinity moves through the fractures, it was considered that the errors in the finite-difference approach might have a significant effect on the flow in the fractures.

4.5 Flexibility

The key point here is that most of the methods outlined in Section 3 can only deal with the case in which the groundwater density in the rock matrix is taken to be constant. Only the finite-difference method (and possibly the Galerkin method) can deal with the case in which the groundwater density in the rock matrix is variable. This case is non-linear. All the other methods can only handle linear problems. Therefore, in order to use the methods it is necessary to make the approximation that the groundwater density in the rock matrix is constant. However, for many of the models that are likely to be used for SKB, this approximation is likely to be reasonable, because the salinities involved range from that of fresh water to that of seawater, which at present has a density about 2.5% greater than that of fresh water (and also see Section 2).

The three main approaches considered could be extended to deal with radioactive decay or with diffusion into a layered rock matrix. However, this would be complicated for the hybrid approach outlined in subsection 3.3. The eigenfunctions that would appear in the expansion similar to that of Equation (34) would be much more complicated than the simple sine functions of Equation (34), and the eigenvalues would in general be determined by complicated transcendental equations.

4.6 Evaluation

On the basis of the comparison above, it was considered that the best approach to implement initially in CONNECTFLOW for SKB would be the refined version of the hybrid approach (see subsection 3.3.2). This decision was made mainly on the basis that the method would provide a highly accurate representation of diffusion into the rock matrix. The method is also likely to involve the least number of intermediate quantities and hence the least I/O time for large models (see subsection 4.3). The method was preferred to the Laplace transform approach, which is also accurate, because the latter may have a significantly larger I/O time.

The hybrid method is also easy to implement. The use of the refined version rather than the basic version originally developed by Carrera et al. /9/ should give the greatest likelihood that the time-step sizes are not unduly constrained (see subsection 3.3.2) in cases in which the model involves wide ranges of properties and hence time scales. It is recognised that the hybrid method is not as flexible as the finite-difference approach, and in particular cannot deal with the (non-linear) case in which the variation of the groundwater density in the rock matrix with salinity is taken into account.

The finite-difference approach is the only method that is sufficiently flexible to handle the non-linear case in which the variation of the groundwater density in the rock matrix with salinity is taken into account. For this reason, the method was initially preferred. However, in view of the limits on the accuracy of the approach discussed in subsection 4.5, the hybrid approach was ultimately preferred, noting that the variations in groundwater density that are likely to be modelled for SKB are small.

It was also noted that there are many common features in the implementation of the various methods. Therefore, once the hybrid approach has been implemented, if it were desired to implement the finite-difference approach at some time in the future, this would not involve a lot of work.

5 Implementation in CONNECTFLOW

In the previous Sections, various alternative approaches for modelling rock-matrix diffusion in CONNECTFLOW for transport of salinity were discussed. It was recommended that the best approach to implement initially would be the refined version of the hybrid approach (see subsection 3.3.2). In this Section, details of the implementation of this algorithm are given.

The algorithm can be summarised as follows:

(i) The equation for the transport of salinity is modified to include a term for the flux into the matrix (see Equation (3))

$$\frac{\partial(\phi_f \rho c)}{\partial t} + \nabla \cdot (\rho \mathbf{q} c) = \nabla \cdot (\phi_f \rho D \cdot \nabla c) + \sigma \rho D_i \left.\frac{\partial c'}{\partial w}\right|_{w=0} . \tag{66}$$

(ii) The last term in Equation (66) can be written as (see Equation (59))

$$\sigma \rho D_i \left.\frac{\partial c'}{\partial w}\right|_{w=0} = A \frac{(c^l - c^{l-1})}{(t^l - t^{l-1})} + B'^l . \tag{67}$$

(iii) For time steps after the first, the coefficients A and B'^l are determined from Equation (58). (For the first time step, similar expressions can be derived from Equation (59).) The recursion relation (see Equation (57))

$$I_n^l = \exp[-\omega_n (t^l - t^{l-1})] I_n^{l-1} + \frac{(c^l - c^{l-1})}{(t^l - t^{l-1})} \frac{(1 - \exp[-\omega_n (t^l - t^{l-1})])}{\omega_n} \tag{68}$$

can be used to simplify Equation (58) giving:

$$\begin{aligned}
&- D_i \frac{\partial c'}{\partial w}(w=0) = \left(c^0 - c'_0\right) \sum_{n=1}^{N} b_n \frac{\exp[-\omega_n t^{l-1}] - \exp[-\omega_n t^l]}{\omega_n (t^l - t^{l-1})} + \\
&\sum_{n=1}^{N} b_n \frac{1 - \exp[-\omega_n (t^l - t^{l-1})]}{\omega_n (t^l - t^{l-1})} I_n^{l-1} + \\
&\frac{(c^l - c^{l-1})}{(t^l - t^{l-1})} \sum_{n=1}^{N} \frac{b_n}{\omega_n (t^l - t^{l-1})} \left\{ \exp[-\omega_n t^*](t^l - t^{l-1} - t^*) - \frac{\exp[-\omega_n t^*] - \exp[-\omega_n (t^l - t^{l-1})]}{\omega_n} \right\} + \\
&2 \frac{(c^l - c^{l-1})}{(t^l - t^{l-1})} \sqrt{\frac{\alpha D_i t^*}{\pi}} \frac{(t^l - t^{l-1} - \tfrac{1}{3} t^*)}{(t^l - t^{l-1})} + \\
&\frac{(c^{l-1} - c^{l-2})}{(t^{l-1} - t^{l-2})} \sum_{n=1}^{N} \frac{b_n}{\omega_n (t^l - t^{l-1})} \left\{ \exp[-\omega_n t^*] t^* - \frac{1 - \exp[-\omega_n t^*]}{\omega_n} \right\} + \\
&2 \frac{(c^{l-1} - c^{l-2})}{(t^{l-1} - t^{l-2})} \sqrt{\frac{\alpha D_i t^*}{\pi}} \frac{\tfrac{1}{3} t^*}{(t^l - t^{l-1})} .
\end{aligned} \tag{69}$$

from which A and B'^l can be derived.

(iv) Hence, for time steps after the first, *A* is given by:

$$A = -\sigma\rho\sum_{n=1}^{N}\frac{b_n}{\omega_n(t^l - t^{l-1})}\left\{\exp[-\omega_n t^*](t^l - t^{l-1} - t^*) - \frac{\exp[-\omega_n t^*] - \exp[-\omega_n (t^l - t^{l-1})]}{\omega_n}\right\} -$$

$$2\sigma\rho\sqrt{\frac{\alpha D_i t^*}{\pi}}\,\frac{(t^l - t^{l-1} - \tfrac{1}{3}t^*)}{(t^l - t^{l-1})}\ . \qquad (70)$$

and B'^l is given by:

$$B'^l = -\sigma\rho\left(c^0 - c'_0\right)\sum_{n=1}^{N} b_n\,\frac{\exp[-\omega_n t^{l-1}] - \exp[-\omega_n t^l]}{\omega_n(t^l - t^{l-1})} -$$

$$\sigma\rho\sum_{n=1}^{N} b_n\,\frac{1 - \exp[-\omega_n (t^l - t^{l-1})]}{\omega_n(t^l - t^{l-1})}\,I_n^{l-1} -$$

$$\sigma\rho\,\frac{(c^{l-1} - c^{l-2})}{(t^{l-1} - t^{l-2})}\sum_{n=1}^{N}\frac{b_n}{\omega_n(t^l - t^{l-1})}\left\{\exp[-\omega_n t^*]t^* - \frac{1 - \exp[-\omega_n t^*]}{\omega_n}\right\} - \qquad (71)$$

$$2\sigma\rho\,\frac{(c^{l-1} - c^{l-2})}{(t^l - t^{l-1})}\sqrt{\frac{\alpha D_i t^*}{\pi}}\,\frac{\tfrac{1}{3}t^*}{(t^l - t^{l-1})}\ .$$

The following aspects of the current implementation should be noted. First, in dealing with the salinity in the rock matrix, the groundwater density at all points in the rock matrix is taken to have the value of the reference groundwater density in CONNECTFLOW. This involves a greater approximation than is strictly necessary. As discussed in Section 2, for example, it is really only necessary to make the approximation that the groundwater density in the rock matrix at a particular location in the continuum representing to the fractures is independent of position within the rock matrix. The groundwater density could vary with position in the continuum. However, in order to implement this more general approximation, it would be necessary to store the groundwater density at all locations in the continuum. In order to keep storage requirements as low as possible, the more general approximation was not implemented.

Second, for the current implementation, it is only possible to specify that the initial salinity in the rock matrix varies with position in the continuum representing the fractures. The initial salinity at a location within the continuum representing the fractures is taken to be independent of position within the rock matrix.

The algorithm requires modifications to three distinct parts of CONNECTFLOW:

(i) The subroutines that handle input of the properties of the rock types in a model have to be extended to deal with the properties required to model rock-matrix diffusion;

(ii) The driver subroutine for the transient (“Crank Nicholson”) calculation has to be modified:

- To calculate the quantities *A* and B'^l for the extra term representing the flux into the rock matrix, which is added into the flow equations (see Equation (67));
- To evaluate and keep a record of the various quantities (mainly the *N* intermediate quantities I_n^{l-1}) used to calculate the flux into the rock matrix (see Equation (68));

(iii) The subroutines that describe the flow equations have to be modified to include the extra term representing the flux into the rock matrix.

These modifications are described in the following subsections. During the testing of the algorithm, it was also found to be necessary to make some changes to the solver (see subsection 7.3.3).

5.1 Data input

Under the subcommand >> PHYSICAL PROPERTIES, the table ROCK TYPE PROPERTIES has been modified to include new columns:

MATRIX GEOMETRY

for each rock this keyword can take the values of either 0, for no rock matrix diffusion,

or 1, for rock matrix diffusion in a "slab"-like geometry. Other geometries could be added in the future.

CAPACITY FACTOR

this keyword specifies the capacity factor for each rock (α in Equation (4)).

MATRIX POROSITY

this keyword specifies the porosity of the rock matrix (ϕ_f) for each rock. For salinity, this is generally the same as the capacity factor. (It is not necessary to specify both the matrix porosity and the capacity factor.)

INTRINSIC DIFFUSION COEFFICIENT

this keyword specifies the intrinsic diffusion coefficient for diffusion into the rock matrix for each rock (D_i in Equations (3) and (4)).

EFFECTIVE DIFFUSION COEFFICIENT

this keyword is an alias for the keyword INTRINSIC DIFFUSION COEFFICIENT. It specifies the intrinsic diffusion coefficient for diffusion into the rock matrix for each rock (D_i in Equations (3) and (4)).

FRACTURE SURFACE AREA PER UNIT VOLUME

this keyword specifies the (average) fracture surface area per unit volume for each rock (σ in Equation (3)).

DIFFUSION LENGTH

this keyword specifies the accessible distance into the rock matrix, that is the distance into the matrix at which the flux of salinity in the matrix is zero (d in Equation (7)). It should be set to half the fracture spacing if it is considered that all of the rock matrix between the fractures is potentially accessible, and to the maximum possible penetration depth, if it is considered that there is a limit to the possible penetration depth.

These keywords allow a user to define all the required rock properties.

The only other changes to the input language are under the subcommand >> SALT TRANSPORT. The following keywords have been added:

INCLUDE MATRIX DIFFUSION

this keyword indicates that the modelled equations for groundwater flow and transport of salinity equation should account for diffusion into the rock matrix.

SAVE RMD HISTORY ON UNIT *integer*
this keyword specifies the FORTRAN stream on which the N intermediate quantities I_n^{l-1} should be saved.

SAVE RMD A AND B ON UNIT *integer*
this keyword specifies the FORTRAN stream on which the quantities A and B'^l should be stored for use in the calculation of the discretised flow equations.

ACCURACY PARAMETER FOR MATRIX DIFFUSION
this keyword specifies the acceptable error between non-dimensional forms of the analytical solution for the flux of salinity into the matrix, the short-time solution (Equation (22)) and the long-time solution (Equation (36)). It determines both t^* and the number, N, of intermediate quantities I_n^{l-1} that need to be saved (Equation (53)). The value should be small. A value of 10^{-4} or smaller is recommended.

5.2 Driver routine

The driver subroutine for the transient ("Crank Nicholson") calculation was modified as follows.

If the keyword INCLUDE MATRIX DIFFUSION is set, then:

(i) Before the first time step, a subroutine RMDINI is called. This subroutine determines the value of t^* and the maximum number of intermediate quantities I_n^{l-1} to be kept. The result is used to allocate workspace for storage of the intermediate quantities I_n^{l-1}, as well as the quantities b_n, ω_n, and $c_0' - c^0$.

(ii) For all time steps, the subroutine RMDAB is called. The primary role of this subroutine is to calculate the quantities A (Equation (70)) and B'^l (Equation (71)), and write them out to a file so that they can be accessed by the subroutines that describe the flow equations.
For the first time step, this subroutine also calculates the quantity , saving it for later use. The initial salinity in the rock matrix is specified via a variable "CMAT" [2].

(iii) For all time steps, the subroutine RMDMOD is called. The purpose of this subroutine is to update the intermediate quantities I_n^{l-1} (Equation (68)), and also to calculate the last two terms in B'^l (Equation (71)), which are independent of the current time step size, so that the quantities are available in RMDAB.

(iv) After the first time step, the subroutine RMDHD is called. This subroutine writes to the history file those quantities that are independent of the time step (e.g. t^*, b_n, ω_n, and $c_0' - c^0$).

(a) For those time steps that the user has requested that the calculated global freedoms be saved (using the usual keywords under the subcommand CRANK NICHOLSON), the subroutine RMDSAV is called. This subroutine writes to the history file those quantities that depend on the time step (e.g. I_n^{l-1}).

(b) If the user requests a "restart", the subroutine RMDSAV reads back the history file for the specified time step and sets up the rock-matrix diffusion algorithm appropriately.

[2] This is how the initial salinity is specified in the latest version of the rock-matrix diffusion algorithm (in Release 8.0). In the prototype version, which was used for the study reported here, the initial salinity in the rock matrix was specified using a function FUNCTION CM0 (CF0, X, MSD, ILYR) where the function arguments are the initial concentration in the fractures, position, number of dimensions, and rock type. This allowed the user to specify the initial salinity as a function of position (in the continuum, but not within the rock matrix) by supplying a suitable version of this function.

5.3 Equation for the transport of salinity

The subroutines that describe the flow equations were modified to include the extra term for the flux into the matrix. In particular:

(i) Subroutine FR3ST, which describes the steady-state part of the equations for transport of salinity, was modified to include B'^{1}. (It should be noted that it was not necessary to modify the subroutine JR3ST that calculates the contribution to the Jacobian from the terms defined in subroutine FR3ST.)

(ii) Subroutine FTR3ST, which describes the transient part of the equations for transport of salinity, was modified to include A. Subroutine JTR3ST, which calculates the Jacobian of the terms defined in subroutine FTR3ST, was modified to include the (trivial) derivative of the extra term in FTR3ST.

It should be noted that only the PRES CONC FORMULATION for coupled groundwater flow and transport of salinity in 3D was modified. (The rock-matrix diffusion option is, therefore, currently, only available for 3D calculations using this formulation.)

6 Test case

The implementation of the algorithm for rock matrix diffusion was verified using a simple test case for which a semi-analytical solution is available.

6.1 Definition of the test case

The test case considers transport of salinity in a horizontal column 10^4 m long. The column is divided into 100 cubic finite elements, each with sides of length 10^2 m. The physical properties are as follows:

(i) the permeability of the column is 10^{-11} m^2;

(ii) the effective fracture (or kinematic) porosity of the column is 10^{-2};

(iii) freshwater and saline water are taken to have the same density: 998.3 kg m^{-3};

(iv) the viscosity of the groundwater is 10^{-3} Pa s;

(v) the salt diffusion coefficient is 10^{-9} m^{2}s^{-1};

(vi) the tortuosity of the rock is 10^0;

(vii) the longitudinal dispersion length is 10^2 m;

(viii) the transverse dispersion length is 10^1 m;

(ix) the acceleration due to gravity is 9.81 ms^{-2}.

The rock matrix diffusion properties are as follows:

(i) the capacity factor or accessible matrix porosity α is 3 10^{-1};

(ii) the intrinsic diffusion coefficient D_i is 5 10^{-11} m^{2}s^{-1};

(iii) the fracture surface area per unit volume σ is 2 m^{-1};

(iv) the diffusion length is given by half the spacing between the fractures, that is $\frac{1-\phi_f}{\sigma}$.

The initial condition is that the salinity is zero in the fractures and in the rock matrix.

The boundary conditions are that at one end of the column, the hydrostatic head is 50 m and the mass fraction of dissolved salinity is 1. At the other end of the column, the hydrostatic head is 0 m and a zero-dispersive-flux boundary condition is specified for the salinity.

6.2 Semi-analytical solution for the test case

The equations for the transport of salinity (Equations (1)–(4)) for the test case can be solved by the method of Laplace transforms (see also subsection 2.1).

The Laplace transform of Equation (3) for the case of constant fracture porosity and constant density in one dimension is

$$\phi_f \rho (p\hat{c} - c^0) + \phi_f \rho v \frac{\partial \hat{c}}{\partial x} = \phi_f \rho \left(\frac{D_m}{\tau} + a_L v \right) \frac{\partial^2 \hat{c}}{\partial x^2} + \sigma \rho D_i \left. \frac{\partial \hat{c}'}{\partial w} \right|_{w=0} \quad , \tag{72}$$

where

v is the porewater velocity, which is given by $\mathbf{v} = \dfrac{\mathbf{q}}{\phi_f}$, and is a constant for this test case;

D_m is the diffusion coefficient of salinity;

τ is the tortuosity;

a_L is the longitudinal dispersion coefficient of salinity.

From the analysis in subsection 2.1, in the case when the salinity in the rock matrix is initially zero

$$-D_i \left. \frac{\partial \hat{c}'}{\partial w} \right|_{w=0} = D_i m \tanh[md] = \sqrt{\alpha D_i p}\, \tanh[md] (= p\hat{g}) \ . \tag{73}$$

Substituting this result into Equation (72) gives

$$\left(p + \frac{\sigma D_i}{\phi_f} \sqrt{\frac{\alpha p}{D_i}} \tanh\left[\sqrt{\frac{\alpha p}{D_i}} d \right] \right) \hat{c} + v \frac{\partial \hat{c}}{\partial x} - \left(\frac{D_m}{\tau} + a_l v \right) \frac{\partial^2 \hat{c}}{\partial x^2} = 0 \ . \tag{74}$$

The solution to this equation is of the form

$$\hat{c} = C_+ \exp(\beta_+ x) + C_- \exp(\beta_- x) \ . \tag{75}$$

where β_+, β_- are the roots of

$$\left(p + \frac{\sigma D_i}{\phi_f} \sqrt{\frac{\alpha p}{D_i}} \tanh\left[\sqrt{\frac{\alpha p}{D_i}} d \right] \right) + v\beta - \left(\frac{D_m}{\tau} + a_l v \right) \beta^2 = 0 \ . \tag{76}$$

Hence

$$\beta_\pm(p) = \frac{v}{2\left(\dfrac{D_m}{\tau} + a_L v \right)} \left(1 \pm \sqrt{1 + \frac{4\left(\dfrac{D_m}{\tau} + a_l v \right) \left(p + \dfrac{\sigma D_i}{\phi_f} \sqrt{\dfrac{\alpha p}{D_i}} \tanh\left[\sqrt{\dfrac{\alpha p}{D_i}} d \right] \right)}{v^2}} \right) . \tag{77}$$

$C_\pm$ are determined from the boundary conditions as follows. At the inlet boundary, the mass fraction of salinity in the fractures is 1 so

$$\hat{c}(x = 0, p) = \frac{1}{p} = C_+ + C_- \ ; \tag{78}$$

At the outlet boundary, there is a zero-dispersive-flux boundary condition for the salinity in the fractures so

$$\frac{\partial \hat{c}}{\partial x}(x = L, p) = 0 = \beta_+ C_+ \exp(\beta_+ L) + \beta_- C_- \exp(\beta_- L) \tag{79}$$

Equations (78) and (79) can be solved to determine $C_\pm$. The results can then be substituted into Equation (75). The end result is an expression for the Laplace transform of the salinity in the fractures. This expression can be inverted numerically using, for example, Talbot's algorithm /8/ to determine $c(x,t)$.

6.3 Comparison of CONNECTFLOW results and the semi-analytical solution

The results of calculations of coupled groundwater flow and transport of salinity with rock-matrix diffusion carried out using CONNECTFLOW were compared with the salinity determined as in subsection 6.2. These were in very good agreement. For example, in Figure 6-1 a comparison of the evolution of the salinity at the mid-point of the column is shown, and in Figure 6-2, comparisons of the salinities along the column at various times are shown.

As commented, the results obtained using CONNECTFLOW are in very good agreement with the results obtained from the semi-analytical solution. This builds confidence that the rock-matrix diffusion algorithm has been coded correctly.

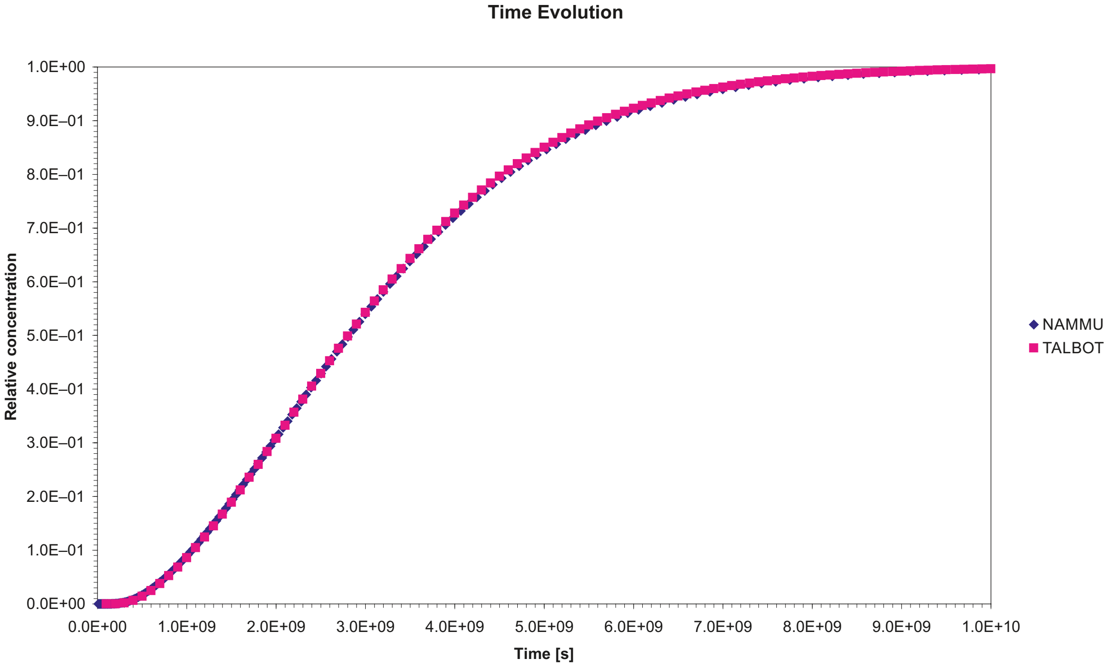

Figure 6-1. *Comparison of the evolution of salinity (in the fractures) at the mid-point of the column calculated using a semi-analytical method and using CONNECTFLOW.*

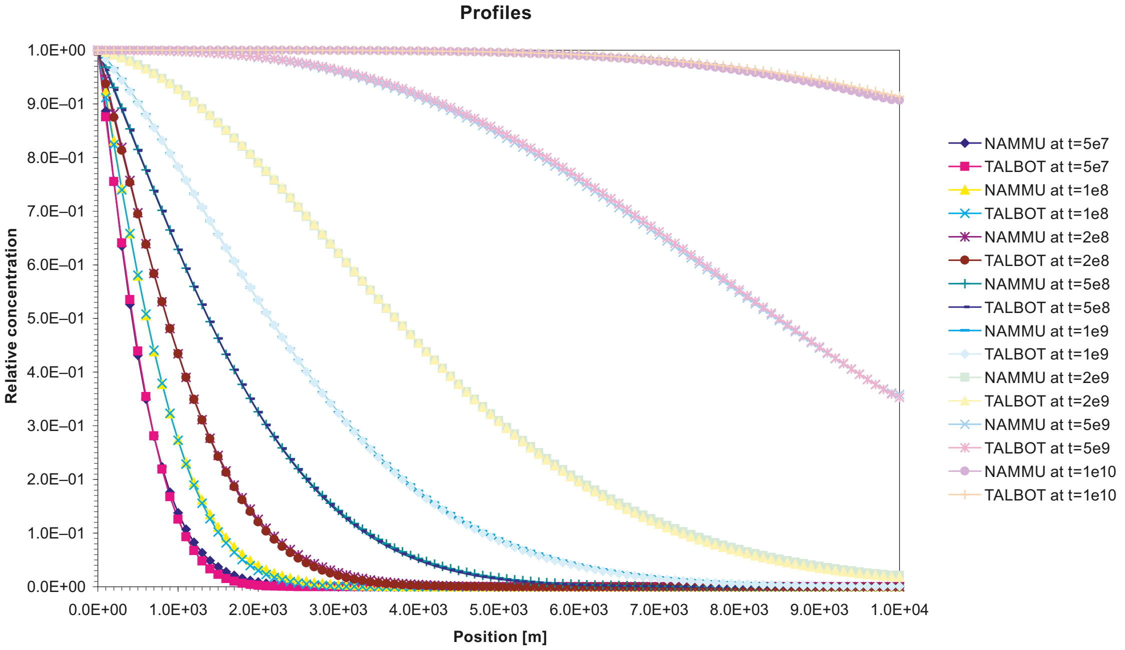

Figure 6-2. *Comparison at various times of the profiles of the salinity (in the fractures) along the column calculated using a semi-analytical method and using CONNECTFLOW.*

7 A realistic example

7.1 Introduction

Once the implementation in CONNECTFLOW of the algorithm for rock-matrix diffusion in transport of salinity had been verified (see Section 6), it was applied to a large site-scale model to examine the use of the approach for the sort of models that might be developed in practical modelling. The model was based on a model that was developed as part of the site descriptive modelling for Forsmark version 1.1 /12/.

The topography and the extent of the region modelled are shown in Figure 7-1.

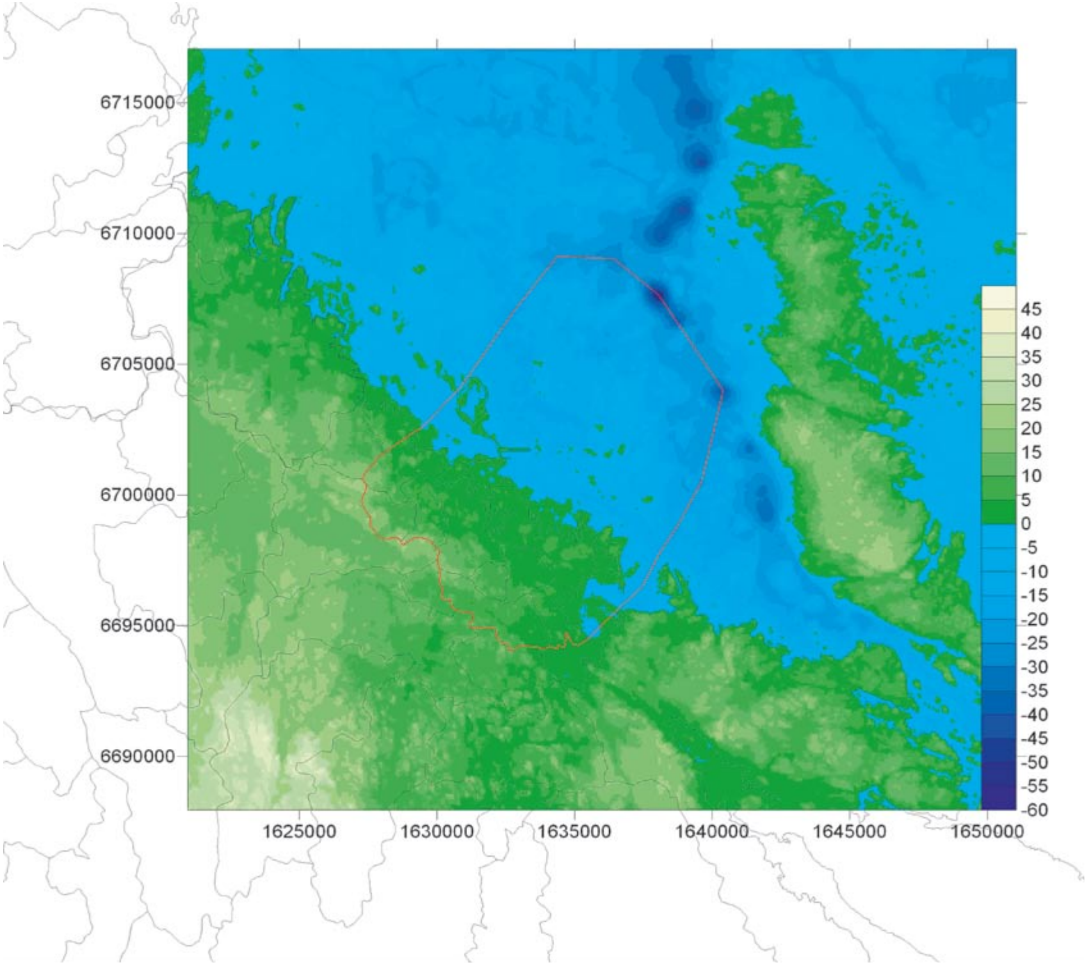

Figure 7-1. *Topography and elevation of the bottom of the sea, also showing the lateral boundary of the model (assumed to coincide with water divides).*

The model comprises 5 layers (L1, L2, L3, L4 and L5, from the top of the model downwards). Layer L1 is 3 m thick and layer L2 is 10 m thick (so the base of L2 is 13 m below the land surface in the model). The base of layer L3 is 400 m below current sea level, the base of layer L4 is 2,100 m below current sea level and the base of layer L5 is 2,300 m below current sea level.

The model includes various fracture zones. The positions of the fracture zones are shown in Figure 7-2.

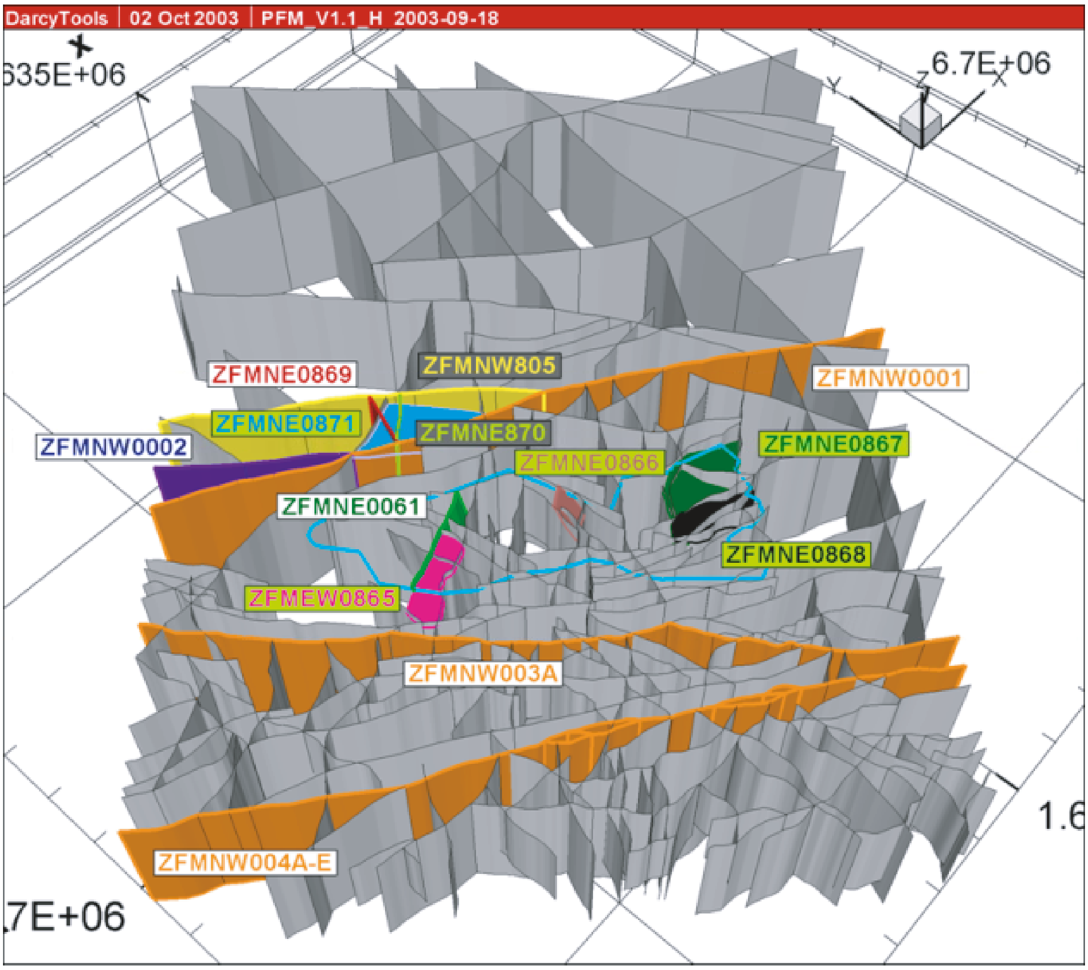

Figure 7-2. *The positions of the fracture zones (reproduced from /14/).*

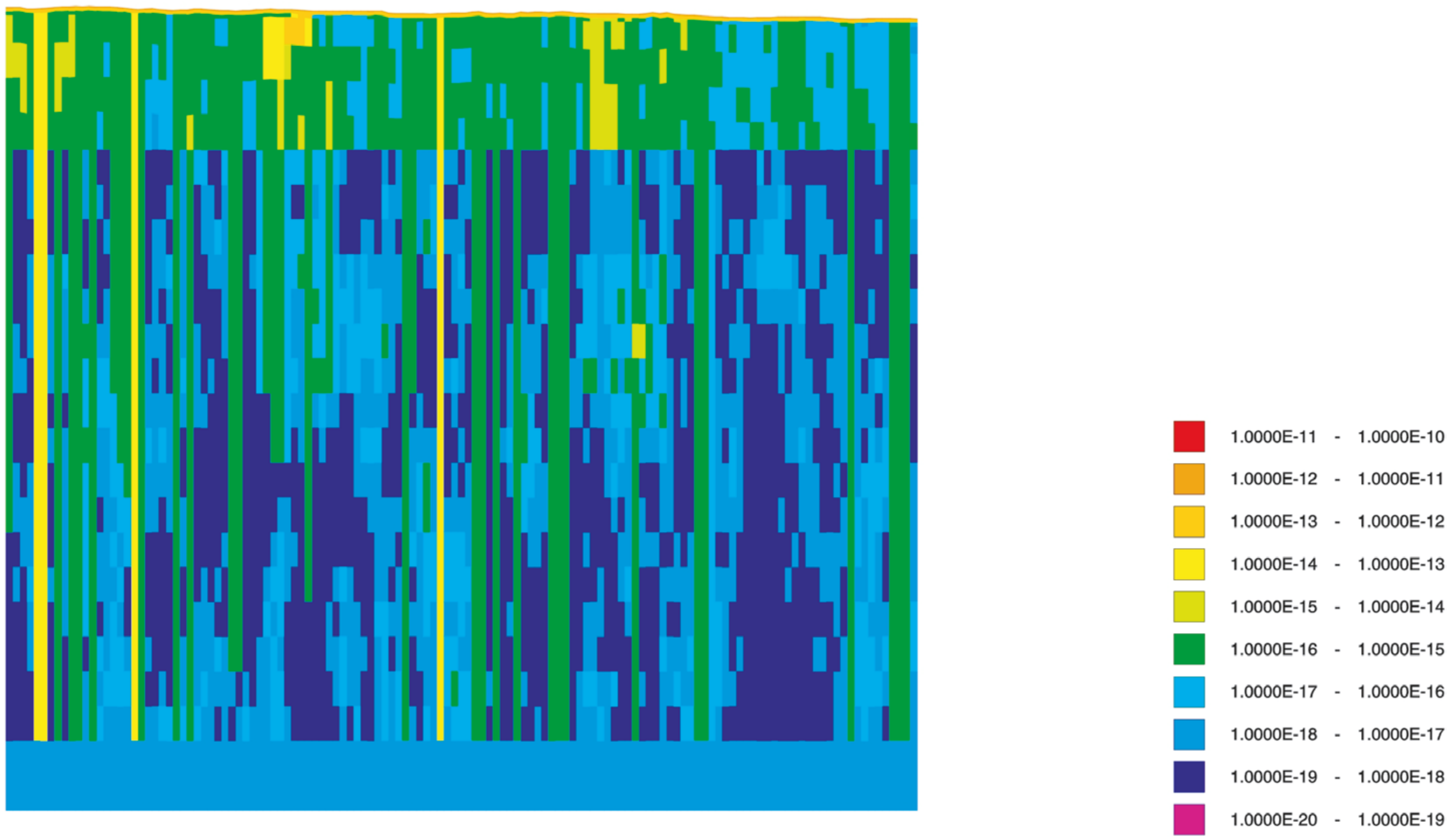

***Figure 7-3.** Permeability on a grid slice that runs from southwest to northeast through the borehole KFM02A. Note that the vertical scale has been exaggerated by a factor of five.*

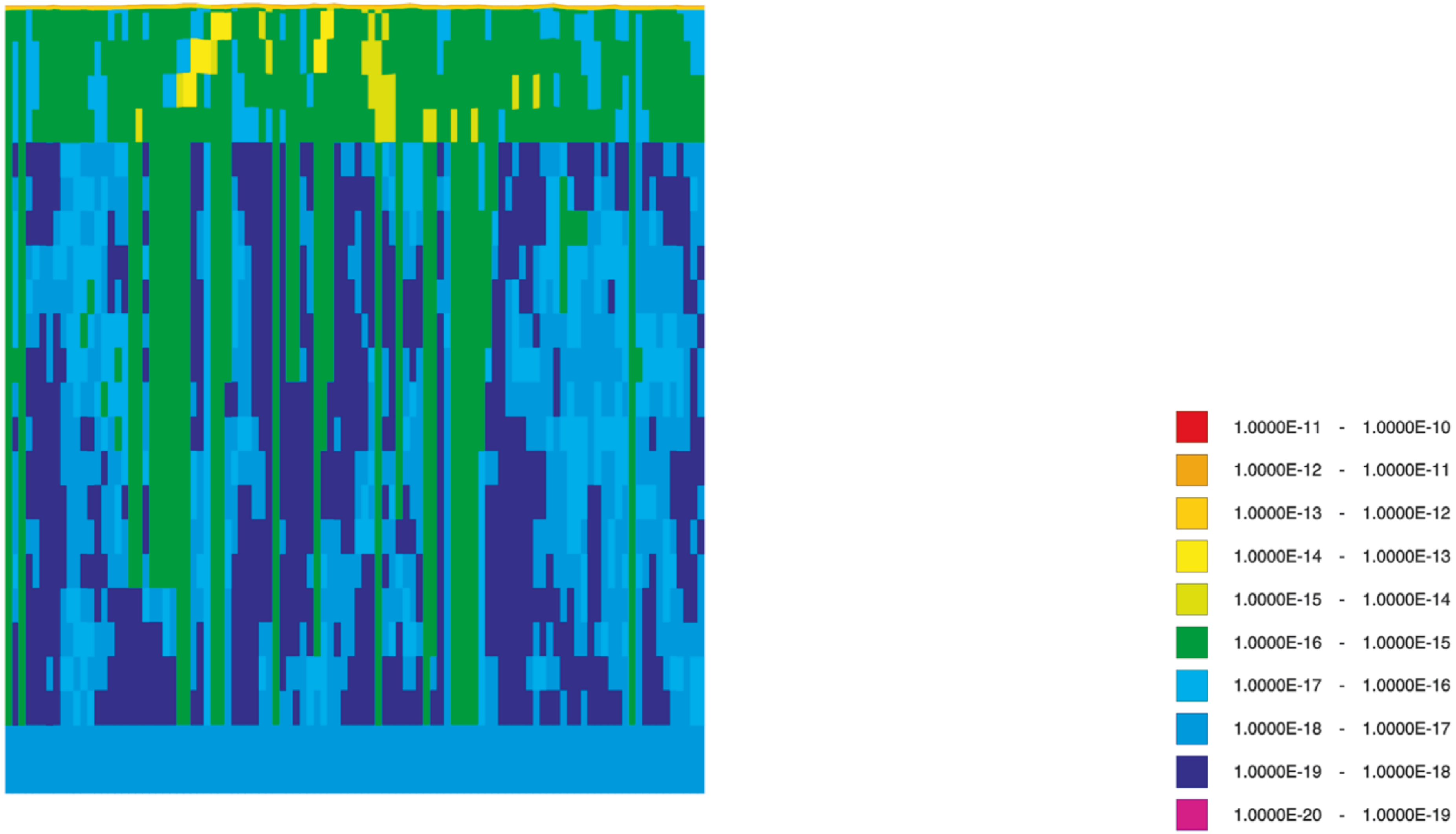

Figure 7-4. *Permeability on a grid slice that runs from southeast to northwest through the borehole KFM02A. Note that the vertical scale has been exaggerated by a factor of five.*

Initial simulations were based on the data set for Case 8 of /12/. However, this Case is harder to solve than the other Cases. This is mainly because the Case has a smaller porosity (10^{-3} instead of $5\ 10^{-3}$). Hence the groundwater velocities are larger, by a factor of 5. Therefore, assuming that advection is the dominant flow process, it would be necessary for this case to use time steps that are five times smaller than the time steps for the other Cases in order for the changes in salinity in a time step to have similar magnitude to the changes in the other Cases. This is because the changes are proportional to the velocity. The calculation would therefore take about five times longer. If it was attempted to use time steps for this Case similar to the time steps for the other Cases, the changes in salinity at each time step would be larger, and the non-linearities would be greater, making the non-linear equations harder to solve.

Therefore, the final simulations presented below were based on the data set for Case 5 of /12/. A Base Case without rock-matrix diffusion corresponding to Case 5 and nine variants with rock-matrix diffusion were considered. The hydrogeological properties of the layers in Case 5 are summarised in Table 7-1. These properties are based on Version 1.1 of the hydrogeological site description of Forsmark. The distribution of the permeability in the model is illustrated by Figures 7-3 and 7-4, which show the permeability on two vertical slices through borehole KFM02A (see Figure 7-5).

Table 7-1. Hydrogeological properties.

Layer	Permeability [m^2]			Porosity [–]	Specific storage coefficient [m^{-1}]	Longitudinal dispersion length [m]	Transverse dispersion length [m]
	k_{xx}	k_{yy}	k_{zz}				
L1	$1.53\ 10^{-12}$			$5\ 10^{-2}$	0	40	5
L2	$2.039\ 10^{-14}$	$2.039\ 10^{-14}$	$2.039\ 10^{-13}$	$5\ 10^{-3}$	0	40	5
L3	Log normal with median $2.237\ 10^{-16}$ and variance of the logarithm 0.4531[1)]			$5\ 10^{-3}$	0	40	5
L4	Log normal with median $1\ 10^{-18}$ and variance of the logarithm 1			$5\ 10^{-3}$	0	40	5
L5	$1\ 10^{-18}$			$5\ 10^{-3}$	0	40	5
FZs	HCD1[2)]			Rock[3)]	0	40	5

[1] A minimum permeability of 10^{-19} m^2 was specified in order to minimise the contrasts in the stochastic permeability field.

[2] The properties are as defined in the Forsmark version 1.1 xml files HCD1.xml.

[3] The porosity of the fracture zone is not used; instead the porosity of the background rock is used, as in the original modelling of Holmen et al. /13/.

Using the model, calculations of the evolution of the groundwater flow system from conditions 10,000 years ago to the present were carried out. During this period, the elevation of the land surface relative to sea level gradually rises because of glacial rebound from conditions when the land was covered by an ice-sheet. The salinity of the Baltic sea also varies. The variation of the shoreline elevation and the sea water salinity are shown in Figures 7-6 and 7-7. These variations lead to changes in the groundwater salinity and the groundwater flow in the region modelled.

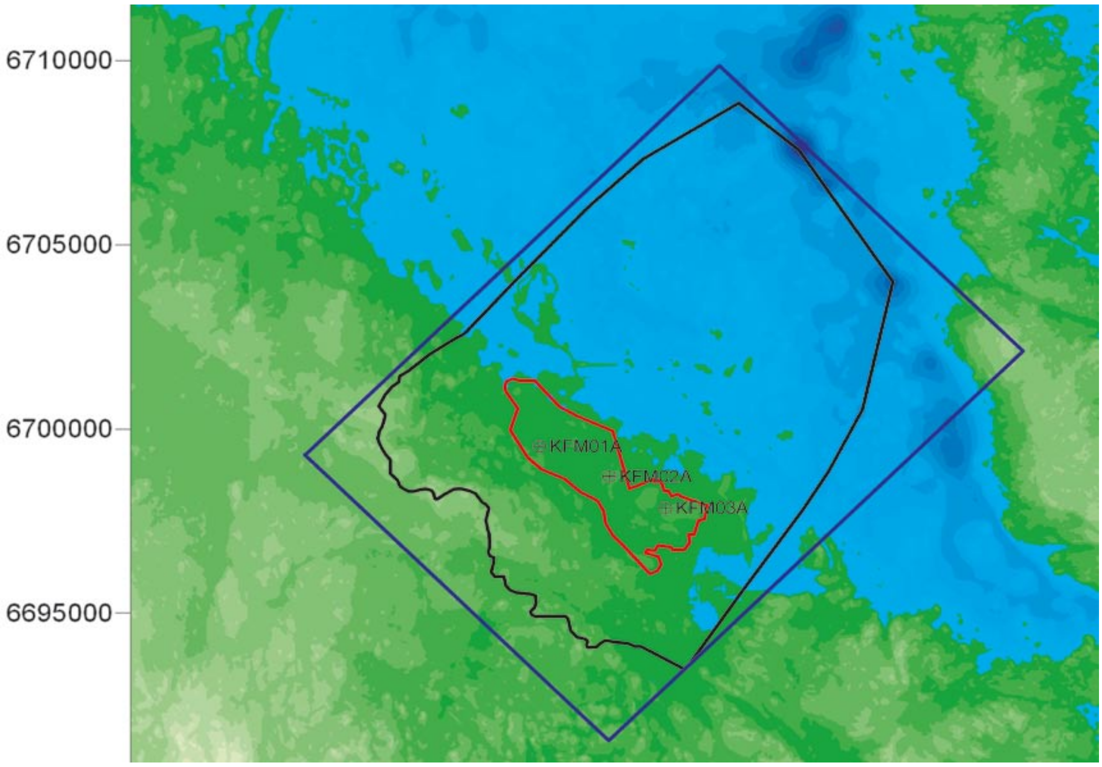

***Figure 7-5.** Location of boreholes. The blue line denotes the regional investigation area, the black line denotes the boundary of the model and the red line denotes the inner investigation area.*

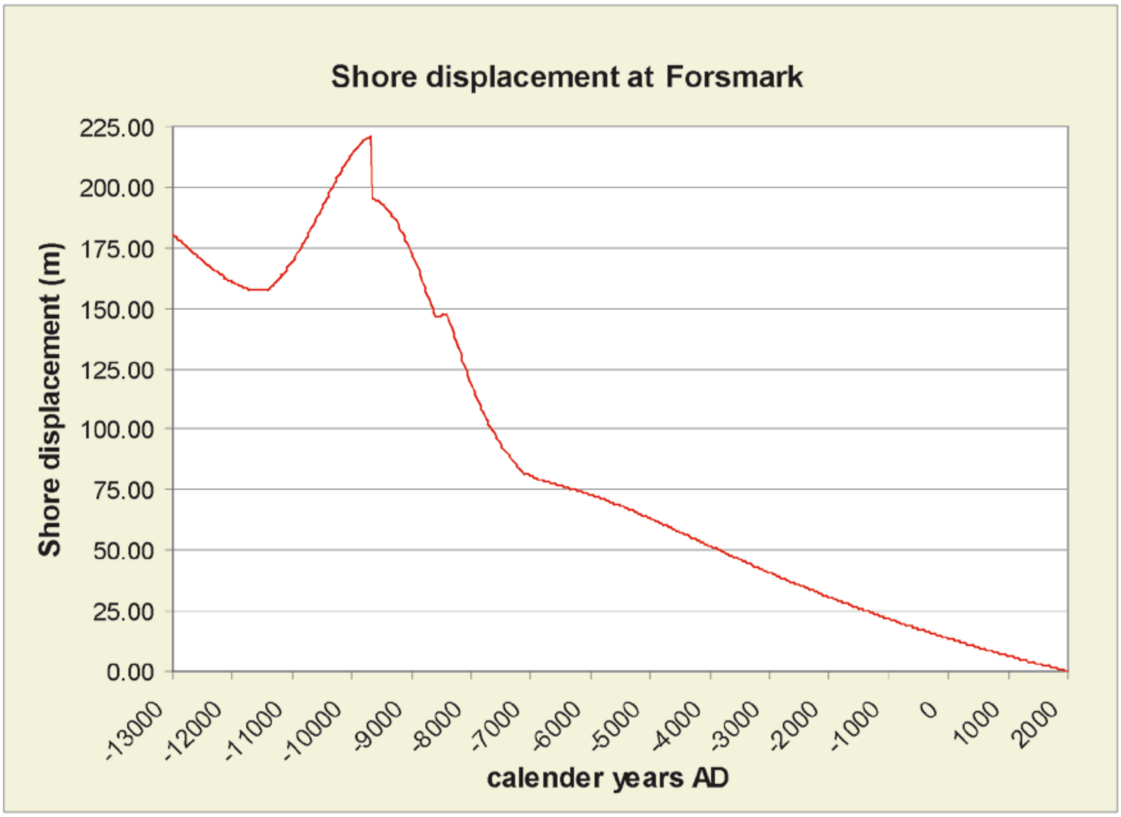

***Figure 7-6.** Historic displacement of the shore at Forsmark.*

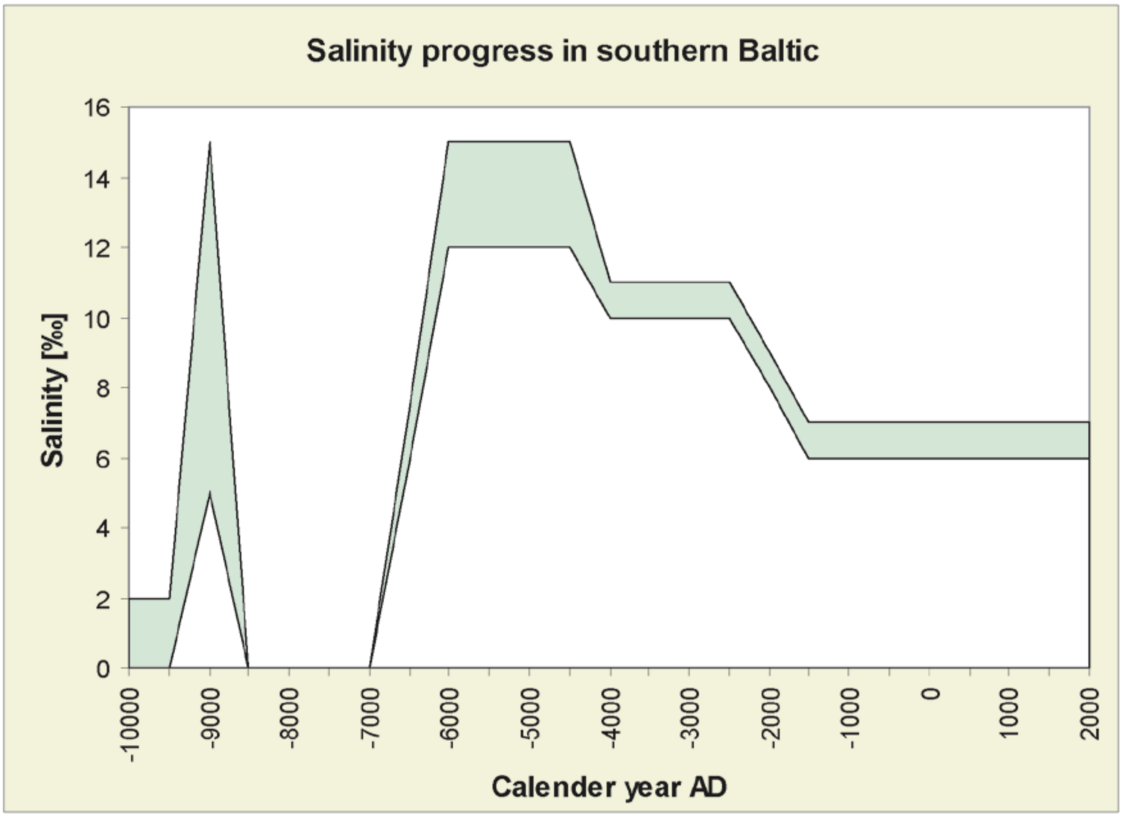

***Figure** 7-7. Evolution of salinity in the southern Baltic Sea.*

The initial salinity distribution (corresponding to a time 10,000 years ago) was taken to be that the groundwater is fresh down to a depth of –500 m; the salinity increases linearly from 0% to 10% between –500 m and –2,100 m; and the salinity is 10% below –2,100 m. The initial groundwater pressure in the model was taken to be the hydrostatic pressure corresponding to the initial salinity distribution.

The sides of the model are located at the assumed positions of regional water divides, and so there is no flow of groundwater through the sides. There is also assumed to be no flux of salinity through the sides. The bottom of the model is a no flow boundary and is taken to have constant salinity. On the top of the model, the boundary condition is effectively that the hydrostatic head is specified as a function of the surface elevation. The boundary condition for the salinity on the top surface is that the salinity is specified at locations where the direction of groundwater flow is into the model and that, at locations where the flow direction is out of the model, there is an outflow, or zero-dispersive flux, condition (which effectively means that the flux of salinity out of the model is given by the product of the calculated groundwater flux and the calculated salinity).

7.2 Rock-matrix diffusion properties

Layers L1 and L2 are taken to be porous media, without rock-matrix diffusion. Rock matrix diffusion is assumed to occur only in layers L3, L4 and L5, which are taken to represent fractured rocks. For these layers:

(i) the capacity factor α is taken to be the matrix porosity used in the original calculation (Case 5 of /13/), that is 5 10^{-3};

(ii) the intrinsic diffusion coefficient D_i is based on the reasonable value for a non-sorbing tracer at the Aberg site in the SR 97 assessment /15/, that is 8.24 10^{-14} m^2s^{-1};

(iii) the diffusion length is given by half the spacing between the fractures, that is $\frac{1-\phi_f}{\sigma}$,, which effectively assumes that all of the rock matrix is accessible by diffusion.

Values of two further parameters for each layer are required: the effective fracture (or kinematic) porosity, ϕ_f, due to the fractures carrying the flow and the fracture surface area per unit volume, σ. These parameters depend on properties of the networks of fractures in the layers. If explicit models for these networks were available, then ϕ_f and σ could be calculated from the fracture-network models. However, Case 5 was not based directly on a fracture-network model, but rather the effective permeability and porosity of the various layers were directly specified. Therefore, ϕ_f and σ cannot be calculated directly. Nevertheless, it is desirable that the values of ϕ_f and σ should be consistent with a fracture network which is consistent with the assumed permeability. This can be achieved as follows. In many cases, the effective permeability, *k*, and fracture porosity for a network can be related (to a good approximation) to the (geometric) mean fracture aperture and the fracture spacing *b* by the following expressions

$$k = \frac{a^3}{12b} \tag{80}$$

$$\phi_f = C_F \frac{a}{b} \ , \tag{81}$$

where C_F is a so-called roughness factor, which here has been assumed to have a value 4.74.

Then, given a value of *k* and an assumed value of ϕ_f, the mean fracture aperture and the fracture spacing can be obtained from

$$a = \sqrt{C_F \frac{12k}{\phi_f}} \ , \tag{82}$$

$$b = C_F \frac{a}{\phi_f} \ . \tag{83}$$

To be realistic, *b* should have a value between about 1 m and 100 m, which constrains the allowed values of ϕ_f. σ can be obtained from *b* using

$$\sigma = \frac{2}{b} \ . \tag{84}$$

Thus it is possible to derive values for ϕ_f and σ that are consistent with a fracture network which is consistent with the assumed permeability. Three possible sets of values were determined in this way. Nine variant cases were defined using these sets and three possible values for the intrinsic diffusion coefficient: the value discussed above, and larger values, which should allow the matrix to be accessed more rapidly. The parameters for the variants are listed in Table 7-2.

Table 7-2. Rock matrix diffusion parameters used in the CONNECTFLOW simulations of the Forsmark site.

Variant	Rock	ϕ_f [–]	α [–]	D_i [m²s⁻¹]	σ [m⁻¹]	Characteristic diffusion time [y]
11	L3	$10^{-4.5}$	5 10^{-3}	8.24 10^{-14}	6.64 10^{-1}	1.09 10^{3}
	L4 / L5	$10^{-5.5}$	5 10^{-3}	8.24 10^{-14}	3.14 10^{-1}	4.87 10^{3}

Variant	Rock	ϕ_f [–]	α [–]	D_i [m²s⁻¹]	σ [m⁻¹]	Characteristic diffusion time [y]
12	L3	$10^{-4.0}$	$5\ 10^{-3}$	$8.24\ 10^{-14}$	$3.74\ 10^{0}$	$3.44\ 10^{1}$
	L4 / L5	$10^{-5.5}$	$5\ 10^{-3}$	$8.24\ 10^{-14}$	$3.14\ 10^{-1}$	$4.87\ 10^{3}$
13	L3	$10^{-4.0}$	$5\ 10^{-3}$	$8.24\ 10^{-14}$	$3.74\ 10^{0}$	$3.44\ 10^{1}$
	L4 / L5	$10^{-5.0}$	$5\ 10^{-3}$	$8.24\ 10^{-14}$	$1.77\ 10^{0}$	$1.54\ 10^{2}$
21	L3	$10^{-4.5}$	$5\ 10^{-3}$	$8.24\ 10^{-13}$	$6.64\ 10^{-1}$	$1.09\ 10^{2}$
	L4 / L5	$10^{-5.5}$	$5\ 10^{-3}$	$8.24\ 10^{-13}$	$3.14\ 10^{-1}$	$4.87\ 10^{2}$
22	L3	$10^{-4.0}$	$5\ 10^{-3}$	$8.24\ 10^{-13}$	$3.74\ 10^{0}$	$3.44\ 10^{0}$
	L4 / L5	$10^{-5.5}$	$5\ 10^{-3}$	$8.24\ 10^{-13}$	$3.14\ 10^{-1}$	$4.87\ 10^{2}$
23	L3	$10^{-4.0}$	$5\ 10^{-3}$	$8.24\ 10^{-13}$	$3.74\ 10^{0}$	$3.44\ 10^{0}$
	L4 / L5	$10^{-5.0}$	$5\ 10^{-3}$	$8.24\ 10^{-13}$	$1.77\ 10^{0}$	$1.54\ 10^{1}$
31	L3	$10^{-4.5}$	$5\ 10^{-3}$	$8.24\ 10^{-12}$	$6.64\ 10^{-1}$	$1.09\ 10^{1}$
	L4 / L5	$10^{-5.5}$	$5\ 10^{-3}$	$8.24\ 10^{-12}$	$3.14\ 10^{-1}$	$4.87\ 10^{1}$
32	L3	$10^{-4.0}$	$5\ 10^{-3}$	$8.24\ 10^{-12}$	$3.74\ 10^{0}$	$3.44\ 10^{-1}$
	L4 / L5	$10^{-5.5}$	$5\ 10^{-3}$	$8.24\ 10^{-12}$	$3.14\ 10^{-1}$	$4.87\ 10^{1}$
33	L3	$10^{-4.0}$	$5\ 10^{-3}$	$8.24\ 10^{-12}$	$3.74\ 10^{0}$	$3.44\ 10^{-1}$
	L4 / L5	$10^{-5.0}$	$5\ 10^{-3}$	$8.24\ 10^{-12}$	$1.77\ 10^{0}$	$1.54\ 10^{0}$

In Table 7-2, the characteristic diffusion time for each layer is also listed. The characteristic diffusion time is an estimate of the time for salinity to diffuse into the rock matrix:

$$\left(\frac{b}{2}\right)^2 \Big/ 4\left(\frac{D_i}{\alpha}\right) = \frac{\alpha}{4D_i\sigma^2} \; . \tag{85}$$

It is clear from Table 7-2 that for the calculations considered, which consider a total time of 10,000 years, the salinity will be able to access most of the matrix.

In view of the remarks above about the difficulties for Case 8, which are due to the low porosity for the Case, it is perhaps worth noting that, although the fracture (or kinematic) porosities for the variants above are lower than the porosity for Case 8, the porosity effectively accessed by the salinity will be larger than the fracture porosity, because of rock-matrix diffusion.

7.3 Numerical issues

The rock-matrix diffusion algorithm was found to perform as expected, adding little to the computational cost of the calculation for a single time step (see subsection 4.2). However, it was found initially, that to avoid instability in the solution of the numerical equations for the transport of salinity, smaller time steps were required (see subsection 4.2). This is a consequence of reducing the porosity in the model from that of the rock matrix to that due to the fractures carrying the flow, which increases the porewater velocity. The smaller time steps would have led to overall computational times that were excessive. In order to deal with this, some enhancements were made to the algorithm for solving the equations for the transport of salinity. These enhancements addressed three topics:

(i) mixed interpolation;

(ii) nodal quadrature;

(iii) a sequential-iteration time-stepping algorithm.

These enhancements are discussed below.

7.3.1 Mixed interpolation

In regions where the flow is small, Darcy's Law (Equation (1)) is a balance between the vertical component of the pressure gradient and the gravitational term (proportional to the density), both of which may be much larger than the flow. In a finite-element discretisation, if the same order of interpolation is used for the pressure and the density, then it is not possible to achieve a pointwise balance between the vertical component of the pressure gradient and the gravitational term. This is because the two terms behave differently, the gravitational term being given by a higher-order piecewise-polynomial than the pressure gradient. An average balance integrated over an element is achieved. However, there are large ripples on the scale of the elements in the velocity obtained from the discretised version of Equation (1) (see, for example, /7/). Various techniques have been proposed to deal with this issue, such as the use of a so-called "consistent velocity" approximation, or the use of an interpolation for the salinity that has lower-order than the interpolation used for the pressure (see /7/). In the calculations reported below, a variant of the latter approach was used in which a lower-order interpolation was used directly for the density (rather than the salinity). In fact, constant interpolation on each element was used for the density, and tri-linear interpolations were used for the pressure and salinity.

7.3.2 Nodal quadrature

The standard Galerkin finite-element method generally gives excellent results provided the problem is well behaved and the grid is fine enough to resolve the features present in the solution. However, in transient calculations, near rapid changes in the solution, numerical solutions often exhibit spatial oscillations on the scale of the elements. One cause of such ripples is the finite-element discretisation of the time-derivative terms in the flow equations. This leads to so-called "mass-matrix ripples". In many cases, these ripples are not significant, and indeed some authors have argued that they are associated with higher accuracy in the results. However, the ripples are, in some sense, unphysical, and in non-linear calculations may lead to problems. For example, the ripples may lead to negative salinities, or to salinities greater than the maximum entering the domain. Such salinities will affect the density, and hence the flow. In particular they may make the numerical calculations much more non-linear, increasing the difficulty of solving the numerical system.

The mass-matrix ripples can be (largely) avoided by the use of the technique known as mass-lumping. This can be implemented in various ways. The simplest is to use nodal quadrature, in which the sample points in the numerical integration scheme used to carry out the integrals over an element required in the finite-element method are taken at the nodes of the element. This leads to a scheme in which the numerically calculated salinities are generally physical (greater than zero and less than the maximum entering the domain). This is likely to lead to a more robust algorithm.

However, it should be noted that other aspects of the discretisation, such as the treatment of the advective term, can also lead to ripples near rapid changes in the solution.

7.3.3 Sequential-iteration algorithm

A sequential-iteration technique was used in the integration of the numerical equations over time. At each time step:

(i) the average density in each element is calculated from a suitable equation;

(ii) the residual pressure at the end of the time step is calculated (using the calculated average density) from a discretised version of the steady-state flow equations (Equations (1) and (2));

(iii) the salinity at the end of the time step is calculated from a discretised version of the salinity transport equation (Equation (3)) including the rock-matrix diffusion contributions. In this calculation, the previously calculated average density and calculated flow are used.

There are different variants of this scheme, in which a single cycle of the above calculations is carried out for each time step, or a fixed number of cycles is carried out, or the cycles are repeated until convergence is obtained for the non-linear equations at each time step. The last algorithm was implemented and verified using a variant of the Henry test case /16/. It was found that, for the realistic example, with this algorithm, it was necessary for the time step to be about a third of a year. With such a time step size, the calculations would have taken prohibitively long. Therefore, the time-stepping scheme in which only a single cycle of calculations is carried out for each time step was adopted for the calculations because it was found possible to use much larger time steps, similar to those used in previous calculations for the model without rock-matrix diffusion /13/.

This time-stepping algorithm might also be useful in future calculations without rock–matrix diffusion because it might allow larger time steps to be taken than would be possible with the algorithm implemented prior to the work described in this report.

7.4 Results

As discussed, the evolution of the system from conditions 10,000 years ago to the present day was calculated. The calculated present-day salinities are illustrated by Figures 8-1 and 8-2. These Figures show the salinities for the Base Case model (without rock-matrix diffusion), corresponding to Case 5 of Holmén et al. /12/, on two vertical slices through borehole KFM02A (the same slices as in Figures 7-3 and 7-4). In these plots, a non-linear colour scale for the salinity is used. The scale covers a range of salinity from 0% to 10% by weight. Note that the intervals of salinity are:

- Below 0.005%. 0.005% corresponds to drinking water standards (i.e. Cl content less than 30 milligram/litre).
- Between 0.005% and 0.05%. 0.05% corresponds to the taste limit (i.e. Cl content less than 300 milligram/litre).
- Between 0.05% and 0.6%. 0.6% corresponds to the salinity of the present Baltic Sea.
- Between 0.6% and 1.2%. 1.2% corresponds to the maximum salinity of the Litorina Sea.
- Between 1.2% and 2%.
- Between 2% and 3%.
- Between 3% and 4%.
- Between 4% and 6%.

- Between 6% and 8%.
- Between 8% and 10%. 10% corresponds to the maximum salinity of the model (i.e. the salinity along the base of the model).

As can be seen, the stratified distribution of salinity at depths below about 500 m has not changed greatly during the period of the calculation. The main changes are that, in some locations, plumes of saline water have flowed up from the saline region at depth into the region above 500 m depth that was taken to be fresh at the start of the calculation, and in other locations, plumes of saline water have flowed down into the fresh region from the seabed. The plumes generally appear to be associated with the conductive features.

The results are also illustrated in Figures 8-3 to 8-5, which show vertical profiles of the calculated salinity for boreholes KFM01A, KFM02A and KFM03A. (The positions of these boreholes are tabulated in Table 7-3 and shown in Figure 7-5.)

Table 7-3. Data on the boreholes.

Borehole	X	Y	Z	Bearing	Inclination	Length
KFM01A	1 631 397	6 699 530	3	318	–85	1,001
KFM02A	1 633 183	6 698 713	7	276	–85	1,002
KFM03A	1 634 631	6 697 852	8	272	–86	1,001

Overall, the results for the Base Case model are similar to those presented for Case 5 by Holmén et al. This is not surprising because the model is the same. Some differences are to be expected because the numerical algorithm used differs from that used by Holmén et al. In particular, there are indications that the undershoots (negative salinities) are smaller in the results presented here for the Base Case model than in the results presented by Holmén et al. This would be consistent with the use of nodal quadrature for the calculations presented here, which removes one source of ripples, namely the mass-matrix ripples.

However, there are still some undershoots in the calculated solution. This will be due to some aspect of the numerical discretisation. It could be the result of using too large a time step, for example. It might also be the result of only carrying out a single cycle of the sequential-iteration time-stepping scheme, rather than iterating to convergence, for example. This could have the following effect. In the first iteration for a time step, the salinity might move in a certain flow field, but the effect of taking into account the fact that the salinity has moved might be to change the flow field. For example, it might lead to a reduced flow, and were this taken into account the salinity might not move as far, or it might lead to an increased flow, which might move the salinity further.

Figures 8-3 to 8-5 also show the calculated salinities in the continuum representing the fractures for the variant calculations with rock-matrix diffusion. The results appear to behave as expected. For Variant-33, the characteristic diffusion times are the shortest and much shorter than the total period simulated (see Table 7-1). Therefore, salinity will have had time to access most of the accessible rock matrix and so the results for this case should be similar to those for the Base Case. In contrast, the characteristic diffusion times for Variant-11 are much longer and comparable to the total period simulated. Therefore, salinity would not have had time to access all of the accessible rock matrix for this variant. Hence, the results for this variant would be expected to differ somewhat from the results for the Base Case. As can be seen, expectation is borne out by the results.

8 Comments

The results presented above demonstrate that the algorithm for rock-matrix-diffusion in transport of salinity can be used in some models of the sort that are likely to be used in practical modelling. The algorithm allows models to be set up in which the effective porosity for the continuum representing a network of fractures is taken to be a realistic porosity corresponding to the fractures. At the same time, the models can represent the possibility that there is a significantly greater porosity available within the rock matrix which may be accessed by diffusion on sufficiently long time scales. The fracture porosity could be obtained from short-term measurements or derived from fracture-network modelling (using either detailed numerical models or simple approximations).

The algorithm appears to be reasonably robust and allow calculations for a range of parameters. However, it should be noted that in general, smaller time steps will be required to model transport with rock-matrix diffusion than would be necessary for a model without rock matrix diffusion and with the porosity taken to be equal to the accessible porosity in the rock matrix (because the groundwater velocity will be faster and modelling the details of the diffusion into the rock matrix will be of interest).

The rock-matrix diffusion algorithm has currently been implemented in CONNECTFLOW only for the transport of salinity. However, it would be possible to use the currently implemented algorithm to model migration of other solutes, which might be sorbing. It would also be possible to use the algorithm to model the migration of a radionuclide, although it would be necessary for the user to post-process the calculated concentrations to take into account the effects of radioactive decay.

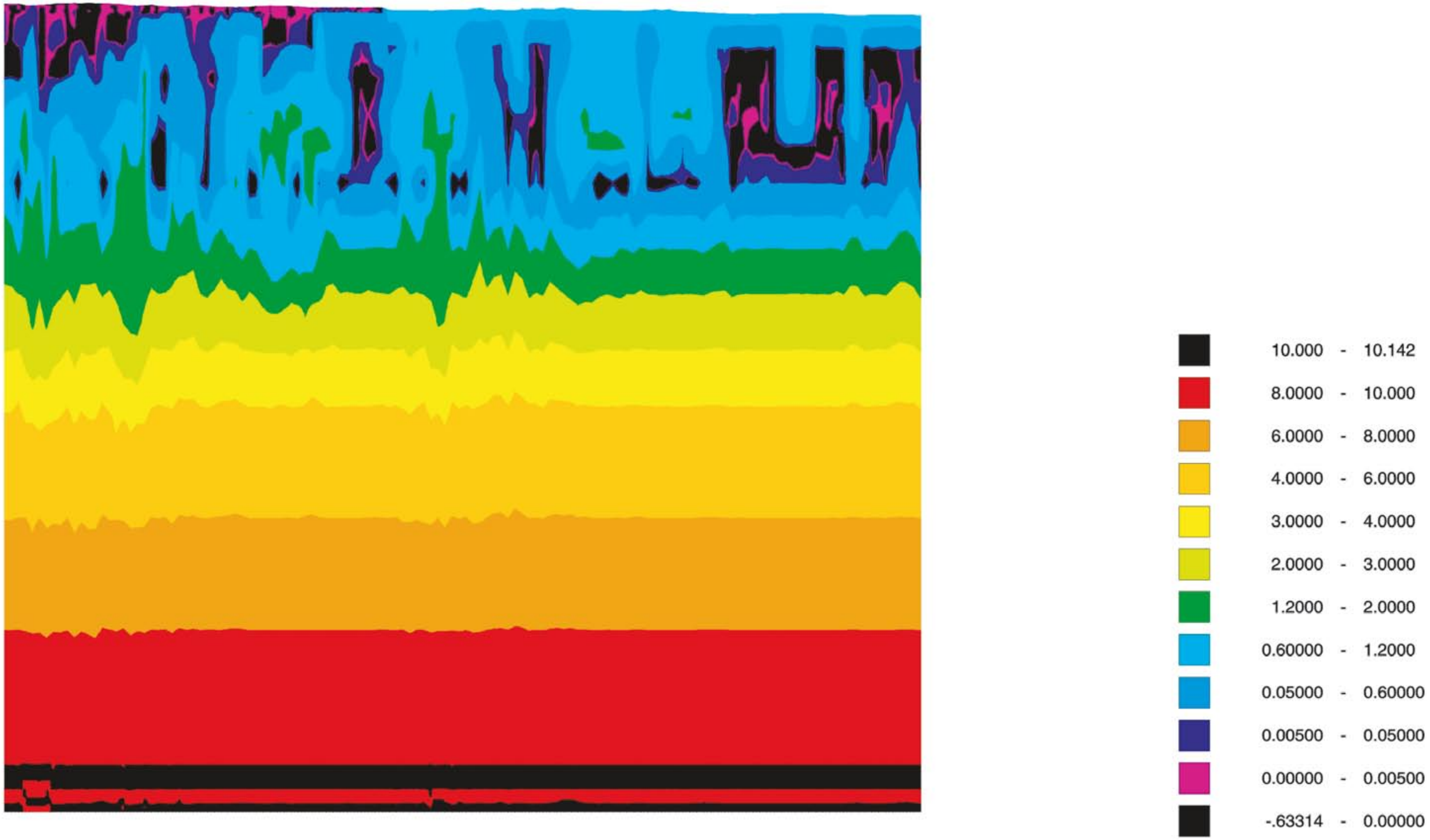

Figure 8-1. *Final distribution of salinity in the Base Case model (without rock-matrix diffusion) on a grid slice that runs from Southwest to Northeast through the borehole KFM02A. Note that the vertical scale has been exaggerated by a factor of five; the salinity should range from 0% to 10% by weight.*

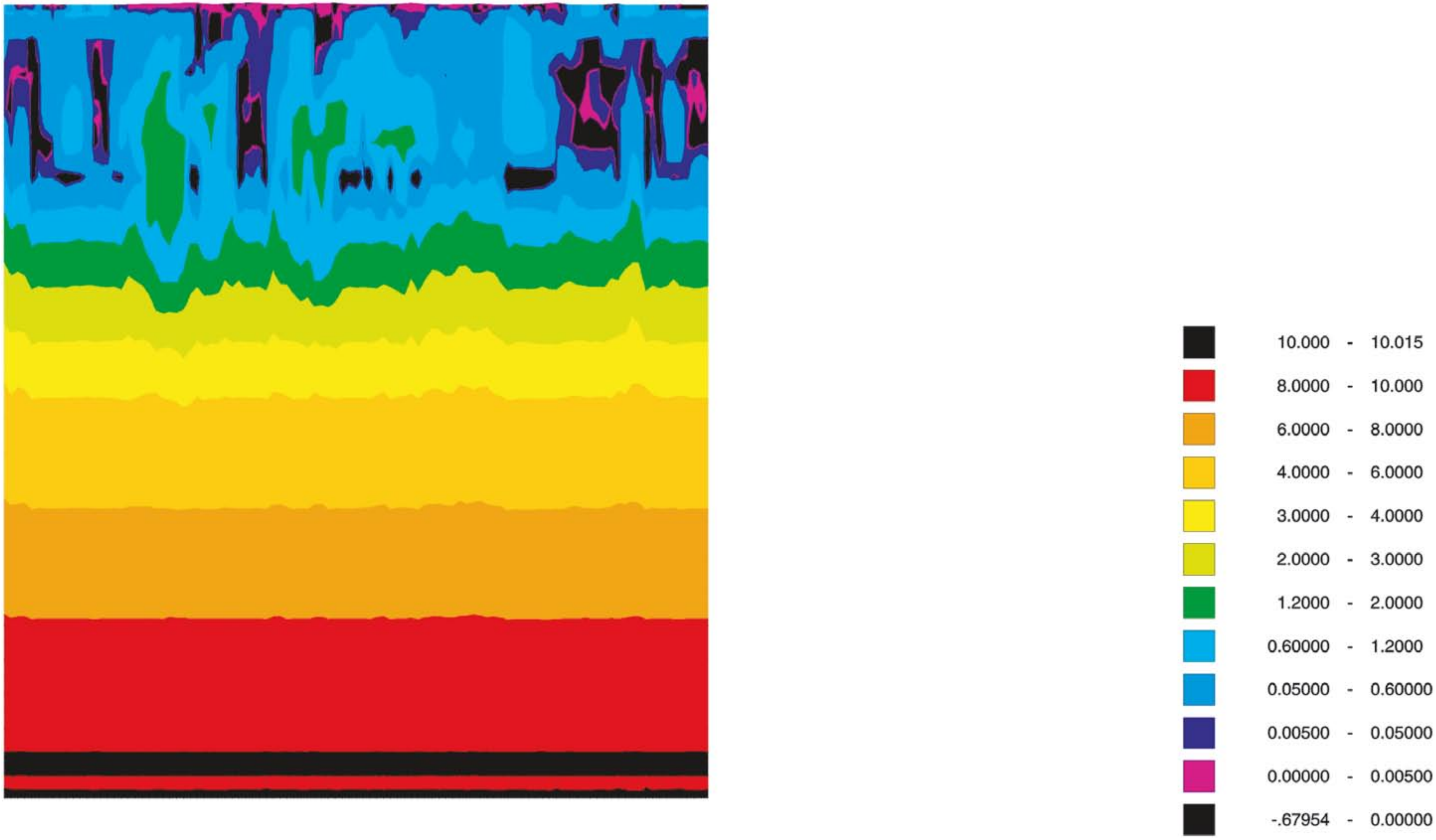

Figure 8-2. *Final distribution of salinity in the Base Case model (without rock matrix diffusion) on a grid slice that runs from Southeast to Northwest through the borehole KFM02A. Note that the vertical scale has been exaggerated by a factor of five; the salinity should range from 0% to 10% by weight.*

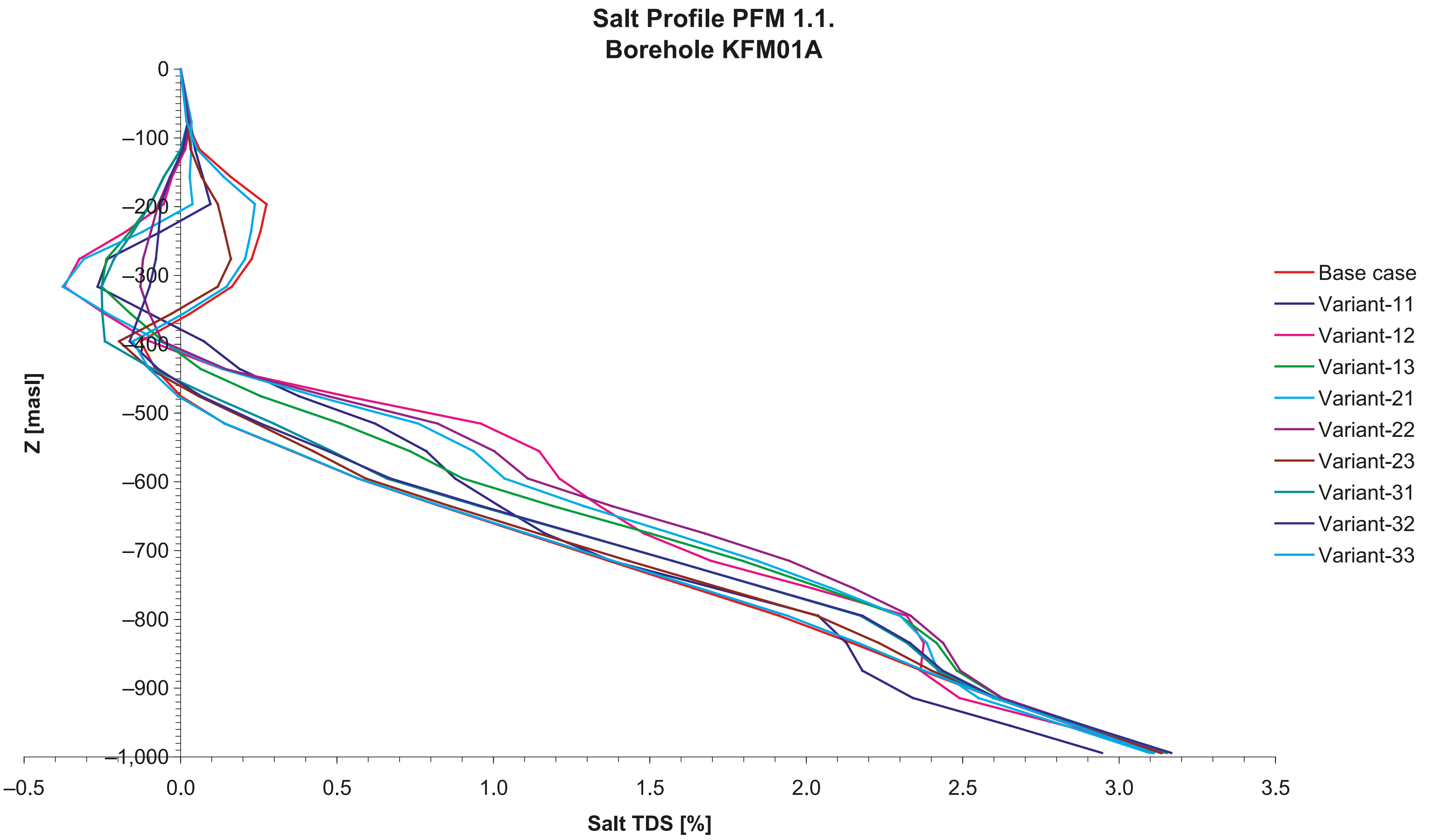

Figure 8-3. *Salt profiles (in the continuum) at 2000 AD for the simulated borehole KFM01A for the various cases. The Base Case is without rock-matrix diffusion, whereas rock-matrix diffusion is used in all the variants.*

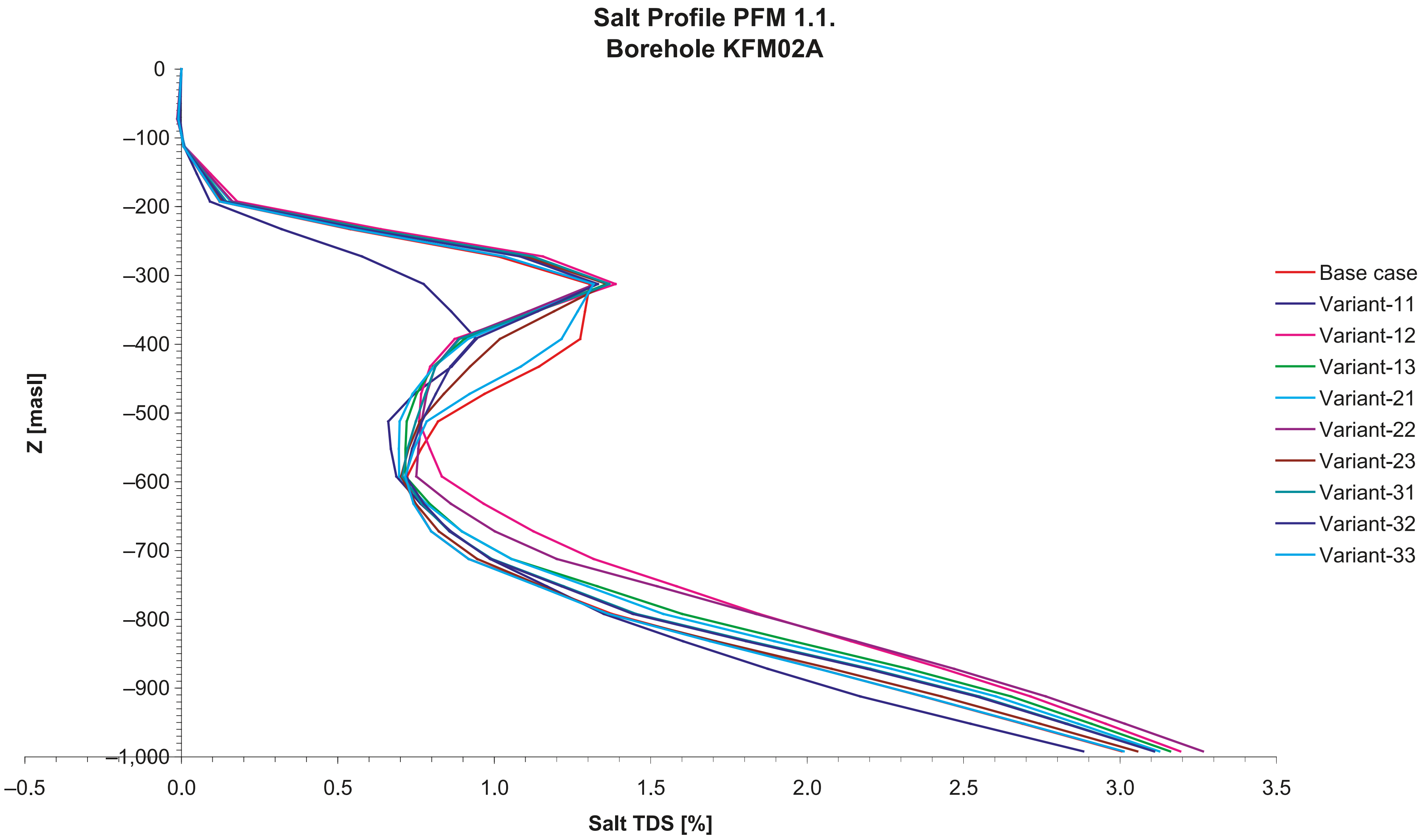

Figure 8-4. *Salt profiles (in the continuum) at 2000 AD for the simulated borehole KFM02A for the various cases. The Base Case is without rock-matrix diffusion, whereas rock-matrix diffusion is used in all the variants.*

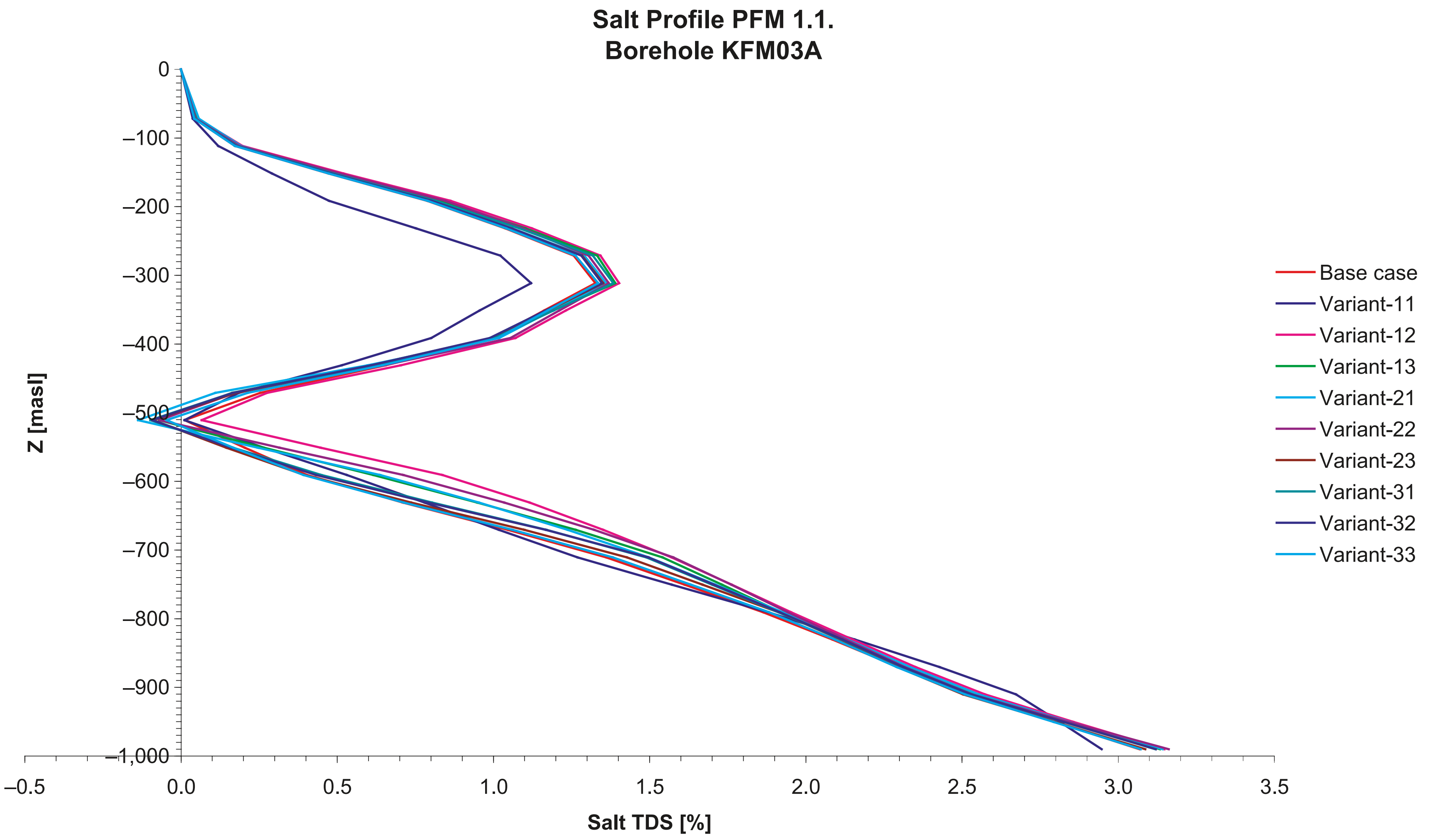

Figure 8-5. *Salt profiles (in the continuum) at 2000 AD for the simulated borehole KFM03A for the various cases. The Base Case is without rock-matrix diffusion, whereas rock-matrix diffusion is used in all the variants.*

9 References

/1/ **NAMMU, 2003.** Release 7.2 – User Guide, Serco Assurance Report SA/ENV-0627.

/2/ **NAMMU, 2003.** Release 7.2 – Technical Summary Document, Serco Assurance Report SA/ENV-0626.

/3/ **NAMMU, 2003.** Release 7.2 – Verification Document, Serco Assurance Report SA/ENV-0629.

/4/ **Hartley L J, Herbert A W, Wilcock P M, 1976.** NAPSAC (Release 4.0) Summary Document, AEA Technology report AEA-D&R-0271.

/5/ **Hartley L J, 2003.** Personal Communication.

/6/ **Löfman J, 1999.** Site Scale Groundwater Flow in Olkiluoto, POSIVA report 99-03.

/7/ **Herbert A W, Jackson C P, Lever D A, 1988.** Coupled Groundwater Flow and Solute Transport with Solution Density Strongly Dependent upon Concentration, Water Resources Research 24, pp 1,781–1,795.

/8/ **Talbot A, 1979.** The Accurate Numerical Inversion of Laplace Transforms, J. Inst. Math. Appl. 23.

/9/ **Carrera J, Sanchez-Vila X, Benet I, Medina A, Galarza G, Guimera J, 1998.** On Matrix Diffusion: Formulations, Solution Methods and Quantitative Effects, Hydrogeology Journal, 6, 1, p 178–190.

/10/ **R J Hopkirk and D J Gilby,** A Method for Modeling the Transport of Nuclides in Fissured Rock with Diffusion into the Solid Matrix, Nagra Technical Report 83-06, 1984.

/11/ **Cliffe K A, Herbert A W, 1990.** Matrix Diffusion User Guide (Release 3), AEA Report AEA-D&R-0052.

/12/ **Cliffe K A, 2003.** Personal Communication.

/13/ **Holmén J, Gylling B, Marsic N, Hartley L, Worth D, 2003.** Groundwater Flow Simulations: Forsmark Area Modelled with ConnectFlow, Memo: Preliminary Site Description (Version 1.1).

/14/ **Follin S, 2003.** Data delivery file PFM_V.1.1_Data_Delivery_SP1.xls.

/15/ **Lindgren M, Lindström F, 1999.** SR 97: Radionuclide Transport Calculations, SKB TR-99-23, Svensk Kärnbränslehantering AB.

/16/ **Huyakorn P S, Andersen P F, Mercer J W, White Jr H O, 1987.** Saltwater Intrusion in Aquifers: Development and Testing of a Three-Dimensional Finite Element Model, Water Resources Res., 23, p 293–312.