

**R-04-50**

**A direct numerical approach to  
solving the transport equations  
for radionuclide transport in  
fractured rock**

Fredrik Vahlund, Svensk Kärnbränslehantering AB

Harald Hermansson

Harald Hermansson Software Engineering AB

September 2006

**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-04-50

# **A direct numerical approach to solving the transport equations for radionuclide transport in fractured rock**

Fredrik Vahlund, Svensk Kärnbränslehantering AB

Harald Hermansson

Harald Hermansson Software Engineering AB

September 2006

*Keywords:* Radionuclide transport, Far field, Colloids, PROPER, MATLAB.

A pdf version of this document can be downloaded from [www.skb.se](http://www.skb.se)

## Abstract

A MATLAB program for simulating the colloid-facilitated radionuclide transport in fractured rock has been developed. The solution method is based on spatial discretisation of the governing partial differential equations and direct numerical integration in time using the MATLAB ODE solver ode15s. The model takes into account advection-dispersion transport of a number of radionuclides in solute and colloidal phases along a stream line, including retention by transversal diffusion into the aqueous rock matrix and chain decay. It would be easy to extend the model to allow for physical parameters that vary in space.

The model has been validated by comparisons with results obtained by the SKB program FARF31, using the standard set of FARF31 test cases, and also with results obtained by the COLLAGE II program.

The MATLAB program can certainly be used for detailed studies in a deterministic setting, but it can be questioned if stochastic studies of PROPER type are currently a feasible option due to the CPU-time requirements of the program. Therefore, the same model was also implemented in FORTRAN and designed to be used as ordinary submodels of the PROPER package. In this implementation the ODE solver used is one from the ODEPACK suite of Lawrence Livermore National Laboratories. This new PROPER submodel is called FARF33.

In order to save computation time in the so called tunnel case, where there is no diffusion in the transversal direction, a simplified PROPER submodel called FARF32 was also developed.

This document is intended to serve both as a detailed description of the physical problem and the implementation of the computer codes and as a user guide for the new computer codes and PROPER submodels.

# Sammanfattning

Ett MATLAB-program för att simulera kolloidpåverkad transport av radionuklider i sprickigt berg har utvecklats. Lösningemetoden baseras på rumsdiskretisering av de styrande partiella differentialekvationerna och direkt numerisk integration i tiden genom användandet av MATLABs ODE-lösare ode15s. Modellen tar hänsyn till transport genom advektion och dispersion av ett antal radionuklider i kolloidform och löst form längsmed en strömlinje, inklusive fördröjning på grund av transversell diffusion in i den vattenförande berggrunden och kedjesöderfall. Det skulle vara enkelt att utöka modellen för att tillåta fysikaliska parametrar som varierar i rummet.

Modellen har validerats genom jämförelser med resultat från SKB-programmet FARF31, på de standardiserade FARF31-testfallen, och också med resultat från programmet COLLAGE II.

MATLAB-programmet kan absolut användas för detaljerade studier för deterministiska fall, även om det är tveksamt om det i nuläget kan användas för stokastiska studier av PROPER-typ på grund av den CPU-tid som erfordras. Därför implementerades samma modell även i FORTRAN på ett sätt som gör det möjligt att använda den som en vanlig submodell i PROPER-paketet. Den ODE-lösare som användes i denna implementation är en ur ODEPACK-sviten från Lawrence Livermore National Laboratories. Denna nya PROPER-submodell kallas FARF33.

För att spara beräkningstid i det så kallade tunnelfallet, där diffusion vinkelrätt in i tunnelväggen saknas, skapades även en förenklad PROPER-submodell kallad FARF32.

Detta dokument är avsett både som en detaljerad beskrivning av det fysiska problemet och implementeringen av datorprogrammen och som användarmanual för de nya programmen och PROPER-submodellerna.

# Contents

<b>1</b>	<b>Introduction</b>	7
<b>2</b>	<b>The mathematical model</b>	9
2.1	Implementing advection and dispersion in the stream tube	9
2.2	Implementing diffusion in the rock matrix	11
2.3	Implementing colloids	12
2.4	The tunnel case	13
2.5	A note on downstream boundary conditions	14
<b>3</b>	<b>Spatial discretisation and computational technique</b>	15
3.1	Minimising the bandwidth of the system matrix	16
3.2	Evaluating the retention term	16
3.3	Time integration in the FORTRAN implementation	16
<b>4</b>	<b>Validation</b>	19
4.1	The FARF31 test batch	19
4.1.1	Problem 1	19
4.1.2	Problem 2	20
4.2	Comparison with COLLAGE II PLUS	21
4.2.1	No colloidal facilitated migration	21
4.2.2	Problem C997	22
4.2.3	Problem C999	22
4.3	Colloid facilitated migration in a KBS-3 repository	23
4.4	A test case for the tunnel model	26
4.5	A test case for FARF33	27
4.6	A test case for testing colloids	27
<b>5</b>	<b>Conclusions and suggestions</b>	29
<b>6</b>	<b>References</b>	31
<b>Appendix 1</b>	A user's guide for the programs	33
<b>Appendix 2</b>	Code listing	41
<b>Appendix 3</b>	Discretisation techniques	57

# 1 Introduction

SKB has previously developed a far-field radionuclide migration code called FARF31 for use with the probabilistic PROPER package /Norman and Kjellbert 1990/. This code has been in use for a number of years. The model includes advection and dispersion of a number of radionuclides along a streamline in fractured rock, with retention due to molecular diffusion into the transversal direction taken into account. This model also includes chain decay. The solution method is based on analytical solution in the Laplace domain followed by numerical inversion of the Laplace transform and convolution with the input function to obtain the output function.

It has recently been suggested that it can be important to take the effect of colloids into account when predicting of migration rates of radionuclides in ground water. This has been studied by Cvetkovic in /Cvetkovic 2003, Cvetkovic 2004/. A program called COLLAGE II Plus /Hicks 2003/ has been studied as a starting point for this project. The model behind this code is roughly equivalent with the FARF31 model, but it also includes a transport equations for radionuclides sorbed onto colloids. This code has no option to include more than one nuclide in the simulation, hence chain decay can not be simulated. The solution method is very similar to the FARF31 method, with analytic solution in the Laplace domain followed by numerical Laplace inversion. One difference is that the COLLAGE II Plus program uses a physical length scale in the streamwise direction while FARF31 employs a coordinate transformation to express the problem in terms of groundwater travel time instead. Another difference is that COLLAGE II Plus sets a downstream boundary condition for the concentration at a finite distance from the measurement point, while FARF31 sets it to zero only at the limit as ground water travel time goes to infinity.

It would probably be possible to extend the FARF31 solution scheme, using recursive functions for the analytic solutions to the transport equations in the Laplace domain, with a colloid phase in addition to the solute as long as only unidirectional transfer between the phases is allowed. But in this project, another more general approach has been chosen instead. The transport equations, including both solute and colloid phases for a number of nuclides, are discretised using a finite volume discretisation in both the physical and temporal dimensions. The resulting system of ordinary differential equations is then solved using the standard ODE-solver Ode15s of MATLAB. This makes it easy to extend the model to allow for physical parameters to vary in the streamwise direction, and even to make the model non-linear by letting the physical parameters depend on the concentrations.

Another objective of the project was to evaluate the possibility of letting MATLAB be a standard tool in the safety analysis process of SKB. As we came to the conclusion that the ODE solvers built-in to MATLAB were still too slow for large scale stochastic simulations, we also implemented the same model in FORTRAN as a PROPER submodel. In order to do this we had to find a replacement for ode15s as ODE solver. The choice fell on LSODE, one of the solvers in the ODEPACK suite of solvers from the Lawrence Livermore National Laboratories. This new PROPER submodel is called FARF33. Another advantage of this approach is that it is fully possible to call a FORTRAN program from MATLAB.

In order to save even more computation time in the so-called tunnel case where the retention term is missing, we also developed a simplified PROPER submodel called FARF32 based on the same methodology and ODE solver.

## 2 The mathematical model

The finite volume method is, like the finite difference method and the finite element method, a numerical scheme used to discretise the system of equations needed to solve, in the present implementation, a transport problem. Perhaps the most logical starting point when introducing the finite volume method is to start with a conservation law for a scalar quantity  $c^i$  over an arbitrary volume  $d\Omega$

$$\frac{\partial}{\partial t} \int_{\Omega} c^i dV + \oint_S \bar{F} \cdot dS = \int_{\Omega} Q_v dV + \oint_S \bar{Q}_s \cdot dS \quad (1)$$

using  $\bar{F}$  to represent fluxes, over the boundaries,  $S$ , of the volume  $dV$  and  $Q$  for surface (index  $s$ ) and volume sources (index  $v$ ). This integral representation simply reads that the change in the quantity  $c^i$  in the volume  $dV$  corresponds to the flux over the boundaries,  $S$ , of  $dV$  and the amount of the quantity that is generated/destroyed inside the volume or at the boundaries. Moreover, applying Gauss's theorem on Equation 1 yields the differential form of the conservation law,

$$\frac{\partial c^i}{\partial t} + \nabla \cdot (\bar{F} - \bar{Q}_s) = Q_v \quad (2)$$

which corresponds to that used in the FARF31 /Norman and Kjellbert 1990/ and COLLAGE II Plus /Hicks 2003/ codes.

By determining the fluxes, (which, in a diffusive/dispersible case are functions of the concentration gradient and in an advective case are functions of the concentration and advective velocity) and the sources as function of the concentration, a system of equations on the form

$$\frac{\partial c^i}{\partial t} = f(c^i, t) \quad (3)$$

may be obtained. Initially, the problem is hence to express sources and fluxes on a form so that the conservation law may be written on a form corresponding to Equation 3. To do this, the conservation law for the sorbed nuclides, the nuclides migrating through the rock matrix and those migrating with the colloidal phase must be determined.

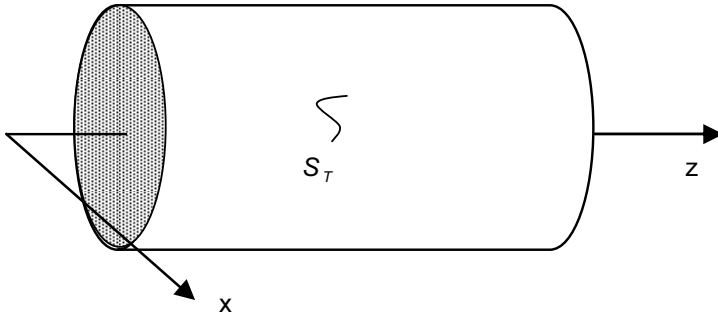
### 2.1 Implementing advection and dispersion in the stream tube

Nuclides solved in (and hence migrating with) the ground-water are both in the FARF31 and in the Collage II Plus code /Hicks 2003/ modelled using the advection-dispersion equation. The flux vector  $\bar{F}$  in Equation 1 is hence written as the sum of an advective,  $\bar{F}_A = u_s c_s$ , and a dispersive component,  $\bar{F}_D = -D_s \nabla c$ , where  $u_s$  represents the Darcy velocity,  $D_s$  the dispersion\* coefficient and  $c_s$  the concentration in the solute.

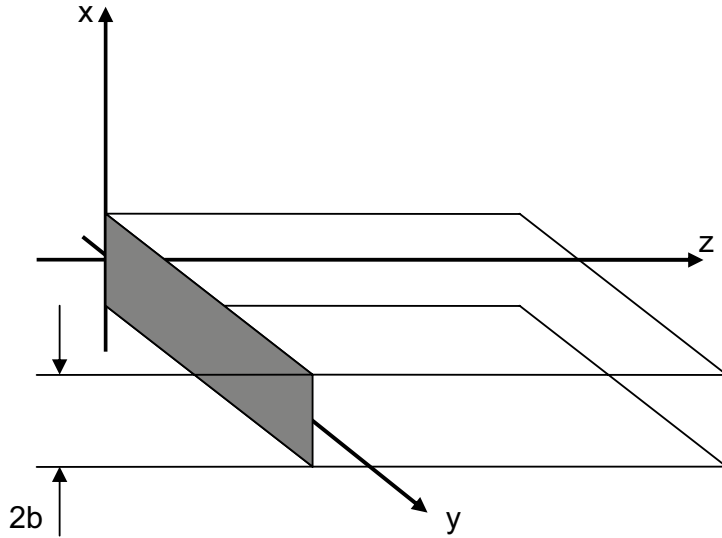
According to the stream tube conceptualisation, (see the manuals for FARF31 /Norman and Kjellbert 1990/ and Collage II Plus /Hicks 2003/ and Figure 2-1) both the advective and the dispersive fluxes are defined to be in the direction of the stream tube (the  $z$ -direction in the current implementation). Moreover, the FARF31 conceptualisation allows nuclides to migrate from the stream-tube into the rock matrix. In the rock matrix it is transported by diffusion in

---

\* here, dispersion represents also molecular diffusion in the stream-wise direction.



**Figure 2-1a.** Coordinate system used in the present report to represent the stream tube conceptualisation.



**Figure 2-1b.** Coordinate system used in the present report to represent the Collage II Plus conceptualisation.

a transversal direction, which is thought of as radial in the FARF31 conceptualisation and perpendicular to the fracture in the conceptualisation of Collage II Plus. In both cases one ends up with the same set of equations.

To encounter for this transverse diffusive flux over the lateral area of the stream tube,  $S_T$ , a surface source term is introduced in Equation 1.

$$\oint_{S_T} \overline{Q_s} \cdot dS = a_w d\Omega D_e \frac{\partial c_p}{\partial x} \Big|_{x=0} \quad (4)$$

Where  $a_w$ , (in line with the FARF31 notation) corresponds to the total surface area at the interface per volume of mobile liquid. Further, the effect of decay and in growth may be written as a volume source term

$$Q_v = -\lambda^i c^i + \lambda^{i-1} c^{i-1} \quad (5)$$

where  $\lambda^i$  represents the decay coefficient of nuclide  $i$  and  $c^{i-1}$  corresponds to the concentration of the mother nuclide. Including these terms in the conservation equation, Equation 1, the integral form of the advection-dispersion equation with transverse diffusion and nuclide decay may be written



$$\frac{\partial}{\partial t} \int_{\Omega} c^i_s dV + \oint_{S_L} \left( u c^i_s - D_s \frac{\partial c^i_s}{\partial z} \right) \cdot dS_L = \int_{\Omega} (\lambda^{i-1} c^{i-1}_s - \lambda^i c^i_s + Q_{colloid}) dV + \oint_{S_T} a_w D_e \frac{\partial c^i_p}{\partial x} \Big|_{x=0} dS_T \quad (6)$$

where  $Q_{colloid}$  represents the exchange term between the solute and the colloid phase (introduced in the subsequent section),  $S_L$  corresponds to the area in the longitudinal direction. Further, the boundary condition at the inlet boundary for Equation 6 may be expressed as

$$2b \left[ u_s c^i_s \Big|_{inlet} - D_s \frac{\partial c^i_s}{\partial z} \Big|_{inlet} \right] = \eta In(t) \quad (7)$$

where  $b$  represents the half-width of the fracture,  $In(t)$ , the time dependent inflow of nuclides and  $\eta$ , the fraction of the nuclide inflow that occurs in the solute (as opposed to the colloid) phase. The factor  $2b$  is the inlet area per unit length for the fracture, and the relation between this parameter and the stream tube area of the FARF31 conceptualisation is studied in detail in section A3.6 below. The same relation can be applied to both inlet and outlet boundary conditions.

At the output boundary, at  $z=L$ , two different boundary conditions have been implemented in the FORTRAN version. The first is a homogeneous boundary condition assuming zero concentration outside the downstream boundary. The second is a free boundary condition where the concentration outside the boundary is calculated using extrapolation and the out-flux is given by

$$Out(t) = 2b \left[ u_s c^i_s \Big|_{outlet} - D_s \frac{\partial c^i_s}{\partial z} \Big|_{outlet} \right] \quad (8)$$

The MATLAB version has only the free boundary condition. The initial condition used in the model is that there are no nuclides present at  $t=0$ .

## 2.2 Implementing diffusion in the rock matrix

In addition to the longitudinal flux, modelled with the advection-dispersion equation, the nuclides are also assumed to be transported into the porous rock matrix adjacent to the fracture. Inside the matrix, nuclides will diffuse both in the water phase through the porous network and through surface diffusion in the solid rock. In the present implementation only transverse flux (in the x-direction) is regarded and the diffusive flux,  $F_p$  in the porous media is modelled using a 1-D Fickian like model

$$F_p = -D_a^i \frac{\partial c^i_p}{\partial x} \quad (9)$$

where  $c^i_p$  represents the concentration of species  $i$  in the void water and  $D_a$  corresponds to an apparent diffusivity coefficient which accounts for the lumped effect of porosity and sorption. Implementing a 2D diffusional model instead of a 1D is straightforward, this has however not been conducted since it prevents accurate comparisons with the reference codes. Following the methodology used in the FARF31 code, the apparent diffusivity is related to the effective diffusivity

$$D_a = \frac{D_e}{R} \quad (10)$$

where the retention factor,  $R$ , is expressed as

$$R^i = \varepsilon^i + K_d^i \rho \quad (11)$$

being a function of the porosity,  $\varepsilon$ , the sorption coefficient,  $K_d$  (from which the concentration in the solid phase,  $c_{solid}$ , may be determined,  $c_{solid} = K_d^i c_p$ ) and the bulk density of the rock,  $\rho$ . Note that this definition differs from the retention coefficient in the Collage II Plus code /Hicks 2003/ and hence the definition of the apparent diffusivity. If the conservation equation is applied for a diffusive flux  $F = F_p$ , Equation 1 may be written

$$\frac{\partial}{\partial t} \int_{\Omega} c_p^i dV - \oint_S \frac{D_e^i}{R^i} \frac{\partial c_p^i}{\partial x} \cdot dS = \int_{\Omega} \left( \frac{R_{i-1}}{R_i} \lambda^{i-1} c^{i-1}_p - \lambda^i c^i_p \right) dV \quad (12)$$

where the decay has been implemented as a volume source term in a corresponding way as in the advection diffusion equation. In the present implementation, the following boundary conditions are used

$$c_p^i = c_s^i, \text{ at the interface } x = 0, \text{ and } \frac{\partial c_p^i}{\partial x} = 0 \text{ at } x = L \quad (13)$$

where  $L$  (this time in the x-direction) is the length to a symmetry boundary (half the length to the closest stream tube or fracture).

An initial condition of  $c_p^i = 0$  at  $t = 0$  is used for the whole domain.

## 2.3 Implementing colloids

In addition to the solute transport, modelled using the advection-dispersion equation, colloid facilitated migration is also regarded as a possible migration mechanism for radionuclides. This is modelled in a way corresponding to that of the Collage II Plus code /Hicks 2003/ where colloid facilitated migration was added as a source term in the advection-dispersion equation. In that implementation, nuclides were assumed to adsorb reversibly to colloids and to migrate with the colloids at an advection velocity not necessarily the same as that for solute transport. In the Collage II Plus implementation /Hicks 2003/ colloids were assumed to be either mobile or immobile (trapped in the fracture) and that there is a linear relation between the fixed and the mobile concentrations.

To encounter for the effect of nuclides being sorbed/desorbed onto colloid in the present finite volume implementation an additional source term,  $Q_{colloid}$ , was added to the advection-dispersion equation (using the same notation as in Collage II Plus)

$$Q_{colloid} = (\kappa_1 + \beta \kappa_2) c_m^i - (\kappa_1 k_1 + \kappa_2 k_2) c_s^i \quad (14)$$

where  $c_m$  corresponds to the concentration of species  $i$  sorbed to mobile colloids.  $k_1$  and  $k_2$  are partition coefficients for radionuclides sorbed to mobile and immobile colloids respectively.  $\kappa_1$  and  $\kappa_2$  correspond to the rate of sorption-desorption onto mobile and immobile colloids respectively, whilst  $\beta$  represents the ratio of immobile to mobile colloids (if  $\beta=0$ , only mobile colloids exists).

In addition to the source term in the advection-dispersion equation, which represents transfer between the solute and the colloid phases, another transport equation is needed to model the mobile colloids

$$\begin{aligned} & \frac{\partial}{\partial t} \int_{\Omega} c_m^i (1 + \beta) dV + \oint_S u_m c_m^i - D_m \frac{\partial c_m^i}{\partial z} dS \\ & = \int_{\Omega} \left[ \lambda^{i-1} c^{i-1}_m - \lambda^i c^i_m - (\kappa_1 + \beta \kappa_2) c_m^i + (\kappa_1 k_1 + \kappa_2 k_2) c_s^i \right] dV \end{aligned} \quad (15)$$

where an advective flow rate  $u_m$  and a dispersion coefficient  $D_m$  has been introduced for the colloid transport. The inlet boundary condition is implemented in a corresponding way as for the solute phase

$$2b \left[ u_m c_m^i \Big|_{inlet} - D_m \frac{\partial c_m^i}{\partial z} \Big|_{inlet} \right] = (1-\eta) In(t) \quad (16)$$

where  $b$  represents the half width of the fracture,  $In(t)$ , the time dependent inflow of nuclides and  $\eta$ , the fraction of the nuclide inflow that occur in the solute (as opposed to the colloid) phase.

At the output boundary, at  $z=L$ , two different boundary conditions have been implemented in the FORTRAN version. The first is a homogeneous boundary condition assuming zero concentration outside the downstream boundary. The second is a free boundary condition, where the concentration outside the boundary is calculated using extrapolation and the out-flux is given by

$$Out(t) = 2b \left[ u_m c_m^i \Big|_{outlet} - D_m \frac{\partial c_m^i}{\partial z} \Big|_{outlet} \right] \quad (17)$$

The MATLAB version has only the free boundary condition. As for the solute phase, the initial condition is that there are no nuclides present at  $t=0$ .

## 2.4 The tunnel case

For the simplest conceivable model, which is similar to the two-dimensional transport problem described above, we use the term “the tunnel case”. In addition to the FARF31 and FARF33 conceptualisations with a two-dimensional geometry, a second advection-dispersion model assuming a one-dimensional geometry has also been developed. This model gives much faster execution in cases where transversal matrix diffusion can be neglected.

In the tunnel case, the governing equation can be obtained from (6) by removing the transversal equation and the retention term, and scaling the time derivative and right hand side with a retention factor depending on the porosity, distribution coefficient and density of the backfill

$$R_i \frac{\partial}{\partial t} \int_{\Omega} c_s^i dV + \oint_{S_L} \left( u c_s^i - D_s \frac{\partial c_s^i}{\partial z} \right) \cdot dS = \int_{\Omega} (R_{i-1} \lambda^{i-1} c^{i-1}_s - R_i \lambda^i c^i_s + Q_{colloid}) dV \quad (18a)$$

If we further assume that the transport properties of the solute and the colloid phase are the same, there is no longer any difference between the solute and the colloid phase, and we end up with only one equation for the sum of the concentration in both phases

$$R_i \frac{\partial}{\partial t} \int_{\Omega} c^i dV + \oint_{S_L} \left( u c^i - D \frac{\partial c^i}{\partial z} \right) \cdot dS = \int_{\Omega} (R_{i-1} \lambda^{i-1} c^{i-1} - R_i \lambda^i c^i) dV \quad (18b)$$

This is the transport equation implemented in the tunnel model, for brevity called FARF32.

The retention factor,  $R_i$ , is modelled as

$$R_i = \varepsilon_i + (1 - \varepsilon_i) K_{d,i} \rho$$

## 2.5 A note on downstream boundary conditions

When using Laplace methods for the solution, as in FARF31, it is possible to set a boundary condition for the concentration as the stream tube length or travel time goes to infinity. This possibility no longer exists in a finite volume discretisation where the computational domain must be finite. What boundary condition to use downstream in this case is really a modelling issue. Therefore, we have implemented two different downstream boundary conditions. One approach is that the dilution is so large downstream of a certain model that the concentration can be neglected compared to the concentration inside the boundary. Using this assumption is equivalent to using the homogeneous downstream boundary condition.

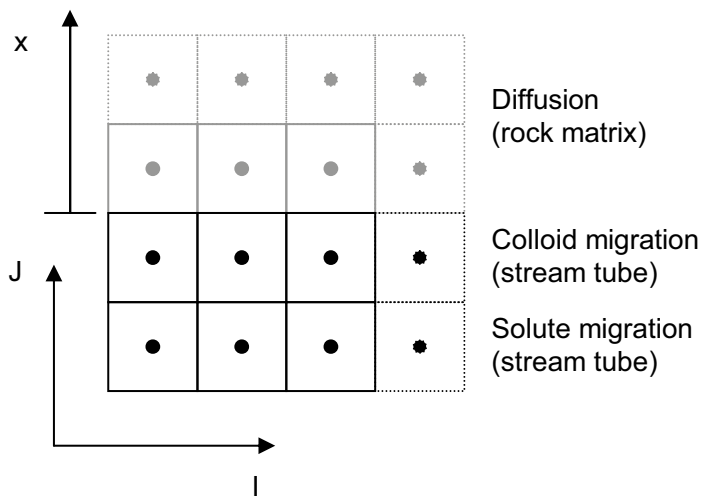
In the presence of colloids we are likely to have two distinct peaks in the solution curve. Then it could be a better approximation to use some kind of extrapolation boundary condition downstream to allow diffusion from the outside back into the computational domain, a situation which could occur between the two peaks long before the time period we are interested in ends. This corresponds to using the free boundary condition.

In the PROPER implementation both choices are available.

### 3 Spatial discretisation and computational technique

In the present finite volume implementation, the computational domain was subdivided into a number ( $NI \times NJ$ ) of hexahedral elements. Each element has a node located in the centre of the element in which the concentration of each nuclide is calculated, Figure 3-1. As fluxes need to be evaluated at the boundaries of an element different interpolation techniques are required to estimate these based on the concentration in adjacent nodes. To do this, the computational domain is divided into three distinct parts, the stream tube with solute transport ( $J=1$ ), the stream tube with colloidal transport ( $J=2$ ) and the rock matrix ( $J>2$ ). The dominating part of the computational domain represents hence diffusional transport in the rock matrix. The discretisation method of the rock matrix transport equation will therefore have large impact of the size of the problem and it is important to have as few elements as possible (but not too few) in the transverse direction. The second component,  $I$ , corresponds to the longitudinal  $z$ -direction with the inlet boundary at  $I=1$  and the outlet boundary at  $I=NI$ .

After performing the spatial discretisation, a suitable discretisation scheme is required for the approximation of the fluxes, described in detail in Appendix 3. For the diffusive (in the rock matrix) and dispersive fluxes (in the stream tube) a central difference scheme (CDS) has been used where the boundary value is estimated using a linear interpolation between the adjacent nodes. For the advective discretisation scheme, the cell Peclet number ( $dz \cdot u_s / D_s$ ), is used to choose between the central difference scheme (when  $dz \cdot u_s / D_s \leq 2$ ) and the first order upwind difference scheme (UDS when  $dz \cdot u_s / D_s > 2$ ) which uses the concentration from the closest upstream node. In order to reduce the size of the mesh in the transverse direction, a non-equidistant mesh is used for the spatial discretisation of the rock matrix with smaller elements close to the fracture or stream tube. Since the influx is given at the boundary of the nodes with  $I=1$ , no special procedure is required when estimating the value at these boundaries. However, for the outlet boundaries, (when evaluating  $c_m^i|_{outlet}$  and  $c_s^i|_{outlet}$  in the stream tube and the boundary at  $x=0$  when evaluating  $\partial c_p^i / \partial x|_{x=0}$  for the rock matrix), there are no nodes outside of the computational domain. Hence, adjacent nodes ( $NI-2..NI$  for the stream tube and  $J=3..4$  for the rock matrix) are used to extrapolate the value at the boundaries.



**Figure 3-1.** Spatial discretisation with the origin of the transversal coordinate at the interface between the stream tube and the rock matrix.

After choosing a suitable discretisation scheme and carrying out the spatial discretisation, the migration problem may be formulated as a system of ordinary differential equation on the form.

$$\frac{\partial}{\partial t} \begin{Bmatrix} c_1 \\ \cdot \\ \cdot \\ c_n \end{Bmatrix} = [A] \begin{Bmatrix} c_1 \\ \cdot \\ \cdot \\ c_n \end{Bmatrix} + \begin{Bmatrix} B_1 \\ \cdot \\ \cdot \\ B_n \end{Bmatrix} \quad (19)$$

where  $c_{1..n}$  represents the concentration in nodes  $1$  to  $n$  ( $=N \times N$ ),  $A$  represents a matrix which is a function of the chosen schemes and  $B$  corresponds to the boundary conditions. This system of ordinary differential equations may then be solved using any suitable solver for this particular kind of problem. In the present implementation, a Matlab solver called ode15s was used. This is an ODE solver suitable for both stiff and non-stiff systems. This solver has, as one of its benefits, an adaptive time stepping method which enables for solving problems occurring over large timescales.

### 3.1 Minimising the bandwidth of the system matrix

One important consideration in maximising the computational speed of the computer program is to minimise the bandwidth of the system matrix  $A$ . In the FORTRAN implementation used for the new PROPER submodels we changed the order of the solution variables so that the concentrations of the various nuclides in one cell are grouped together. Since it is likely that the number of cells in the streamwise direction is larger than the number of cells in the transversal direction we also grouped the solution variables for all cells at the same streamwise position together at the next level. As a final optimisation of the solution scheme we switched the order of the solute and colloid cells. The resulting system matrix is very sparse so it is very favourable to use a sparse solver for the linear systems in the ODE solver. This is readily done in MATLAB, but in ODEPACK we used a banded solver instead.

### 3.2 Evaluating the retention term

In order to evaluate the retention term, the x-derivative of the concentration is needed at the interface between the fracture/stream tube and the rock matrix. The value is calculated by extrapolation from the cells in the rock matrix which are closest to the interface. We have tried both a first and second order extrapolation scheme. There is a small systematic difference in results, and using the lower order extrapolation gives an effect similar to using a slightly lower value for the aspect ratio. Using the lower order extrapolation simplifies calculation of the system matrix and makes it sparser. Therefore, we kept the simpler variant in the PROPER submodel FARF33.

### 3.3 Time integration in the FORTRAN implementation

The ODEPACK suite of ODE solvers was developed at the Lawrence Livermore National Laboratories in the 80's and has become the standard choice for anyone who needs a free ODE solver written in FORTRAN. We have used LSODE, the most basic solver capable of solving both stiff and non-stiff problems in an efficient manner. More information about ODEPACK can be found at the Lawrence Livermore website at <http://www.llnl.gov/CASC/odepack/>.

In PROPER all output time series are handled by the so called “time series manager”. This piece of code tries to eliminate all “uninteresting” solution points in order to minimise the space needed to store the output time series. This approach used in the time series manager of PROPER is not really useful in cases where the solution is not bell shaped in the sense that it contains one distinct peak. Nothing has been done in this project to solve this problem. Instead we have developed an algorithm which saves solution points in a manner so that the solution doesn’t get more destroyed by the time series manager than necessary.

## 4 Validation

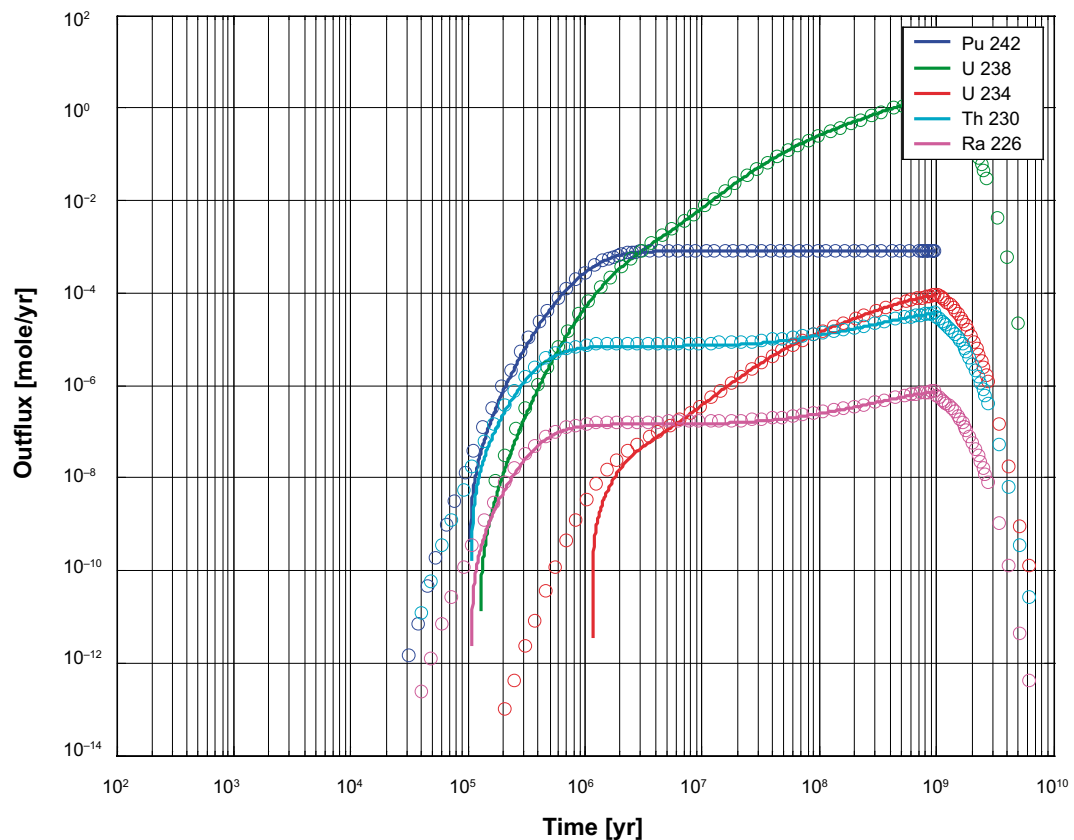
The program has been validated by running problem 1, 2, 3 and 6 from the standard set of FARF31 test cases (the FARF31 test batch) as documented in /Lindgren et al. 2002/. The simulation results have been compared to results obtained by running FARF31 and are documented below. The results generally show good agreement provided that the mesh resolution is good enough.

### 4.1 The FARF31 test batch

Two different test cases from the FARF31 test batch have been used to validate the FVFARF code. In both examples, chain decay has been used, hence comparisons with the Collage II Plus code were not possible. However, in subsection 4.2.1, all three codes have been tested using the same test case.

#### 4.1.1 Problem 1

In Figure 4-1, problem 1 from the FARF31 test batch is shown for simulations with FVFARF and FARF31. It can be seen that the agreement between the two codes are good except in the initial part of the simulation where there exists some deviation. This deviation is the same as that seen in problem 2; in the following section where a discussion about possible reasons for this deviation has been carried out.

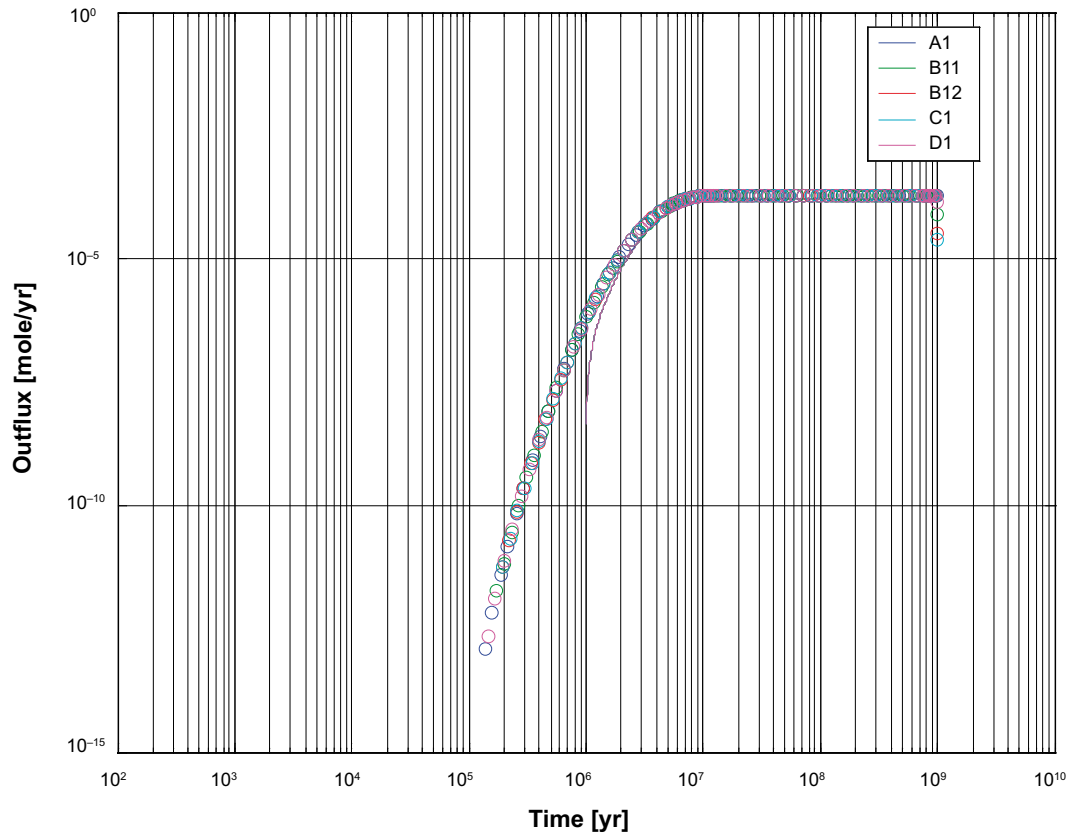


**Figure 4-1.** Problem 1 from the FARF31 test batch, chain decay. Rings denote results obtained by FVFARF and solid lines denote results obtained by FARF31.



### 4.1.2 Problem 2

Figure 4-2 shows the results from FARF31 and FVFARF for problem 2 in the FARF31 test batch. The example includes five different hypothetical nuclides and the test case has been designed to give equal results for all nuclides. It can be seen that the agreement between the FARF31 results and the FVFARF results are good in general. However, for small times, the same deviation as seen in problem 1 exists. For single nuclide runs, it can be shown that the agreement between the two cases increases as the number of elements in the transversal direction and the tolerances are adjusted (compare with the agreement for the test case in section 4.2.1).



**Figure 4-2.** Problem 2 from the FARF31 test batch, a case where the outflux should be the same for all nuclides. Rings denote results obtained by FVFARF and solid lines denote results obtained by FARF31.

## 4.2 Comparison with COLLAGE II PLUS

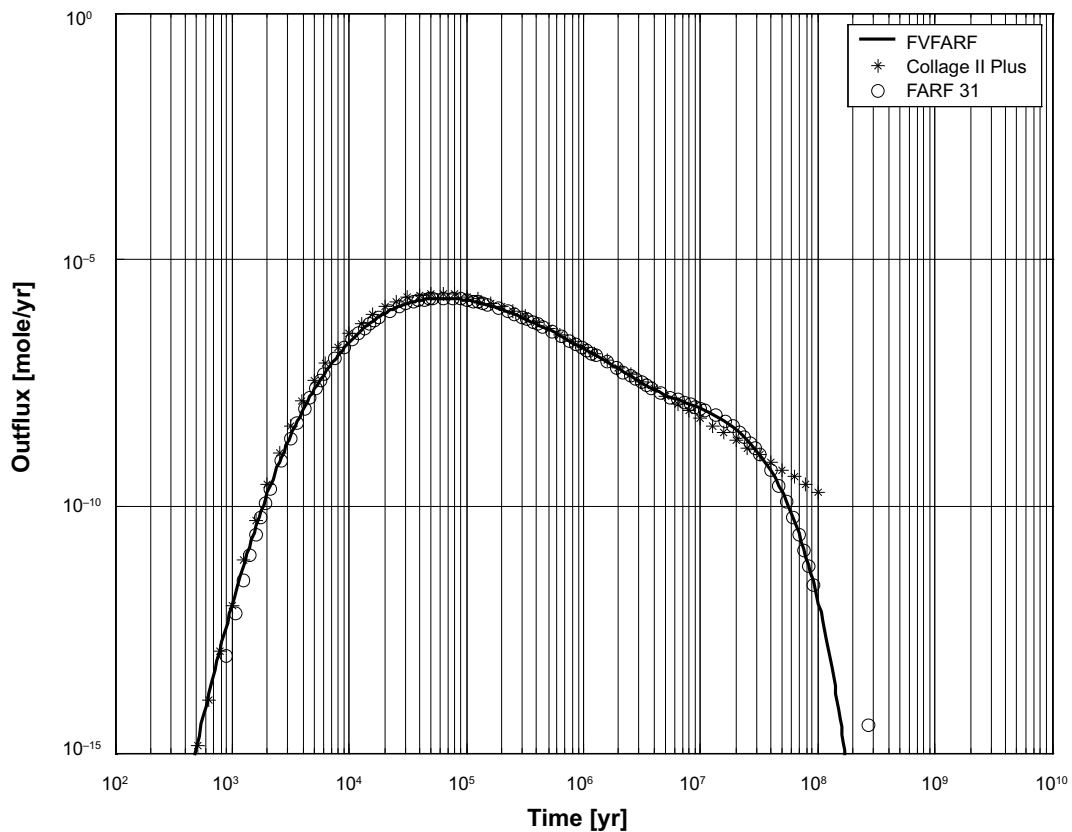
The program has also been validated by comparisons with test cases inspired by those in the COLLAGE II Plus manual /Hicks 2003/. Table 4-1 shows the difference in parameters used in the test cases based on the conversion procedure described in section A3.6.

### 4.2.1 No colloidal facilitated migration

Figure 4-3 shows the release rate for a test case without colloids. It can be seen that the agreement between the FVFARF, Collage II Plus and FARF31 is excellent except at large times. The reason for the deviation at  $t > 10^7$  years relates to the treatment of the transversal inner boundary where the FARF31 and the FVFARF uses a maximum penetration depth whilst the Collage II Plus model assumes infinite diffusion (by changing the Penetration depth the position where the curves starts to deviate can be changed).

**Table 4-1. Input parameters used by the FARF31 and the Collage II Plus models, respectively.**

FARF Data		Collage II Plus Data	
tw	1,000	u	1
Peclet	50	D	50
$\epsilon$	0.01	$\epsilon$	0.01
Penetration depth	5	R	675.1
Kd	0.25	L	1,000
De	$7.875 \cdot 10^{-4}$	b	$10^{-2}$
Aspec	1		
Half life	$10^{15}$	$\lambda$	0



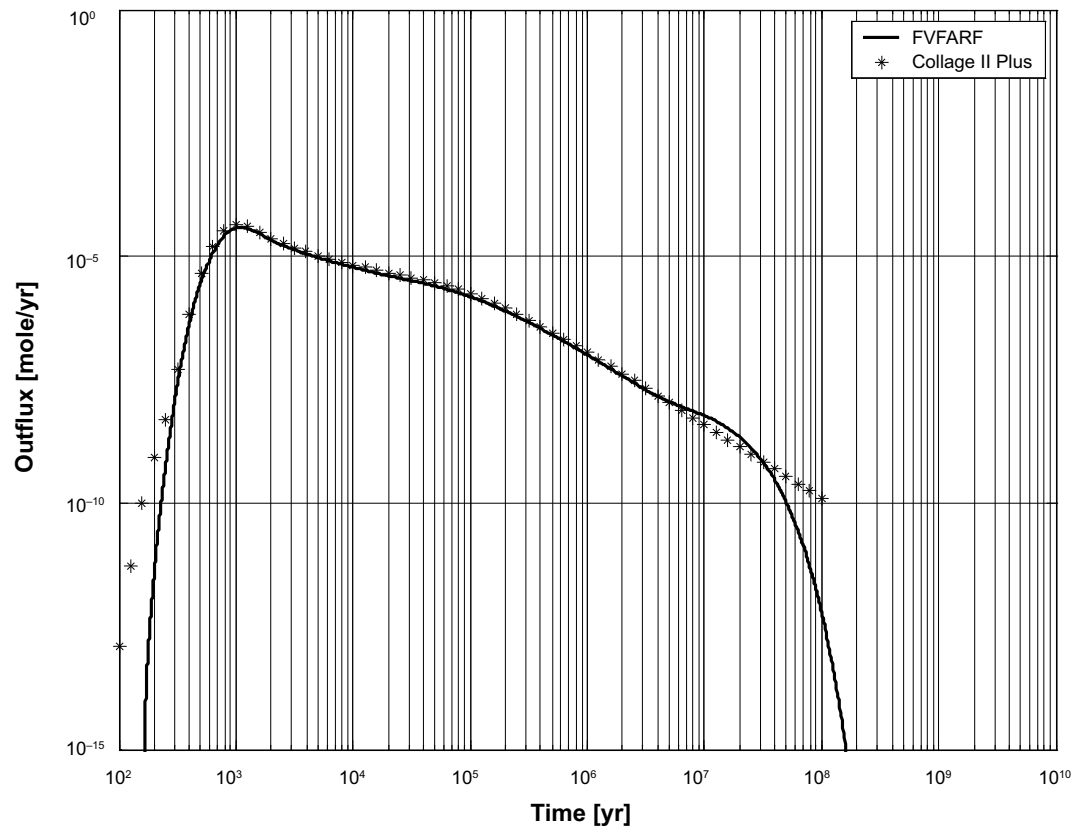
**Figure 4-3. Comparison between FVFARF, FARF31 and Collage II Plus for case without colloids.**

### 4.2.2 Problem C997

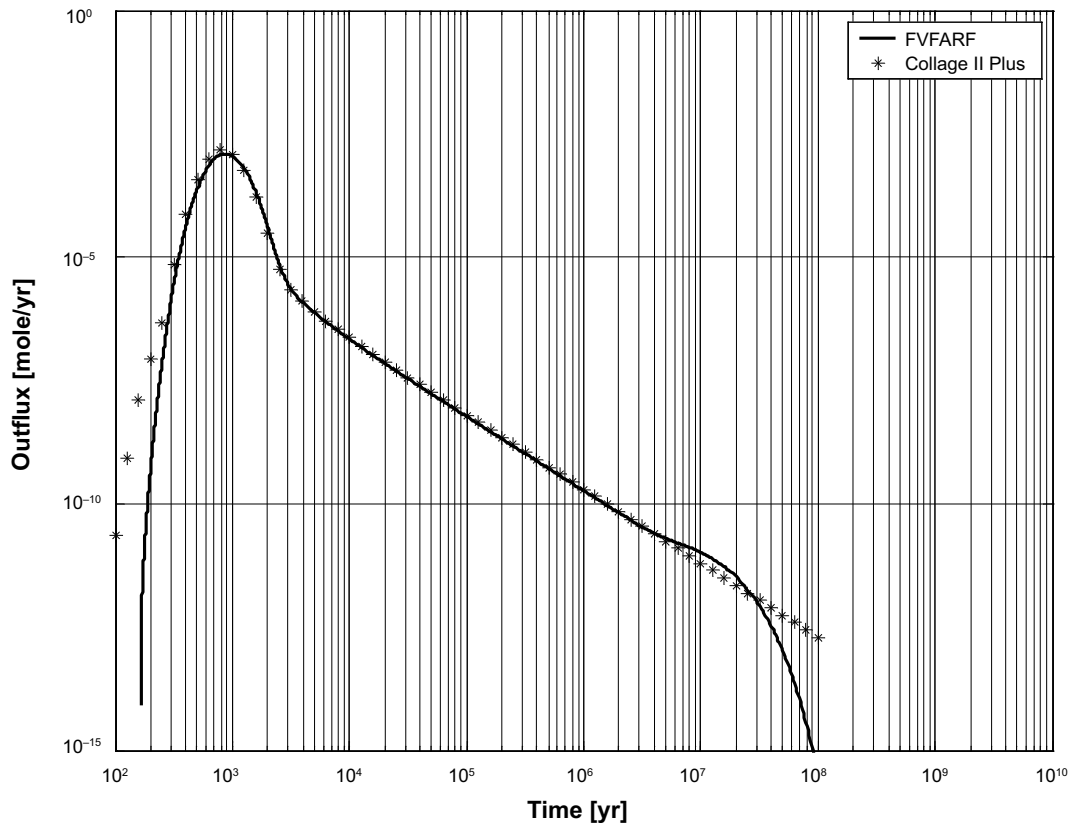
Figure 4-4 shows the results for a simulation with a slow sorption-desorption rate between mobile and immobile colloids. It can be seen that the agreement between the two cases is good.

### 4.2.3 Problem C999

Figure 4-5 shows the results for a test case with colloids and a fast sorption-desorption rate. Also these results are good.



**Figure 4-4.** Comparison between FVFARF and Collage II Plus for a case with slow sorption-desorption from mobile and immobile colloids ( $\kappa_1 = \kappa_2 = 10^{-6} \text{ y}^{-1}$ ). Radionuclides enter the fracture on colloids.



**Figure 4-5.** Comparison between FVFARF and Collage II Plus for case with rapid sorption-desorption from mobile colloids ( $\kappa_1=1 \text{ y}^{-1}$ ) and slow sorption-desorption from immobile colloids ( $\kappa_2=10^{-6} \text{ y}^{-1}$ ). Radionuclides enter the fracture in solute phase.

### 4.3 Colloid facilitated migration in a KBS-3 repository

A study of the effects of colloid facilitated migration in a KBS-3 repository has been made using the Collage II Plus program and the results are documented in Chapter 4 of /Klos et al. 2002/. These simulations are based on the parameters used in the SR 97 far-field simulations /SKB 1999/, but colloids have been added to the model.

To validate the FVFARF code we have tried to repeat some of the simulations of the above mentioned report. However, the simulations reported in /Klos et al. 2002/ are based on some misconceptions of the FARF31 model which must be analysed. Table 4-2 shows the input parameters used by SKB and those used by /Klos et al. 2002/.

In the simulations all colloids have been assumed to be fully mobile, i.e. no filtering occurs. All radionuclides enter the flow path in the solute phase. The downstream boundary condition has been applied right at the end of the flow path. We have concentrated on the case which is described in /Klos et al. 2002/ as a worst case. This is a case motivated by future non-saline groundwater conditions giving a colloid partition coefficient ( $kI$ ) of  $4.5 \cdot 10^{-3}$  and a parameter scan of the sorption-desorption rate in order to find the worst case. The results from the simulations reported in /Klos et al. 2002/ are plotted with blue colour in figure 4.8 (p. 38) in this report.

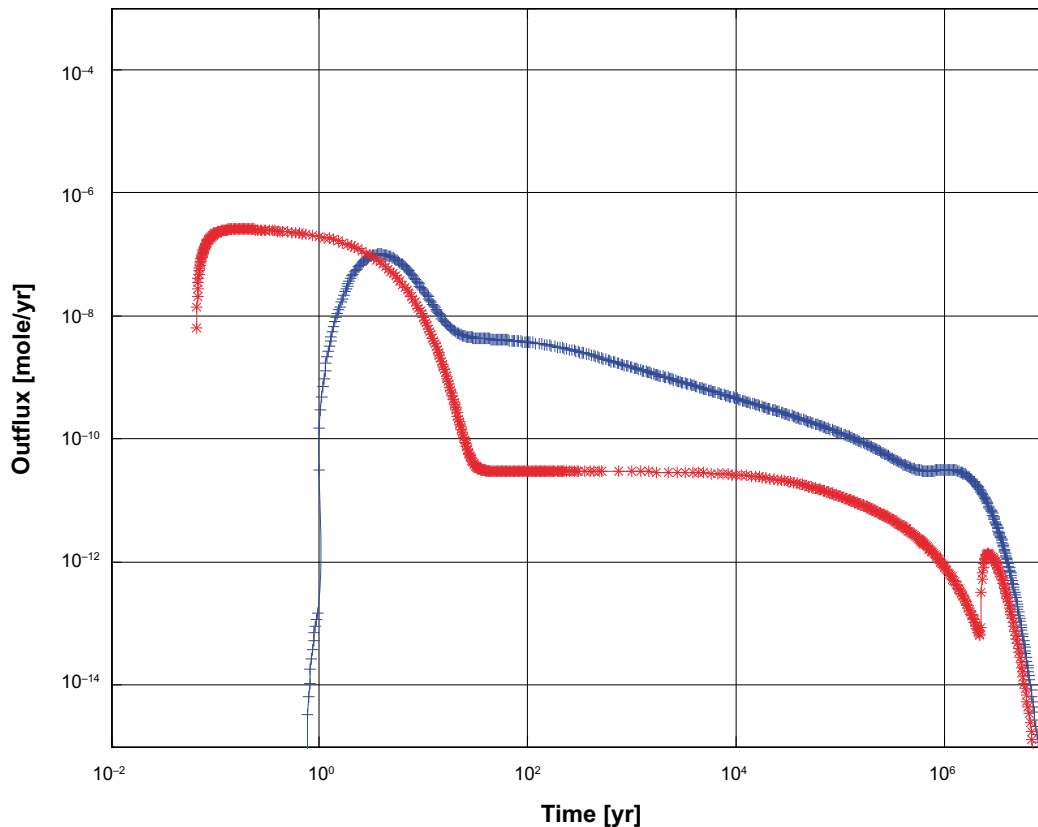
**Table 4-2. Input parameters used in SR 97 and in /Klos et al. 2002/, respectively.**

Parameter	SR 97	/Klos et al. 2002/	Comment
Residence time (y)	10	16.7	
Peclet number (-)	10	$5.9 \cdot 10^{-2}$	
Matrix porosity (-)	0.005	0.005	Possible printing error
Penetration depth (m)	2	N/A	
Distribution coeff. ( $\text{m}^3 \text{kg}^{-1}$ )	5	5	
Matrix diffus. coeff. ( $\text{m}^2 \text{y}^{-1}$ )	$1.2614 \cdot 10^{-6}$	$1.262 \cdot 10^{-6}$	$= 4 \cdot 10^{-14} (\text{m}^2 \text{s}^{-1})$
Flow wetted area ( $\text{m}^{-1}$ )	$1 \cdot 10^4$	N/A	ASPEC
Decay constant ( $\text{y}^{-1}$ )	$1.8568 \cdot 10^{-6}$	$1.86 \cdot 10^{-6}$	Half life: 373 300 (y)
Solute velocity ( $\text{m y}^{-1}$ )	N/A	30	
Solute dispersion coeff. ( $\text{m}^2 \text{y}^{-1}$ )	N/A	1.695	
Length of flow path (m)	N/A	500	
Fracture half width (m)	N/A	$1 \cdot 10^{-4}$	Inverse of ASPEC
Colloid velocity multiplier (-)	N/A	1.32	
Colloid dispersion multiplier (-)	N/A	2.8	
Colloid partition coefficient (-)	N/A	$4.5 \cdot 10^{-3}$	Worst case
Sorption-desorption rate ( $\text{y}^{-1}$ )	N/A	$0.8 \cdot 10^{-1}$	Worst case

Choosing the parameter values documented in /Klos et al. 2002/ will give a residence time of 17.7 years and a Peclet number of  $5.9 \cdot 10^{-2}$  according to the definition used in FARF31, contrary to what is stated in the report. This means that the advective term will be 170 times weaker compared to the dispersive term in the simulation documented in /Klos et al. 2002/ than it was in the SR 97 simulation.

Another difference is that Collage II Plus uses a different model than FARF31 for the retardation coefficient  $R$ . The value used in FARF31 would be  $1.35 \cdot 10^4$  compared to the value  $2.69 \cdot 10^6$ . To compensate for this difference, the matrix diffusion coefficient must be multiplied by a factor 199. We have chosen to set the flow wetted area to be the inverse of the fracture half width, which must also be multiplied by the same factor, see A3.6 for further details.

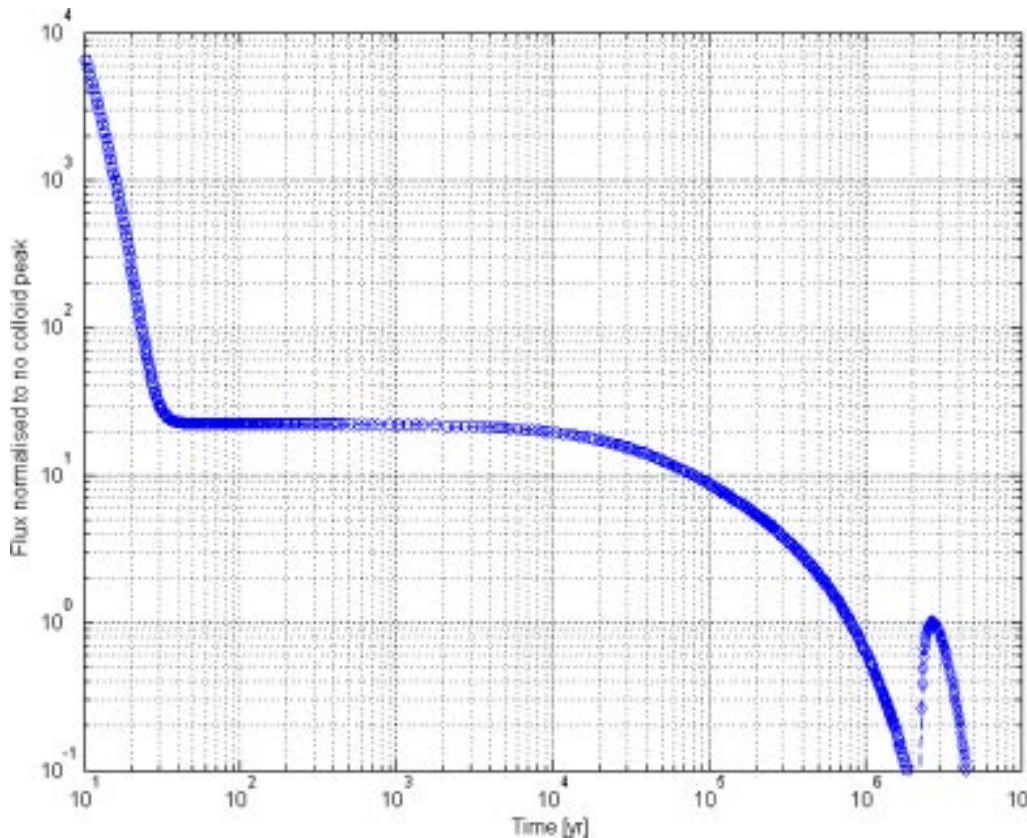
Figure 4-6 below shows the result from the comparison between a FVFARF simulation with the parameter values used in /Klos et al. 2002/ and one with the correct parameter values from SR 97. The same values of the colloid parameters have been used in both cases. Red colour and star markers (\*) indicates values from /Klos et al. 2002/, and is the sum of the colloid and solute release rate. Blue colour and plus markers (+) indicates correct SR 97 parameter values.



**Figure 4-6.** Comparison between total outfluxes based on the parameter values used in /Klos et al. 2002/ and the parameter values used in SR 97. FVFARF has been used in both simulations. Red colour and star markers denote parameter values used by Klos et al. Blue colour and plus markers denote SKB parameter values.

The outflux normalised to the no colloid peak for the worst case simulation is shown in Figure 4-7. The same scaling and axis limits have been used as in Figure 4.8 of /Klos et al. 2002/ to facilitate a comparison, but note that FVFARF is able to resolve the colloid peak also under these extreme conditions. Rings are colloids and diamonds are solute.

As Figure 4-7 shows, the results are similar but not identical. One remaining difference between the FVFARF and the Collage II Plus programs is that FVFARF uses extrapolation of concentration as boundary condition at the downstream boundary, while Collage II Plus sets the concentration to zero. Setting the concentration to zero is probably a better assumption at such a low Peclet number, and this model difference probably accounts for the remaining differences in results between the Collage II Plus and the FVFARF programs.



*Figure 4-7. Outflux of solute and colloid phases normalised to the no-colloids peak. Ring markers denote colloid phase and diamond markers denote solute phase.*

#### 4.4 A test case for the tunnel model

A test case (called test case 7) suitable for testing FARF32 has been constructed. It contains one fictitious nuclide, and the following input data are used:

Stream tube length (m)	200
Darcy velocity (m/y)	1
Dispersion coefficient (m <sup>2</sup> /y)	0.001
Porosity (1)	1
Sorption coefficient (m <sup>3</sup> /kg)	1
Half life (y)	1·10 <sup>10</sup>

The input function is a square pulse of unit height and a length of 400 years. The input data correspond to a travel time of 200 years and a Peclet number of  $2 \cdot 10^5$ , so the solution should be a unit pulse which is 200 years delayed and somewhat diffused at the edges. The result from a run with FARF32 standalone is showed in Figure 4-8.

Running FARF33 on corresponding input data gives the same result as can be easily verified.

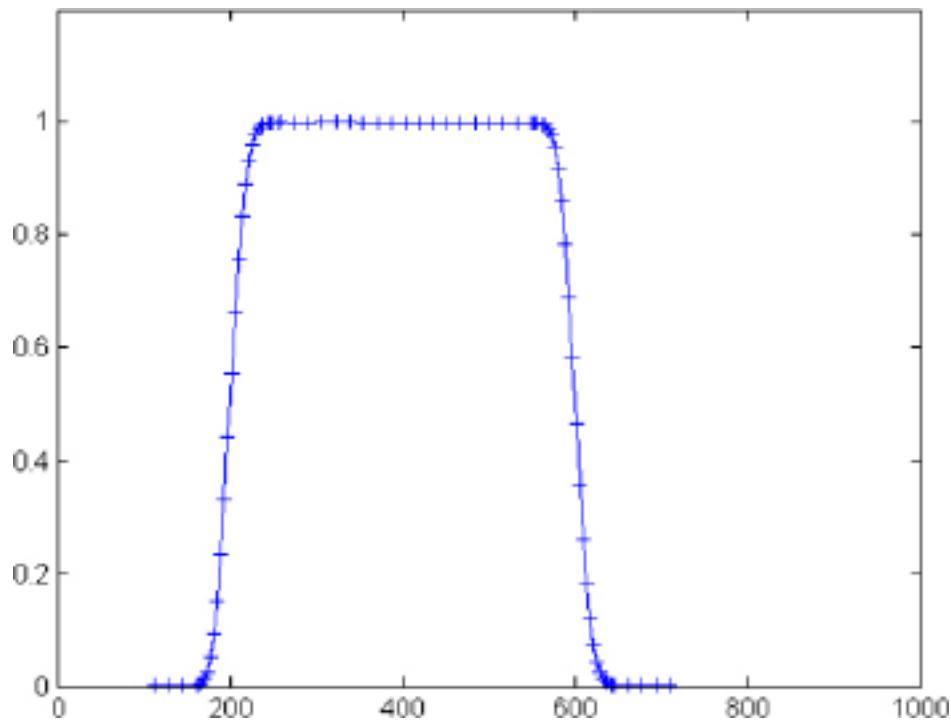


Figure 4-8. Solution to test case 7 as calculated by FARF32 standalone on PC.

#### 4.5 A test case for FARF33

A test case (called test case 8) suitable for testing FARF33 has also been constructed. Input data are the same as in example 7 with the following additions:

Apect ratio ( $m^{-1}$ )	10
Penetration depth (m)	1
Diffusivity ( $m^2/y$ )	$1 \cdot 10^{-6}$

The same input function gives a completely different solution, as shown in Figure 4-9.

#### 4.6 A test case for testing colloids

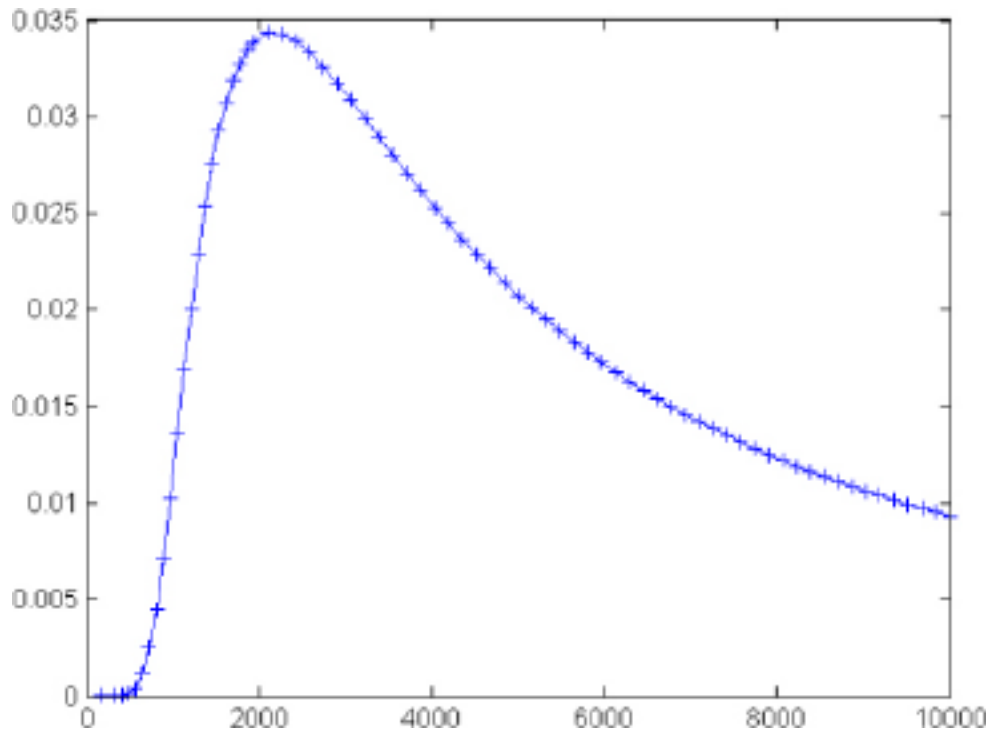
Since the transport equations for both solute and colloid phases are linear, a superposition of solutions is also a solution. This can be used to construct a test case for colloids (called test case 9) based on test cases 7 and 8. Input data are the same as in example 8 with the following additional colloid parameters:

Solute injection fraction (ETA) (1)	0.95
Ratio of immobile colloids (BETA) (1)	0
Partition coefficients (1)	0
Rate of sorption-desorption ( $y^{-1}$ )	0

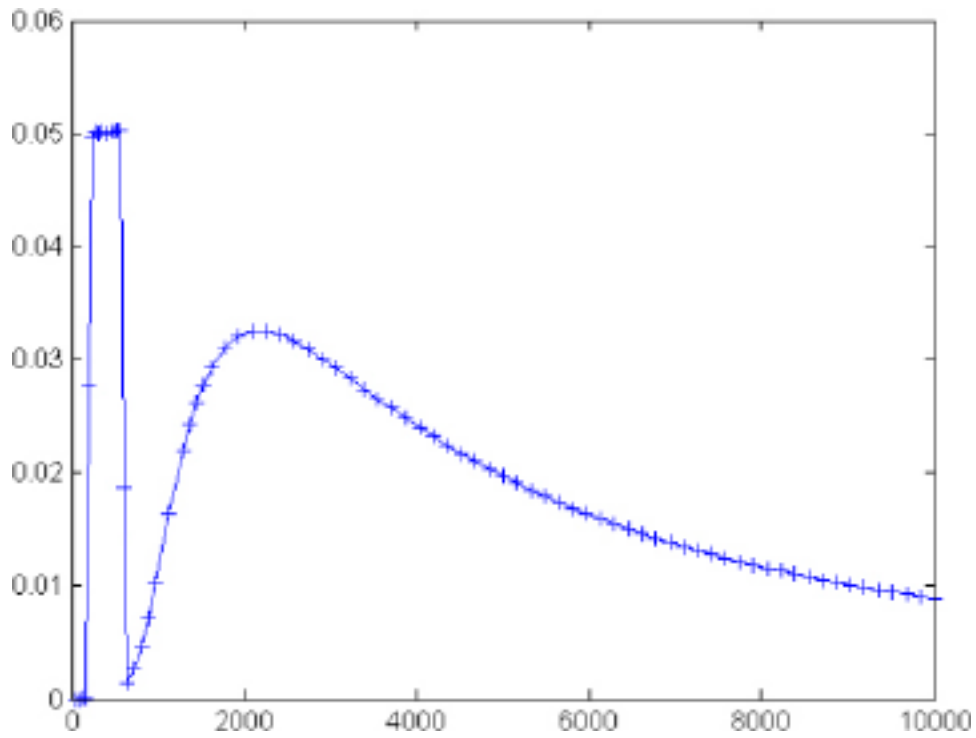
This means that 95% of the radionuclide enters the domain in the solute phase and the other 5% sorbed to colloids, and that no transfer between the phases can occur. The solution to this problem is shown in Figure 4-10.

The solution shown in Figure 4-10 is a superposition of the solutions to test cases 7 and 8.





*Figure 4-9. Solution to test case 8 as calculated by FARF33 standalone on PC.*



*Figure 4-10. Solution to test case 9 as calculated by FARF33 standalone on PC.*

## 5 Conclusions and suggestions

Colloid-facilitated radionuclide transport in fractured rock, including retention and chain decay, can be simulated using the developed MATLAB program. The spatial resolution of the computational grid must be of the order 100 by 100 to replicate results obtained by other simulation codes based on Laplace transforms. For a decay chain including five nuclides, this gives a system matrix of rank 50,000 by 50,000. The system matrix becomes very sparse and block-diagonally banded due to the difference operators used. It is fully feasible to solve a problem of this size on an ordinary PC, if computation times in the order of 10 minutes are accepted. Instead it is the required memory which puts a limit on the problem size.

It is important that the computational grid is large enough in the transversal direction in order to obtain accurate results. Based on a few experiments, we suggest a rule of thumb for the necessary size of the computational domain as:

$$\frac{L_{\max}^2}{D_{i,\max} T_{\text{stop}}} \geq 100$$

where  $L_{\max}$  is the distance to the outer boundary,  $T_{\text{stop}}$  is the end time of the simulation, and  $D_{i,\max}$  is the maximum diffusion coefficient for any nuclide in the simulation. At the same time it is important that the grid is not too coarse near the stream tube, to avoid underestimating the transfer of concentration from the solute phase to the pore water. This can be a problem if the diffusion coefficients vary strongly between the nuclides. One remedy could be to use different grid spacing in the transversal direction for different nuclides and use interpolation to evaluate the source term coming from radioactive decay. Another remedy could be to use a higher order discretisation scheme in the transversal direction.

## 6 References

**Cvetkovic V, 2003.** Potential impact of colloids on plutonium migration at the Äspö site, Royal Institute of Technology (KTH). SKB R-03-16, Svensk Kärnbränslehantering AB.

**Cvetkovic V, 2004.** Colloid-facilitated radionuclide transport in single fractures: Sensitivities and scoping calculations for tracer tests at Äspö, KTH, Division of Water Resources Engineering. SKB R-04-49, Svensk Kärnbränslehantering AB.

**Hicks T W, 2003.** COLLAGE II Plus User's Guide, Version 2, Galson Sciences Ltd.

**Klos R A, White M J, Wickham S M, Bennett D G, Hicks T W, 2002.** Quantitative Assessment of the Potential Significance of Colloids to the KBS-3 Disposal Concept, SKI Report 02:34, ISSN 1104-1374.

**Lindgren M, Gylling B, Elert M, 2002.** FARF31 Version 1.3 User's Guide. SKB Arbetsrapport TS-02-03, Svensk Kärnbränslehantering AB.

**Norman S, Kjellbert N, 1990.** FARF31 – A far field radionuclide migration code for use with the PROPER package. SKB Technical Report 90-01, Svensk Kärnbränslehantering AB.

**SKB, 1999.** SR 97 – Deep repository for spent nuclear fuel. SR 97 – Post-closure safety. Main report – Vol. I, Vol. II and Summary. SKB TR-99-06, Svensk Kärnbränslehantering AB.

## A1 A user's guide for the programs

### A1.1 Using the FVFARF program

The MATLAB program FVFARF consists of the following thirteen m-files:

**Table A1-1. A list of the FVFARF m-files.**

File name	Comment
FVFARF_create_mesh.m	
FVFARF_func.m	The ODE-function
FVFARF.m	The main function
FVFARF_input.m	
FVFARF_interpolate_ts.m	
FVFARF_output.m	
FVFARF_read.m	
FVFARF_read_colpar.m	
FVFARF_read_par.m	
FVFARF_read_ts.m	
FVFARF_set_par.m	
FVFARF_setup.m	
FVFARF_setup_system.m	

All m-files are listed in Appendix 2. They must reside somewhere in the MATLAB-path in order to use the program.

Four input files are needed to supply the program with input data. They should all be saved in the same directory and named in.dat, in.par, in.ts and in.col. The first three have the same syntax as the corresponding files for the stand alone version of FARF31 /Lindgren et al. 2002/. The fourth one, in.col, contains additional parameters needed to describe colloid behaviour and some numerical parameters. Comment lines and blank lines are allowed in all input files. Comment lines begin with a hashmark character (#).

**Table A1-2. A list of the FVFARF input files.**

in.dat	contains mainly nuclide data
in.par	contains mainly transport and physical parameters
in.ts	contains the input time series
in.col	contains colloid and numerical parameters

The program is started by calling the main function from the MATLAB-prompt with a directory name as argument. All input files should reside in the directory used as argument. For example:

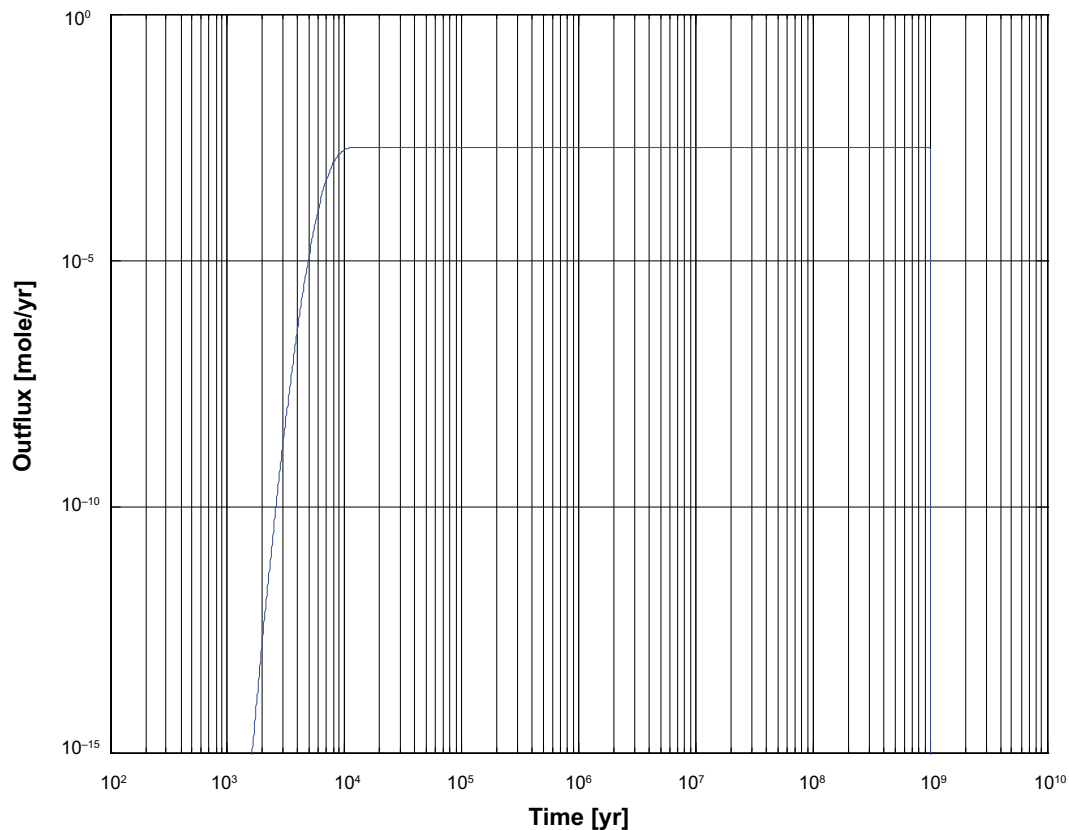
```
>> [T,C]=FVFARF('C:\SKB\Kolloider\Harald\Problem2');
Calling ode15s!
1485 successful steps
35 failed attempts
2591 function evaluations
2 partial derivatives
256 LU decompositions
2582 solutions of linear systems

elapsed_time =

    8.9430

Warning: Negative data ignored.
> In C:\MATLAB6p5p1\toolbox\matlab\graph2d\ylabel.m at line 27
   In C:\SKB\Kolloider\Harald\FVFARF.m at line 200
Warning: Negative data ignored.
> In C:\MATLAB6p5p1\toolbox\matlab\graph2d\xlabel.m at line 27
   In C:\SKB\Kolloider\Harald\FVFARF.m at line 201
>>
```

If output arguments are given ([T,C] in the example above), they will contain the output time series on return. Some diagnostic messages are printed as can be seen in the example above. The output time series are plotted in a log-log diagram in a separate window, which may look as follows:



If the simulation contains more than one nuclide, the corresponding output time series will be plotted using different colours.

### A1.1.1 The input file in.dat

The data specified is first some general data for the simulation assigned by keywords followed by the definitions of the radionuclides and chains. The following keywords and values are allowed:

Keyword	Value	Description
PRINT	ON OFF DEBUG	Not used
CASENAME	<i>casename</i>	Not used
DIFFUSIVITY	SINGLE ELEMENT_SPECIFIC	Variable used to control if one single diffusion coefficient is used or if chemical element specific diffusion coefficients are used.

The second part of the file includes the definition of the radionuclides and chains. One line for each nuclide has to be specified:

Variable	Value type	Unit	Description
CHNAM	String	(-)	Name of nuclide.
THALF	Real	(a)	Half-life of nuclide.
IDAUGH	Integer	(-)	Set to 1 if nuclide has a daughter, and to 0 otherwise.
ISOURC	Integer	(-)	Set to 1 if there is an input time series for this nuclide, and to 0 otherwise.

If a nuclide has a daughter nuclide, IDAUGH should be set to 1, and the definition of that daughter nuclide has to follow immediately after the mother nuclide.

Here is an example of the input file in.dat:

```
#
# Problem 1 for Farf31 stand-alone 1.2
# File: in.dat          01/09/21
#
# Print option
PRINT ON
# Effective diffusivity
DIFFUSIVITY ELEMENT_SPECIFIC
#
CASENAME farf
# Nuclide Thalf      Daughter Source
Pu242      379000      1      1
U238      4.51E+09      1      1
U234      247000      1      1
Th230      80000      1      1
Ra226      1600      0      1
#
# Daughter = 1
# The following nuclide is daughter
#
# Source = 1
# Has a source curve as input
#
```

### A1.1.2 The input file in.par

The following is a list of the input parameters read from the file in.par. The parameter values are assigned with keywords and can be specified in arbitrary order.

Keyword	Unit	Description
TW	(a)	Groundwater travel time.
PECLET	(-)	Longitudinal dispersion Peclet number.
ASPEC	(1/m)	Surface sorption area per unit of liquid volume.
F	(a/m)	=TW*ASPEC. Optional parameter to be specified instead of TW or ASPEC.
EPS	(-)	Matrix porosity.
DE	(m <sup>2</sup> /a)	Effective matrix diffusion coefficient. Only given if DIFFUSIVITY is defined as SINGLE in the file in.dat.
PENDEP	(m)	Maximum depth of diffusive penetration in the transversal direction.
KDR_XX	(m <sup>3</sup> /kg)	Distribution (sorption) coefficient for chemical element XX (not nuclide).
DE_XX	(m <sup>2</sup> /a)	Effective matrix diffusion coefficient for chemical element XX (not nuclide). Only given if DIFFUSIVITY is defined as ELEMENT_SPECIFIC in the file in.dat.

For the last two keywords, XX denotes the first two letters of the name of the chemical element. One line should be given for each chemical element in the simulation. Note that for a chemical element with only one letter also the first digit must be given, for example U2 for uranium.

Here is an example of the input file in.par:

```
#
# Problem 1 for Farf31 stand-alone 1.2
# File: in.par          01/09/21
#
# New optional parameter F=TW*ASPEC

F      1.e5
# Groundwater travel time
TW     10000.
# Peclet number
PECLET 4000.0
# Specific surface
#ASPEC 1000.
# Diffusion porosity of rock
EPS    0.005
# Depth of penetration
PENDEP 2.0
# Chemical element information:
# Kd value(s)
KDR_Pu 1.
KDR_U2 1.
KDR_Th 1.
KDR_Ra 1.
# Diffusion coefficient (s)
DE_Pu 3.e-6
DE_U2 3.e-6
DE_Th 3.e-6
DE_Ra 3.e-6
```

### A1.1.3 The input file in.ts

The input file in.ts contains the definitions of the input time series as time-value-pairs. The series must be given in the same order as in the definition of nuclides in the file in.dat. Each time series starts with the name of the nuclide on the first line, and thereafter the time series given as pairs of time and input value.

Here is an example of the input file in.ts:

```
Pu242
1.0E2 1.0E-0
1.0E9 1.0E-0
U238
1.0E2 1.0E-0
1.0E9 1.0E-0
U234
1.0E2 1.0E-0
1.0E9 1.0E-0
Th230
1.0E2 1.0E-0
1.0E9 1.0E-0
Ra226
1.0E2 1.0E-0
1.0E9 1.0E-0
```

### A1.1.4 The input file in.col

The input file in.col contains some additional parameters needed for the colloid model and some numerical parameters. The following is a list of the input parameters read from the file in.par. The parameter values are assigned with keywords and can be specified in arbitrary order.

Keyword	Unit	Description
TSTART	(a)	Start time for the simulation.
TSTOP	(a)	Stop time for the simulation.
ABSTOL	(-)	Absolute tolerance for the time integration.
RELTOL	(-)	Relative tolerance for the time integration.
NI	(-)	Number of grid points in the longitudinal (streamwise) direction.
NJ	(-)	Number of grid points in the transversal direction.
STRETCHING	(-)	Parameter for the stretching of the grid in the transversal direction.
PHI	(-)	Rock matrix porosity value for the COLLAGE retention model. Only used if FARFPOROSITY is set to 0.
ETA	(-)	Fraction of injected inventory in solute phase.
BETA	(-)	Fraction of immobile colloids.
K1	(-)	Partition coefficient for radionuclides sorbing onto mobile colloids.
K2	(-)	Partition coefficient for radionuclides sorbing onto immobile colloids.
KAPPA1	1/a	Rate of sorption-desorption of mobile colloids.
KAPPA2	1/a	Rate of sorption-desorption of immobile colloids.
FARFPOROSITY	(-)	Set this parameter to 0 if the COLLAGE retention model should be used, and to 1 if the FARF retention model should be used.



Here is an example of the input file in.col:

```
#
# Problem 1 for FVFARF stand-alone 1.2
# File: in.col          04/01/07
#

# Start time for time integration
TSTART 100.
# Stop time for time integration
TSTOP 9.9e9
# Absolute tolerance for time integration
AbsTol 1e-12
# Relative tolerance for time integration
RelTol 1e-6
# Number of points in streamwise direction
NI 30
# Number of points in transversal direction
NJ 5
# Transverse stretching parameter
STRETCHING 1.33333333333333
# Rock matrix porosity
phi 0.0
# Solute injection fraction
eta 1.0
# Immobile colloid fraction
beta 0.0
# Colloid transfer parameters
k1 0.0
k2 0.0
kappal 0.0
kappa2 0.0
# If FARFporosity is set to 1 the porosity model of FARF is used
# If FARFporosity is set to 0 the porosity model of COLLAGE is used
FARFporosity 1.0
```

## A1.2 Using the new PROPER submodels FARF32 and FARF33

Some adjustments were made to the FVFARF program to make it fit into the PROPER simulation environment as two new submodels, FARF32 and FARF33. FARF32 is intended for the tunnel case, i.e. a situation where the radionuclide transport occurs in a one-dimensional tunnel in the rock matrix without diffusion into the solid rock. FARF33 contains all functionality of FARF31 and adds on colloid transport. Both submodels exist both as standalone programs and as submodels linked into the PROPER monitor.

### A1.2.1 The parameter file for FARF32

FARF32 takes the following input data as stochastic parameters:

Keyword	Unit	Description
L	(m)	Length of flow path.
U_XX	(m/a)	Darcy velocity for element XX (not nuclide) in tunnel.
DL_XX	(m <sup>2</sup> /a)	Longitudinal dispersion coefficient for chemical element XX (not nuclide) in tunnel.
EPS	(-)	Porosity of tunnel filling.
KDR_XX	(m <sup>3</sup> /kg)	Distribution (sorption) coefficient for element XX (not nuclide) in tunnel filling.

The standard mass density set in the prm-file is used in the calculation of the retention factor. In the version of the submodel linked into PROPER, these stochastic parameters are given in the same order as listed in the table.

### A1.2.2 The parameter file for FARF33

In FARF33 we have kept the traditional input parameters of FARF31 and added a few new stochastic parameters for the colloid transport. The new parameters are:

Keyword	Unit	Description
ETA	(-)	Solute injection fraction.
BETA	(-)	Fraction of immobile colloids.
K1	(-)	Partition coefficient for radionuclides sorbing onto mobile colloids.
K2	(-)	Partition coefficient for radionuclides sorbing onto immobile colloids.
KAPPA1	(a <sup>-1</sup> )	Rate of sorption-desorption of mobile colloids.
KAPPA2	(a <sup>-1</sup> )	Rate of sorption-desorption of immobile colloids.

In the version of the submodel linked into PROPER, these stochastic parameters are given after all other parameters in the same order as listed in the table.

### A1.2.3 Additions to the prm-file

In order to control the time steps and tolerance of the ODEPACK solver and some other features of the solution scheme, a number of new keywords were defined as numerically related parameters in the prm-file:

Keyword	Unit	Description
TSTART	(a)	Start time for the solution.
TSTOP	(a)	Stop time for the solution.
TSTEP	(a)	Initial time step.
STPMIN	(a)	Minimum time step.
STPMAX	(a)	Maximum time step.
RELTOL	(-)	Relative tolerance.
ABSTOL	(mole/m <sup>3</sup> )	Absolute tolerance.
NI	(-)	Number of cells in the streamwise direction.
NJ	(-)	Number of cells in the transversal direction.
STRETCHING	(-)	Stretching factor for the mesh in the transversal direction.
LUDS	on/off	Use UDS if the keyword is present and CDS otherwise.
LFREE	on/off	Use the free downstream boundary condition if the keyword is present and the homogeneous one otherwise.
ICOLL	0 or 1	One means that colloids should be present in the simulation and zero means that there should be no colloids in the simulation.

Most of these parameters have default values.

## A2 Code listing

### A2.1 FVFARF.m

```

function [T,out]=FVFARF(dirnam)
%FVFARF This is the main function for the finite volume FARF program
%
% FVFARF(dirnam) Main function for the finite volume FARF program. It
% takes a directory name as argument and reads the input files from that
% directory.
%
%
% Create file names
%
infile=fullfile(dirnam,'in.dat');
parfile=fullfile(dirnam,'in.par');
tsfile=fullfile(dirnam,'in.ts');
colfile=fullfile(dirnam,'in.col');

%
% Read all input data files and create nuclides
%
[nuclides,single]=FVFARF_read(infile);
[parameters,elements]=FVFARF_read_par(parfile);
colpar=FVFARF_read_colpar(colfile);
nuclides=FVFARF_set_par(nuclides,single,parameters,elements);
nuclides=FVFARF_read_ts(nuclides,tsfile);
NK=size(nuclides,1);
ix=find(strncmpi(parameters(:,1),'PECLET',6));
if isempty(ix)
    error('Peclet number missing in input data.');
```

```

end
pe=parameters{ix,2};
ix=find(strncmpi(parameters(:,1),'TW',2));
ix1=find(strncmpi(parameters(:,1),'ASPEC',5));
ix2=find(strncmpi(parameters(:,1),'F',1));
if ~isempty(ix) & ~isempty(ix1) & ~isempty(ix2)
    error('Too many parameters in input.');
```

```

end
if isempty(ix2)
    tw=parameters{ix,2};
    aw=parameters{ix1,2};
elseif isempty(ix1)
    tw=parameters{ix,2};
    f=parameters{ix2,2};
    aw=f/tw;
else
    aw=parameters{ix1,2};
    f=parameters{ix2,2};
    tw=f/aw;
end
ix=find(strncmpi(parameters(:,1),'EPS',3));
if isempty(ix)
    error('Matrix porosity missing in input data.');
```

```

end
eps=parameters{ix,2};

%
```

```

% Create mesh
%
ix=find(strncmp(colpar(:,1),'NI',2));
if isempty(ix)
    error('Mesh size (NI) missing in input data.');
```

```

end
NI=colpar{ix,2};
ix=find(strncmp(colpar(:,1),'NJ',2));
if isempty(ix)
    error('Mesh size (NJ) missing in input data.');
```

```

end
NJ=colpar{ix,2};
% Set L to an arbitrary number, e.g. 1.0
L=1.0;
ix=find(strncmp(parameters(:,1),'PENDEP',6));
if isempty(ix)
    error('Penetration depth missing in input data.');
```

```

end
XMAX=parameters{ix,2};
ix=find(strncmp(colpar(:,1),'STRETCHING',10));
if isempty(ix)
    error('Mesh stretching factor (STRETCHING) missing in input data.');
```

```

end
re=colpar{ix,2};
[dz,dx]=FVFARF_create_mesh(NI,NJ,L,XMAX,re);

%
% Setup the system matrix for the ODE. Read COLLAGE parameters first.
%
us=L/tw;
Ds=us^2*tw/pe;
b=1/aw;
ix=find(strncmp(colpar(:,1),'PHI',3));
if isempty(ix)
    error('Rock matrix porosity (PHI) missing in input data.');
```

```

end
phi=colpar{ix,2};
% Set velocity and diffusivity equal for solute and colloid fraction
um=us;
Dm=Ds;
ix=find(strncmp(colpar(:,1),'ETA',3));
if isempty(ix)
    error('Solute injection fraction (ETA) missing in input data.');
```

```

end
eta=colpar{ix,2};
ix=find(strncmp(colpar(:,1),'KAPPA1',6));
if isempty(ix)
    error('Transfer coefficient (KAPPA1) missing in input data.');
```

```

end
kappal=colpar{ix,2};
ix=find(strncmp(colpar(:,1),'KAPPA2',6));
if isempty(ix)
    error('Transfer coefficient (KAPPA2) missing in input data.');
```

```

end
kappa2=colpar{ix,2};
ix=find(strncmp(colpar(:,1),'K1',2));
if isempty(ix)
    error('Transfer coefficient (K1) missing in input data.');
```

```

end
k1=colpar{ix,2};
ix=find(strncmp(colpar(:,1),'K2',2));
if isempty(ix)
    error('Transfer coefficient (K2) missing in input data.');
```

```

end
k2=colpar{ix,2};
ix=find(strncmpi(colpar(:,1),'BETA',4));
if isempty(ix)
    error('Immobility fraction (BETA) missing in input data.');
```

```

end
beta=colpar{ix,2};
ix=find(strncmpi(colpar(:,1),'FARFPOROSITY',12));
if isempty(ix)
    error('Porosity model missing in input data.');
```

```

end
FARFPorosity=colpar{ix,2};
[A,B,LScheme]=FVFARF_setup_system(nuclides,NI,NJ,dz,dx,us,um,Ds,Dm,phi,b,eps
,kappal,kappa2,k1,k2,beta,eta,FARFPorosity);

%
% Allocate space for the solution and setup the initial condition
%
C=zeros(NI*NJ*NK,1);
y0=zeros(NI*NJ*NK,1);

%
% Set a sparsity pattern for the Jacobian
%
SJ = A~=0;
options = odeset('JPattern',SJ);

%
% Set statistics on
%
options = odeset(options,'Stats','on');
```

```

%
% Set RelTol
%
ix=find(strncmpi(colpar(:,1),'RELTOL',6));
if isempty(ix)
    reltol=1e-6;
else
    reltol=colpar{ix,2};
end
options = odeset(options,'RelTol',reltol);

%
% Set AbsTol
%
ix=find(strncmpi(colpar(:,1),'ABSTOL',6));
if isempty(ix)
    abstol=1e-12;
else
    abstol=colpar{ix,2};
end
options = odeset(options,'AbsTol',abstol);

%
% Solve the ODE using ode15s
%
ix=find(strncmpi(colpar(:,1),'TSTART',6));
if isempty(ix)
    TSTART=0;
else
    TSTART=colpar{ix,2};
end
ix=find(strncmpi(colpar(:,1),'TSTOP',5));
```

```

if isempty(ix)
    error('Stop time (TSTOP) missing in input data.');
```

```

else
    TSTOP=colpar{ix,2};
end
disp('Calling ode15s!');
tic; [T,C]=ode15s(@FVFARF_func,[TSTART TSTOP],y0,options,A,B,nuclides);toc;

%
% Calculate the output function for each nuclide and plot it
%
colors={'b','g','r','c','m','y','k'};
NT=size(T,1);
for K=1:NK
    out=FVFARF_output(LSCHEME,NT,NI,NJ,K,C,dz,us,um,Ds,Dm,b);
    loglog(T,out,colors{K});
    hold on
end
ylabel('Outflux [mole/yr]')
xlabel('Time [yr]')
grid on
axis([ 1e2 1e10 1e-15 1e0])

```

## A2.2 FVFARF\_create\_mesh.m

```

function [dz,dx]=FVFARF_create_mesh(NI,NJ,L,XMAX,re)
%FVFARF_create_mesh Discretize the computational domain for FVFARF
%
% FVFARF_create_mesh(NI,NJ) creates a mesh with NJ points in the transverse
% direction and NI points in the longitudinal direction. The physical size
% is
% L in the longitudinal direction and XMAX in the transverse direction. The
% stretching factor re is used in the transverse direction.
%
dz=L/NI;

dx(1)=0;
dx(2)=0;
dx(3)=1;
for J=4:NJ
    dx(J)=dx(J-1)*re;
end
dx=dx/sum(dx)*XMAX;

```

## A2.3 FVFARF\_func.m

```

function dy=FVFARF_func(t,y,A,B,nuclides)
dy=A*y+FVFARF_input(t,B,nuclides);

```

## A2.4 FVFARF\_input.m

```

function term=FVFARF_input(t,B,nuclides)
%FVFARF_input Calculates the constant term of the FVFARF system function
%
% FVFARF_input(t,B,nuclides) calculates the constant term of the FVFARF
% system function based on the time t, the RHS vector B and a cell array of
% nuclides which contains the input time series definition for all
% nuclides.
%
```

```

%
% Initialize output
%
term=zeros(size(B));

%
% Get information about which nuclides have input functions
%
isource=[nuclides(:,4)];

%
% Get time series values for all nuclides
%
NK=size(nuclides,1);
NINJ=size(B,1)/NK;
ix=1;
for i=1:NK
    value=isource(i)*FVFARF_interpolate_ts(t,nuclides{i,7});
    term(ix)=term(ix)+B(ix)*value;
    term(ix+1)=term(ix+1)+B(ix+1)*value;
    ix=ix+NINJ;
end

```

## A2.5 FVFARF\_interpolate\_ts.m

```

function y=FVFARF_interpolate_ts(t,ts)
%FVFARF_interpolate_ts Evaluate a timeseries at a specific time
%
% FVFARF_interpolate_ts(t,ts) evaluates the timeseries ts at time t using
% linear interpolation. If t is outside the definition of ts, zero is
% returned.
%
y=interp1(ts(:,1),ts(:,2),t,'linear',0.0);

%
% Actually, the implementation below is clearly faster, so we keep it here,
% but commented. It relies on that the time series is sorted.
%
% ix=find(ts(:,1)>t);
% if isempty(ix)
%     % This is the case when t is beyond the definition of ts
%     y=0;
% elseif ix(1)==1
%     % This is the case when t is before the definition of ts
%     y=0;
% else
%     % This is the ordinary case
%     ix2=ix(1);
%     ix1=ix2-1;
%     t2=ts(ix2,1);
%     y2=ts(ix2,2);
%     t1=ts(ix1,1);
%     y1=ts(ix1,2);
%     y=y1+(t-t1)/(t2-t1)*(y2-y1);
% end

```

## A2.6 FVFARF\_output.m

```
function out=FVFARF_output(LSCHEME,NT,NI,NJ,K,C,dz,us,um,Ds,Dm,b);
%FVFARF_output Calculate the output function of the FVFARF program
%
% FVFARF_output(LSCHEME,NT,NI,NJ,K,C,dz,us,um,Ds,Dm,b) calculates the
% output function of the FVFARF program by integrating the outflux of a
% nuclide in time at the outlet of the stream tube. The output function is
% the sum of solved nuclide and colloid transportes nuclide at the outlet.
%
%
% Loop over all time points in solution time series
%
for it=1:NT
    %
    % Calculate cell indices at the outlet
    %
    I=NI;
    J=1;
    P=(K-1)*NI*NJ+(I-1)*NJ+J;
    W=P-NJ;
    NW=W+1;
    WW=P-2*NJ;
    NWW=WW+1;
    N=P+1;
    E=P+NJ;
    %
    % Switch over longitudinal discretization schemes
    %
    switch LSCHEME
        case 'CDS'
            OUT_s=2*b*(us*(15/8*C(it,P)-5/4*C(it,W)+3/8*C(it,WW))-
Ds/dz*(2*C(it,P)-3*C(it,W)+C(it,WW)));
            OUT_m=2*b*(um*(15/8*C(it,N)-5/4*C(it,NW)+3/8*C(it,NWW))-
Dm/dz*(2*C(it,N)-3*C(it,NW)+C(it,NWW)));
        case 'UDS'
            OUT_s=2*b*(C(it,P)*us-Ds/dz*(2*C(it,P)-3*C(it,W)+C(it,WW)));
            OUT_m=2*b*(C(it,N)*um-Dm/dz*(2*C(it,N)-3*C(it,NW)+C(it,NWW)));
    end
    %
    % Sum solved and colloid transported concentration
    %
    out(it)=OUT_s+OUT_m;
end
```

## A2.7 FVFARF\_read.m

```
function [nuclides,single] = FVFARF_read(filename)
%FVFARF_read Read an ordinary FARF31 input file (in.dat)
%
% FVFARF_read(filename) reads an ordinary FARF31 data file for the stand
% alone version and returns a cell array of nuclides plus an indicator for
% element specific diffusivity
%
%
% Initialize return values
%
n=0;
nuclides={};
single=0;
%
```



```

% Open input file
%
fid=fopen(filename,'rt');
if fid==-1
    error('Could not open file.');
```

```

end
%
% Read and parse all lines
%
while 1
    line=fgetl(fid);
    if ~ischar(line)
        break;
    end
    [token,rem]=strtok(line);
    %
    % Discard comment lines and empty lines
    %
    if ~(isempty(token)) & ~(token(1)=='#')
        switch upper(token)
            case {'PRINT'}
            case {'CASENAME'}
            case {'DIFFUSIVITY'}
                if findstr(upper(rem),'SINGLE')
                    single=1;
                end
            otherwise
                %
                % Add this nuclide to the array of nuclides
                %
                n=n+1;
                nuclides(n,1)={token};
                %
                % Extract half life of nuclide
                %
                [token,rem]=strtok(rem);
                nuclides(n,2)={str2double(token)};
                %
                % Check if this nuclide has a daughter
                %
                [token,rem]=strtok(rem);
                nuclides(n,3)={str2num(token)};
                %
                % Check if this nuclide has input
                %
                [token,rem]=strtok(rem);
                nuclides(n,4)={str2num(token)};
            end
        end
    end
end
%
% Close input file
%
fclose(fid);
```

## A2.8 FVFARF\_read\_colpar.m

```

function par=FVFARF_read_colpar(filename)
%FVFARF_read_colpar Read a colloid parameter file for FVFARF (in.col)
%
% FVFARF_read_colpar(filename) reads a colloid parameter file for
% FVFARF. It returns a cell array of parameter data.
%
```

```

%
% Initialize return values
%
n=0;
par={};
%
% Open input file
%
fid=fopen(filename,'rt');
if fid==-1
    error('Could not open file.');
```

```

end
%
% Read and parse all lines
%
while 1
    line=fgetl(fid);
    if ~ischar(line)
        break;
    end
    [token,rem]=strtok(line);
    %
    % Discard comment lines and empty lines
    %
    if ~(isempty(token)) & ~(token(1)=='#')
        switch upper(token)
            case {'TSTART' 'TSTOP' 'ABSTOL' 'RELTOL' 'NI' 'NJ' 'STRETCHING'
'PHI' 'ETA' 'BETA' 'K1' 'K2' 'KAPPA1' 'KAPPA2' 'FARFPOROSITY'}
                %
                % Add this parameter to the array of parameters
                %
                n=n+1;
                par(n,1)={token};
                %
                % Extract parameter value
                %
                [token,rem]=strtok(rem);
                par(n,2)={str2double(token)};
            otherwise
                error('Unknown keyword found in colloid parameter file.
(%s)', upper(token));
        end
    end
end
end
%
% Close input file
%
fclose(fid);

```

## A2.9 FVFARF\_read\_par.m

```

function [par,el]=FVFARF_read_par(filename)
%FVFARF_read_par Read an ordinary FARF31 parameter file (in.par)
%
% FCFARF_read_par(filename) reads an ordinary FARF31 parameter file for
% the stand alone version. It returns a cell array of parameter data.
%
%
%
% Initialize return values
%
n=0;

```

```

par={};
ne=0;
el={};
%
% Open input file
%
fid=fopen(filename,'rt');
if fid==-1
    error('Could not open file.');
```

```

end
%
% Read and parse all lines
%
while 1
    line=fgetl(fid);
    if ~ischar(line)
        break;
    end
    [token,rem]=strtok(line);
    %
    % Discard comment lines and empty lines
    %
    if ~(isempty(token)) & ~(token(1)=='#')
        switch upper(token)
            case {'F' 'TW' 'PECLET' 'ASPEC' 'EPS' 'DE' 'PENDEP'}
                %
                % Add this parameter to the array of parameters
                %
                n=n+1;
                par(n,1)={token};
                %
                % Extract parameter value
                %
                [token,rem]=strtok(rem);
                par(n,2)={str2double(token)};
            otherwise
                if strncmpi(token,'KDR_',4)
                    elnam=token(5:end);
                    if ne>0
                        ix=find(strncmpi(el(:,1),elnam,2));
                        if isempty(ix)
                            ne=ne+1;
                            ix=ne;
                        end
                    else
                        ne=ne+1;
                        ix=ne;
                    end
                    el(ix,1)={elnam};
                    el(ix,2)={str2double(rem)};
                elseif strncmpi(token,'DE_',3)
                    elnam=token(4:end);
                    if ne>0
                        ix=find(strncmpi(el(:,1),elnam,2));
                        if isempty(ix)
                            ne=ne+1;
                            ix=ne;
                        end
                    else
                        ne=ne+1;
                        ix=ne;
                    end
                    el(ix,1)={elnam};
                    el(ix,3)={str2double(rem)};
                end
            end
        end
    end
end
end

```

```

        end
    end
end
%
% Close input file
%
fclose(fid);

```

## A2.10 FVFARF\_read\_ts.m

```

function nuclides = FVFARF_read_ts(n,filename)
%FVFARF_read Read an ordinary FARF31 timeseries file (in.ts)
%
% FVFARF_read_ts(filename) reads an ordinary FARF31 timeseriea file for the
stand
% alone version and fills a cell array of nuclides with time series data
%
%
% Initialize return values
%
nuclides=n;
ix=[];
ts=[];
len=0;
%
% Open input file
%
fid=fopen(filename,'rt');
if fid==-1
    error('Could not open file.');
```

```

        ts(len,2)=str2double(token);
    end
end
end
%
% Add the time series to the current nuclide
%
if ~(isempty(ix))
    nuclides(ix,7)={ts};
    ix=[];
    ts=[];
    len=0;
end
%
% Close input file
%
fclose(fid);

```

## A2.11 FVFARF\_set\_par.m

```

function nuclides=FVFARF_set_par(n,s,par,el)
%FVFARF_set_par Set physical parameters of each nuclide
%
% FVFARF_set_par(n,s,par,el) sets the distribution coefficient Kd and the
% effective diffusion coefficient of each nuclide in a cell array of
nuclides n,
% based on the contents of the data structures created by reading the input
and
% parameter files, i.e. the cell arrays, par and el.
%
%
% Initialize nuclides
%
nuclides=n;
%
% Set the distribution coefficient for each nuclide
%
for i=1:size(nuclides,1)
    ix=find(strncmpi(el(:,1),nuclides(i,1),2));
    if isempty(ix)
        error('Could not find element data in input.')
```

```

    else
        nuclides(i,5)=el(ix,2);
    end
end
%
% Check if a single diffusion coefficient should be used
%
if s
    ix=find(strncmpi(par(:,1),'DE',2));
    if isempty(ix)
        error('Could not find single diffusivity in input.');
```

```

    else
        de=par(ix,2);
        for i=1:size(nuclides,1)
            nuclides(i,6)=de;
        end
    end
else
    %

```

```

% Set the nuclide specific diffusion coefficient for each nuclide
%
for i=1:size(nuclides,1)
    ix=find(strncmpi(el(:,1),nuclides(i,1),2));
    if isempty(ix)
        error('Could not find element data in input.')
```

```

    else
        nuclides(i,6)=el(ix,3);
    end
end
end
end

```

## A2.12 FVFARF\_setup.m

```

function [A,B] =
FVFARF_setup(NI,NJ,dx,dz,us,um,Ds,Dm,Dp,phi,b,R,kappal,kappa2,k1,k2,beta,LAM
BDA,COMPIN,eta,LSHEME)
%FVFARF_SETUP Set up the system matrix for one nuclide of FVFARF
%
% FVFARF_setup(NI,NJ) sets up the system matrix and the right hand side
% for the ODE generated by discretizing the FVFARF equations in space.
% NI is the number of points in the streamwise direction
% NJ is the number of points in the direction perpendicular to the stream
% tube (i.e. into the rock matrix)

%
% Create output variables
%
A = zeros(NI*NJ,NI*NJ);
B = zeros(NI*NJ,1);
%
% Outermost loop in the J direction (perpendicular)
%
for J=1:NJ
    %
    % Innermost loop in the I direction (streamwise)
    %
    for I=1:NI
        %
        % Calculate all indices we need in this cell
        % P is the cell itself
        % N means one cell further from the stream tube
        % S means one cell closer to the stream tube
        % W means one cell closer to the inlet
        % E means one cell closer to the outlet
        % Double and triple indices means a distance of 2 and 3 cells
        % respectively
        %
        P=(I-1)*NJ+J;
        W=P-NJ;
        WW=P-2*NJ;
        S=P-1;
        SS=P-2;
        N=P+1;
        NN=P+2;
        NNN=P+3;
        E=P+NJ;
        EE=P+2*NJ;
        %
        % Switch between the stream tube and the rock matrix
        %
        switch J
            %
            % J=1 is the index for the solution transport

```

```

%
case 1
%
% Switch between upwind differences and central differences
%
switch LSCHEME
% Upwind
case 'UDS'
%
% Switch between inflow, outflow and ordinary case
%
switch I
case 1
%INFLOW BOUNDARY
%CONVECTIVE SCHEME UDS
A(P,P)=A(P,P)-us/dz-Ds/dz^2;
A(P,E)=A(P,E)+Ds/dz^2;
B(1)=COMPIN*eta/dz/2/b;
case NI
%OUTFLOW BOUNDARY
%CONVECTIVE SCHEME UDS
A(P,P)=A(P,P)-us/dz+Ds/dz^2;
A(P,W)=A(P,W)+us/dz-2*Ds/dz^2;
A(P,WW)=A(P,WW)+Ds/dz^2;
otherwise
%CONVECTIVE SCHEME UDS
A(P,P)=A(P,P)-us/dz-2*Ds/dz^2;
A(P,W)=A(P,W)+us/dz+Ds/dz^2;
A(P,E)=A(P,E)+Ds/dz^2;
end
%Central
case 'CDS'
%
% Switch between inflow, outflow and ordinary case
%
switch I
case 1
%INFLOW BOUNDARY
A(P,P)=A(P,P)-us/dz/2-Ds/dz^2;
A(P,E)=A(P,E)-us/dz/2+Ds/dz^2;
B(1)=COMPIN*eta/2/b/dz;
case NI
%OUTFLOW BOUNDARY
A(P,P)=A(P,P)-11/8*us/dz+Ds/dz^2;
A(P,W)=A(P,W)+7/4*us/dz-2*Ds/dz^2;
A(P,WW)=A(P,WW)-3/8*us/dz+Ds/dz^2;
otherwise
A(P,P)=A(P,P)-2*Ds/dz^2;
A(P,W)=A(P,W)+us/dz/2+Ds/dz^2;
A(P,E)=A(P,E)-us/dz/2+Ds/dz^2;
end
end

%ADD TRANSVERSE TERM
A(P,P)=A(P,P)-phi*Dp/b*(8/3/dx(3));
A(P,NN)=A(P,NN)+phi*Dp/b*(8/3/dx(3)+2/3/(dx(3)+dx(4)));
A(P,NNN)=A(P,NNN)-phi*Dp/b*(2/3/(dx(3)+dx(4)));
%ADD DECAY
A(P,P)=A(P,P)-LAMBDA;
%ADD COLLOID TERM
A(P,P)=A(P,P)-(kappal*k1+kappa2*k2); %P represents
nuclides in solution
A(P,N)=A(P,N)+(kappal+beta*kappa2); %in this case North is
the mobile colloids

```

```

%
% J=2 is the index for the colloid transport
%
case 2
%
% Switch between upwind differences and central differences
%
switch LSCHEME
% Upwind
case 'UDS'
%
% Switch between inflow, outflow and ordinary case
%
switch I
case 1
%INFLOW BOUNDARY
%CONVECTIVE SCHEME UDS
A(P,P)=A(P,P)-um/(1+beta)/dz-
Dm/(1+beta)/dz^2;
A(P,E)=A(P,E)+Dm/(1+beta)/dz^2;
B(2)=COMPIN*(1-eta)/(1+beta)/dz/2/b;
case NI
%OUTFLOW BOUNDARY
%CONVECTIVE SCHEME UDS
A(P,P)=A(P,P)-
um/(1+beta)/dz+Dm/(1+beta)/dz^2;
A(P,W)=A(P,W)+um/(1+beta)/dz-
2*Dm/(1+beta)/dz^2;
A(P,WW)=A(P,WW)+Dm/(1+beta)/dz^2;
otherwise
%CONVECTIVE SCHEME UDS
A(P,P)=A(P,P)-um/(1+beta)/dz-
2*Dm/(1+beta)/dz^2;
A(P,W)=A(P,W)+um/(1+beta)/dz+Dm/(1+beta)/dz^2;
A(P,E)=A(P,E)+Dm/(1+beta)/dz^2;
end
%Central
case 'CDS'
%
% Switch between inflow, outflow and ordinary case
%
switch I
case 1
%INFLOW BOUNDARY
A(P,P)=A(P,P)-um/(1+beta)/dz/2-
Dm/(1+beta)/dz^2;
A(P,E)=A(P,E)-
um/(1+beta)/dz/2+Dm/(1+beta)/dz^2;
B(2)=COMPIN*(1-eta)/(1+beta)/2/b/dz;
case NI
%OUTFLOW BOUNDARY
A(P,P)=A(P,P)-
11/8*um/(1+beta)/dz+Dm/(1+beta)/dz^2;
A(P,W)=A(P,W)+7/4*um/(1+beta)/dz-
2*Dm/(1+beta)/dz^2;
A(P,WW)=A(P,WW)-
3/8*um/(1+beta)/dz+Dm/(1+beta)/dz^2;
otherwise
A(P,P)=A(P,P)-2*Dm/dz^2/(1+beta);
A(P,W)=A(P,W)+um/(1+beta)/dz/2+Dm/dz^2/(1+beta);
A(P,E)=A(P,E)-
um/(1+beta)/dz/2+Dm/dz^2/(1+beta);
end

```



```

end
%ADD DECAY
A(P,P)=A(P,P)-LAMBDA;
%ADD COLLOID TERM
A(P,P)=A(P,P)-(kappa1+beta*kappa2)/(1+beta);
A(P,S)=A(P,S)+(kappa1*k1+kappa2*k2)/(1+beta);
%
% All other J values for diffusion in the rock matrix
%
otherwise
%
% Switch between cells next to the stream tube, interior
% cells and cells at the outer boundary
%
switch J
case 3
%South boundary
A(P,P)=A(P,P)-
8/3*Dp/R/dx(J)*(1/(dx(J)+dx(J+1))+1/dx(J));
A(P,N)=A(P,N)+8/3*Dp/R/dx(J)/(dx(J)+dx(J+1));
A(P,SS)=A(P,SS)+8/3*Dp/R/dx(J)/(dx(J));
case NJ
A(P,P)=A(P,P)-Dp/R/dx(J)*2/(dx(J)+dx(J-1));
A(P,S)=A(P,S)+Dp/R/dx(J)*2/(dx(J)+dx(J-1));
otherwise
A(P,P)=A(P,P)-
Dp/R/dx(J)*(2/(dx(J)+dx(J+1))+2/(dx(J)+dx(J-1)));
A(P,S)=A(P,S)+Dp/R/dx(J)*2/(dx(J)+dx(J-1));
A(P,N)=A(P,N)+Dp/R/dx(J)*2/(dx(J)+dx(J+1));
end
%ADD DECAY
A(P,P)=A(P,P)-LAMBDA;
end
%
% End swich over J cell locations
%
end
%
% End I loop
%
end
%
% End J loop

```

## A2.13 FVFARF\_setup\_system.m

```

function [A,B,LSHEME]=FVFARF_setup_system(nuclides,NI,NJ,dz,dx,us,um,Ds,Dm,
phi,b,eps,kappa1,kappa2,k1,k2,beta,eta,FARFporosity)
%FVFARF_setup_system Set up the system matrix of the FVFARF system
%
% FVFARF_setup_system(A,B,nuclides) sets up the sysem matrix A and the RHS
% B of the FVFARF system based on a cell array of nuclide data.
%
%
% Allocate space for the system matrix and the RHS
%
NK=size(nuclides,1);
A=zeros(NI*NJ*NK,NI*NJ*NK);
B=zeros(NI*NJ*NK,1);
%

```

```

% Select discetization schemes
%
if (dz*us/Ds)>2
    LSCHEME='UDS';
else
    LSCHEME='CDS';
end

%
% Calculate decay constant LAMBDA for all nuclides
%
thalf=[nuclides(:,2)];
LAMBDA=log(2)./thalf;

%
% Setup daughter vector
%
daughter=[nuclides(:,3)];

%
% Calculate retardation factor for all nuclides
%
Kd=[nuclides(:,5)];
if FAREFporosity>0
    % This is the ordinary FARF model for the retardation factor
    R=eps+Kd*2700.0;
else
    % This is the COLLAGE model for the retardation factor
    R=1.0+Kd*2700.0*(1.0-phi)/phi
end

%
% Calculate effective matrix diffusivity for all nuclides
%
Dp=[nuclides(:,6)];

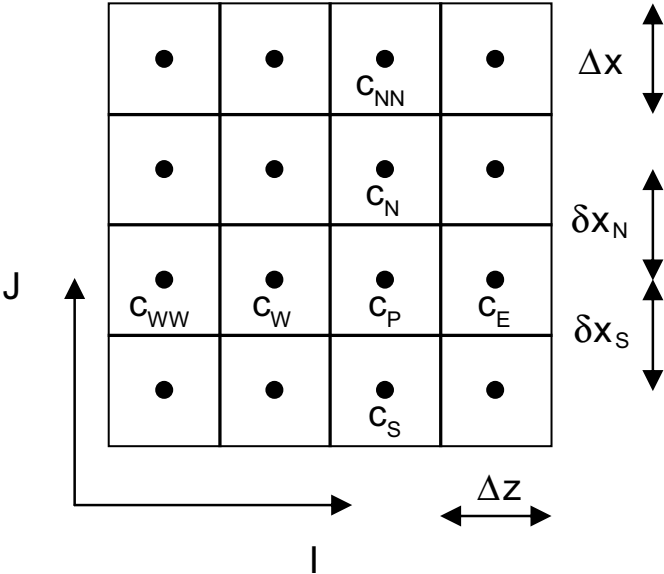
%
% Setup the system matrix and right hand side of the ODE. Loop over
% nuclides:
%
for K=1:NK
    %
    % Setup the nuclide specific system matrix and RHS
    %
    [AA,BB] =
FVFARF_setup(NI,NJ,dx,dz,us,um,Ds,Dm,Dp(K),phi,b,R(K),kappa1,kappa2,k1,k2,beta,
LAMBDA(K),1.0,eta,LSCHEME);
    %
    % Add the contribution in its correct place
    %
    rc1=NI*NJ*(K-1)+1;
    rc2=rc1+NI*NJ-1;
    A(rc1:rc2,rc1:rc2)=AA;
    B(rc1:rc2)=BB;
    %
    % Add source terms for decay of mother nuclide
    %
    if K>1
        c1=NI*NJ*(K-2)+1;
        c2=c1+NI*NJ-1;
        A(rc1:rc2,c1:c2)=A(rc1:rc2,c1:c2)+daughter(K-1)*LAMBDA(K-
1)*eye(NI*NJ)*R(K-1)/R(K);
    end
end
end

```

**A3 Discretisation techniques**

**A3.1 Computational domain**

When discretising a cell P, the compass notation, Figure A3-1, is used to denote neighbouring cells. Upper case indices denote that the property is evaluated at the centre of the node while lower case indices denote properties evaluated at the boundary.  $\Delta$  is used to denote the size of an element while  $\delta$  denotes the distance between nodes (this distinction is relevant for non-equidistant meshes).



*Figure A3-1. Notations used when accessing properties in the computational domain.*

### A3.2 Discretisation of the advection – dispersion equation

Start with the conservation equation

$$\frac{\partial}{\partial t} \int_{\Omega} c^i dV + \oint_S \left( uc^i_s - D_s \frac{\partial c^i_s}{\partial z} \right) \cdot dS = \int_{\Omega} Q dV$$

$$\text{where } Q = \lambda^{i-1} c^{i-1}_s - \lambda^i c^i_s + (\kappa_1 + \beta \kappa_2) c^i_m - (\kappa_1 k_1 + \kappa_2 k_2) c^i_s + a_w D_e \frac{\partial c^i_p}{\partial x} \Big|_{x=0}$$

Applying Gauss's theorem yields

$$\frac{\partial c_s}{\partial t} = \frac{D_s}{\Delta z} \left( \frac{\partial c_s}{\partial z} \Big|_e - \frac{\partial c_s}{\partial z} \Big|_w \right) - \frac{u_s}{\Delta z} (c_s \Big|_e - c_s \Big|_w) + Q$$

Discretise using UDS

$$c_s \Big|_e = c_P$$

$$c_s \Big|_w = c_W$$

$$\frac{\partial c_s}{\partial z} \Big|_e = \frac{c_E - c_P}{\Delta z}$$

$$\frac{\partial c_s}{\partial z} \Big|_w = \frac{c_P - c_W}{\Delta z}$$

$$\frac{\partial c_s}{\partial t} = \frac{D_s}{\Delta z} \left( \frac{c_E - c_P}{\Delta z} - \frac{c_P - c_W}{\Delta z} \right) - \frac{u_s}{\Delta z} (c_P - c_W) + Q$$

$$\frac{\partial c_s}{\partial t} = c_P \left( -2 \frac{D_s}{(\Delta z)^2} - \frac{u_s}{\Delta z} \right) + c_W \left( \frac{D_s}{(\Delta z)^2} + \frac{u_s}{\Delta z} \right) + c_E \left( \frac{D_s}{(\Delta z)^2} \right)$$

Discretise using CDS

$$c_s \Big|_e = \frac{c_P + c_E}{2}$$

$$c_s \Big|_w = \frac{c_W + c_P}{2}$$

$$\frac{\partial c_s}{\partial z} \Big|_e = \frac{c_E - c_P}{\Delta z}$$

$$\frac{\partial c_s}{\partial z} \Big|_w = \frac{c_P - c_W}{\Delta z}$$

$$\frac{\partial c_s}{\partial t} = \frac{D_s}{\Delta z} \left( \frac{c_E - c_P}{\Delta z} - \frac{c_P - c_W}{\Delta z} \right) - \frac{u_s}{\Delta z} \left( \frac{c_P + c_E}{2} - \frac{c_W + c_P}{2} \right) + Q$$

$$\frac{\partial c_s}{\partial t} = c_P \left( -2 \frac{D_s}{(\Delta z)^2} \right) + c_W \left( \frac{u_s}{2\Delta z} + \frac{D_s}{(\Delta z)^2} \right) + c_E \left( -\frac{u_s}{2\Delta z} + \frac{D_s}{(\Delta z)^2} \right)$$

### A3.2.1 Inlet boundary

At the inlet, the flux at the western boundary is given

$$2b \left[ u_s c_s \Big|_w - D_s \frac{\partial c}{\partial z} \Big|_w \right] = \eta In(t)$$

hence, the conservation equation for the western element is

$$\frac{\partial c_s}{\partial t} = \frac{D_s}{\Delta z} \frac{\partial c_s}{\partial z} \Big|_e - \frac{u_s}{\Delta z} c_s \Big|_e + Q + \frac{\eta In(t)}{2b\Delta z}$$

Discretise using UDS

$$\begin{aligned} c_s \Big|_e &= c_P \\ \frac{\partial c_s}{\partial z} \Big|_e &= \frac{c_E - c_P}{\Delta z} \\ \frac{\partial c_s}{\partial t} &= \frac{D_s}{\Delta z} \frac{c_E - c_P}{\Delta z} - \frac{u_s}{\Delta z} c_P + Q + \frac{\eta In(t)}{2b\Delta z} \\ \frac{\partial c_s}{\partial t} &= c_P \left( -\frac{u_s}{\Delta z} - \frac{D_s}{(\Delta z)^2} \right) + c_E \left( \frac{D_s}{(\Delta z)^2} \right) + Q + B \\ B &= \frac{\eta In(t)}{2b\Delta z} \end{aligned}$$

Discretise using CDS

$$\begin{aligned} c_s \Big|_e &= \frac{c_P + c_E}{2} \\ \frac{\partial c_s}{\partial z} \Big|_e &= \frac{c_E - c_P}{\Delta z} \\ \frac{\partial c_s}{\partial t} &= \frac{D_s}{\Delta z} \frac{c_E - c_P}{\Delta z} - \frac{u_s}{\Delta z} \frac{c_P + c_E}{2} + Q + \frac{\eta In(t)}{2b\Delta z} \\ \frac{\partial c_s}{\partial t} &= c_P \left( -\frac{u_s}{2\Delta z} - \frac{D_s}{(\Delta z)^2} \right) + c_E \left( -\frac{u_s}{2\Delta z} + \frac{D_s}{(\Delta z)^2} \right) + Q + B \\ B &= \frac{\eta In(t)}{2b\Delta z} \end{aligned}$$

### A3.2.2 Outlet boundary

Upstream nodes are used to extrapolate the value at the outlet boundary using a second order polynomial

$$Az^2 + Bz + C = y(z) \quad \text{where} \quad y(0) = c_p \quad y(-\Delta z) = c_w \quad y(-2\Delta z) = c_{ww}$$

$$A = \frac{c_{ww} - 2c_w + c_p}{2\Delta z^2}$$

$$B = \frac{3c_p + c_{ww} - 4c_w}{2\Delta z}$$

$$C = c_p$$

yielding

$$c_s|_e = \frac{c_{ww} - 2c_w + c_p}{2\Delta z^2} \left( \frac{\Delta z}{2} \right)^2 + \frac{3c_p + c_{ww} - 4c_w}{2\Delta z} \left( \frac{\Delta z}{2} \right) + c_p = \frac{c_{ww} - 2c_w + c_p}{8} + \frac{3c_p + c_{ww} - 4c_w}{4} + c_p = \frac{15}{8}c_p - \frac{5}{4}c_w + \frac{3}{8}c_{ww}$$

$$\frac{\partial c_s}{\partial z} \Big|_e = 2 \frac{c_{ww} - 2c_w + c_p}{2\Delta z^2} \left( \frac{\Delta z}{2} \right) + \frac{3c_p + c_{ww} - 4c_w}{2\Delta z} = \frac{c_{ww} - 2c_w + c_p}{2\Delta z} + \frac{3c_p + c_{ww} - 4c_w}{2\Delta z} = \frac{2c_p - 3c_w + c_{ww}}{\Delta z}$$

Discretise using UDS

$$c_s|_e = c_P$$

$$c_s|_w = c_W$$

$$\frac{\partial c_s}{\partial z}\bigg|_e = \frac{\partial c_s}{\partial z}\bigg|_w + \frac{\frac{\partial c_s}{\partial z}\bigg|_w - \frac{\partial c_s}{\partial z}\bigg|_{ww}}{\Delta z} \Delta z = 2 \frac{\partial c_s}{\partial z}\bigg|_w - \frac{\partial c_s}{\partial z}\bigg|_{ww} = \frac{2c_P - 3c_W + c_{WW}}{\Delta z}$$

$$\frac{\partial c_s}{\partial z}\bigg|_w = \frac{c_P - c_W}{\Delta z}$$

$$\frac{\partial c_s}{\partial t} = \frac{D_s}{\Delta z} \left( \frac{2c_P - 3c_W + c_{WW}}{\Delta z} - \frac{c_P - c_W}{\Delta z} \right) - \frac{u_s}{\Delta z} (c_P - c_W) + Q$$

$$\frac{\partial c_s}{\partial t} = c_P \left( -\frac{u_s}{\Delta z} + \frac{D_s}{(\Delta z)^2} \right) + c_W \left( \frac{u_s}{\Delta z} - \frac{2D_s}{(\Delta z)^2} \right) + c_{WW} \left( \frac{2D_s}{(\Delta z)^2} \right) + Q$$

Discretise using CDS

$$c_s|_e = \frac{15}{8}c_P - \frac{5}{4}c_W + \frac{3}{8}c_{WW}$$

$$c_s|_w = \frac{c_P + c_W}{2}$$

$$\frac{\partial c_s}{\partial z}\bigg|_e = \frac{\partial c_s}{\partial z}\bigg|_w + \frac{\frac{\partial c_s}{\partial z}\bigg|_w - \frac{\partial c_s}{\partial z}\bigg|_{ww}}{\Delta z} \Delta z = 2 \frac{\partial c_s}{\partial z}\bigg|_w - \frac{\partial c_s}{\partial z}\bigg|_{ww} = \frac{2c_P - 3c_W + c_{WW}}{\Delta z}$$

$$\frac{\partial c_s}{\partial z}\bigg|_w = \frac{c_P - c_W}{\Delta z}$$

$$\frac{\partial c_s}{\partial t} = \frac{D_s}{\Delta z} \left( \frac{2c_P - 3c_W + c_{WW}}{\Delta z} - \frac{c_P - c_W}{\Delta z} \right) - \frac{u_s}{\Delta z} \left( \frac{15}{8}c_P - \frac{5}{4}c_W + \frac{3}{8}c_{WW} - \frac{c_P + c_W}{2} \right) + Q$$

$$\frac{\partial c_s}{\partial t} = c_P \left( -\frac{7u_s}{8\Delta z} + \frac{D_s}{(\Delta z)^2} \right) + c_W \left( \frac{9u_s}{4\Delta z} - \frac{2D_s}{(\Delta z)^2} \right) + c_{WW} \left( -\frac{3u_s}{8\Delta z} - \frac{2D_s}{(\Delta z)^2} \right) + Q$$

### A3.2.3 Outflow

The outflow from the stream tube from the solute phase is expressed as

$$2b \left[ u_s c_s \Big|_e - D_s \frac{\partial c}{\partial z} \Big|_e \right] = Out$$

Discretise using UDS

$$c_s \Big|_e = c_P$$

$$c_s \Big|_w = c_W$$

$$\frac{\partial c_s}{\partial z} \Big|_e = \frac{2c_P - 3c_W + c_{WW}}{\Delta z}$$

$$\frac{\partial c_s}{\partial z} \Big|_w = \frac{c_P - c_W}{\Delta z}$$

$$2b \left[ u_s c_P - D_s \frac{2c_P - 3c_W + c_{WW}}{\Delta z} \right] = Out$$

$$2b \left[ c_P \left( u_s - 2 \frac{D_s}{\Delta z} \right) + c_W 3 \frac{D_s}{\Delta z} - c_{WW} \frac{D_s}{\Delta z} \right] = Out$$

Discretise using CDS

$$c_s \Big|_e = \frac{15}{8} c_P - \frac{5}{4} c_W + \frac{3}{8} c_{WW}$$

$$\frac{\partial c_s}{\partial z} \Big|_e = \frac{2c_P - 3c_W + c_{WW}}{\Delta z}$$

$$2b \left[ u_s \left( \frac{15}{8} c_P - \frac{5}{4} c_W + \frac{3}{8} c_{WW} \right) - D_s \frac{2c_P - 3c_W + c_{WW}}{\Delta z} \right] = Out$$

$$2b \left[ c_P \left( \frac{15u_s}{8} - \frac{2D_s}{\Delta z} \right) + c_W \left( -\frac{5u_s}{4} + \frac{3D_s}{\Delta z} \right) + c_{WW} \left( \frac{3u_s}{8} - \frac{D_s}{\Delta z} \right) \right] = Out$$

### A3.3 Matrix diffusion

$$\frac{\partial}{\partial t} \int_{\Omega} c^i_p dV - \oint_S \frac{D_e^i}{R^i} \frac{\partial c^i_p}{\partial x} \cdot dS = \int_{\Omega} Q dV$$

$$\text{where } Q = \lambda^{i-1} c^{i-1}_s - \lambda^i c^i_{sm}$$

$$R = \varepsilon + K_d \rho$$

yields

$$\frac{\partial c_p}{\partial t} = \frac{1}{\Delta x_p} \frac{D_e}{R} \left( \frac{\partial c_p}{\partial x} \Big|_n - \frac{\partial c_p}{\partial x} \Big|_s \right) + Q$$



Discretise using CDS

$$\left. \frac{\partial c_p}{\partial x} \right|_n = 2 \frac{c_N - c_P}{\Delta x_P + \Delta x_N}$$

$$\left. \frac{\partial c_p}{\partial x} \right|_s = 2 \frac{c_P - c_S}{\Delta x_P + \Delta x_S}$$

$$\frac{\partial c_p}{\partial t} = 2 \frac{D_e}{\Delta x_P R} \left( \frac{c_N - c_P}{\Delta x_P + \Delta x_N} - \frac{c_P - c_S}{\Delta x_P + \Delta x_S} \right) + Q$$

$$\frac{\partial c_p}{\partial t} = c_P \left( \frac{-2D_e}{\Delta x_P R (\Delta x_P + \Delta x_N)} + \frac{-2D_e}{\Delta x_P R (\Delta x_P + \Delta x_S)} \right) + c_N \left( \frac{2D_e}{\Delta x_P R (\Delta x_P + \Delta x_N)} \right) + c_S \left( \frac{2D_e}{\Delta x_P R (\Delta x_P + \Delta x_S)} \right) + Q$$

### A3.3.1 Outer boundary in the transverse direction

At the outer in the transverse boundary a symmetry boundary condition is used

Discretise using CDS

$$\left. \frac{\partial c}{\partial x} \right|_n = 0$$

$$\left. \frac{\partial c}{\partial x} \right|_s = 2 \frac{c_P - c_S}{\Delta x_S + \Delta x_N}$$

$$\frac{\partial c}{\partial t} = \frac{D_e}{\Delta x_P R} \left( 0 - 2 \frac{c_P - c_S}{\Delta x_S + \Delta x_N} \right) + Q$$

$$\frac{\partial c}{\partial t} = c_P \left( \frac{-2D_e}{\Delta x_P R (\Delta x_S + \Delta x_N)} \right) + c_S \left( \frac{2D_e}{\Delta x_P R (\Delta x_S + \Delta x_N)} \right) + Q$$

### A3.4 Mobile colloids

$$\frac{\partial}{\partial t} \int_{\Omega} c^i_m (1 + \beta) dV + \oint_S u_m c^i_m - D_m \frac{\partial c^i_m}{\partial z} dS = \int_{\Omega} Q dV$$

$$\text{where } Q = \left[ \lambda^{i-1} c^{i-1}_m - \lambda^i c^i_m - (\kappa_1 + \beta \kappa_2) c^i_m + (\kappa_1 k_1 + \kappa_2 k_2) c^i_s \right]$$

Applying Gauss's theorem and perform discretisation analogous to the advection-dispersion equation yields

Discretise using UDS

$$(1 + \beta) \frac{\partial c_m}{\partial t} = \frac{D_m}{\Delta z} \left( \frac{c_P - c_E}{\Delta z} - \frac{c_W - c_P}{\Delta z} \right) - \frac{u_m}{\Delta z} (c_P - c_W) + Q$$

Discretise using CDS

$$(1 + \beta) \frac{\partial c_m}{\partial t} = \frac{D_m}{\Delta z} \left( \frac{2c_P - 3c_W + c_{WW}}{\Delta z} - \frac{c_P - c_W}{\Delta z} \right) - \frac{u_m}{\Delta z} \left( \frac{15}{8} c_P - \frac{5}{4} c_W + \frac{3}{8} c_{WW} - \frac{c_W + c_P}{2} \right) + Q$$

### A3.4.1 Inlet boundary

Analogous to the advection-dispersion equation

Discretise using UDS

$$(1 + \beta) \frac{\partial c_m}{\partial t} = \frac{D_m}{\Delta z} \frac{c_E - c_P}{\Delta z} - \frac{u_m}{\Delta z} c_P + Q + \frac{\eta \ln(t)}{2b\Delta z}$$

Discretise using CDS

$$(1 + \beta) \frac{\partial c_m}{\partial t} = \frac{D_m}{\Delta z} \frac{c_E - c_P}{\Delta z} - \frac{u_m}{\Delta z} \frac{c_P + c_E}{2} + Q + \frac{\eta \ln(t)}{2b\Delta z}$$

### A3.4.2 Outlet boundary

Analogous to the advection-dispersion equation

Discretise using UDS

$$(1 + \beta) \frac{\partial c_m}{\partial t} = \frac{D_m}{\Delta z} \left( \frac{2c_P - 3c_W + c_{WW}}{\Delta z} - \frac{c_P - c_W}{\Delta z} \right) - \frac{u_m}{\Delta z} (c_P - c_W) + Q$$

Discretise using CDS

$$(1 + \beta) \frac{\partial c_m}{\partial t} = \frac{D_m}{\Delta z} \left( \frac{2c_P - 3c_W + c_{WW}}{\Delta z} - \frac{c_P - c_W}{\Delta z} \right) - \frac{u_m}{\Delta z} \left( \frac{c_P + c_E}{2} - \frac{c_P + c_W}{2} \right) + Q$$

### A3.4.3 Outflow

Analogous to the advection-dispersion equation

Discretise using UDS

$$2b \left[ c_P \left( u_m - 2 \frac{D_m}{\Delta z} \right) + c_W \left( 3 \frac{D_m}{\Delta z} - c_{WW} \frac{D_m}{\Delta z} \right) \right] = Out$$

Discretise using CDS

$$2b \left[ c_P \left( \frac{15u_m}{8} - \frac{2D_m}{\Delta z} \right) + c_W \left( -\frac{5u_m}{4} + \frac{3D_m}{\Delta z} \right) + c_{WW} \left( \frac{3u_m}{8} - \frac{D_m}{\Delta z} \right) \right] = Out$$

## A3.5 Source term discretisation

The different source terms Q are discretised and added to their respective equations

### A3.5.1 Decay

The volume source term for decay is

$$\int_{\Omega} (\lambda^{i-1} c^{i-1} - \lambda^i c^i) dV$$

$$\text{Hence } Q = \lambda^{i-1} c_P^{i-1} - \lambda^i c_P^i$$

$$\frac{\partial c_P^i}{\partial t} = \lambda^{i-1} c_P^{i-1} - \lambda^i c_P^i$$

### A3.5.2 Sink for transverse flux in the advection-dispersion equation

The term for the transverse flux of the transport equation is

$$\int_{\Omega} a_w d\Omega D_e \left. \frac{\partial c_p^i}{\partial x} \right|_{x=0} dV$$

$$\text{hence } Q = a_w D_e \left. \frac{\partial c_p^i}{\partial x} \right|_{x=0}$$

Extrapolate the node values in the transverse direction (to the north of the stream tube using the compass notation described above) to get the differential at the boundary. Remember that in this case (the stream tube is located at J=1 and the rock matrix starts at J=3), the concentrations  $c_{NN}$  and  $c_{NNN}$  are used ( $c_N$  corresponds to the concentration in the stream tube).

$$\left. \frac{\partial c_p}{\partial x} \right|_{x=\frac{\Delta x}{4}} = 2 \frac{c_{NN} - c_P}{\Delta x_{NN}}$$

$$\left. \frac{\partial c_p}{\partial x} \right|_{x=\Delta x} = 2 \frac{c_{NNN} - c_{NN}}{\Delta x_{NN} + \Delta x_{NNN}}$$

$$\left. \frac{\partial c_p}{\partial x} \right|_{x=0} = \left. \frac{\partial c_p}{\partial x} \right|_{x=\frac{\Delta x}{4}} + \frac{\left. \frac{\partial c_p}{\partial x} \right|_{x=\frac{\Delta x}{4}} - \left. \frac{\partial c_p}{\partial x} \right|_{x=\Delta x}}{\frac{3\Delta x_p}{4}} \left( -\frac{1}{4} \Delta x_{NN} \right) = \frac{4}{3} \left. \frac{\partial c_p}{\partial x} \right|_{x=\frac{\Delta x}{4}} - \frac{1}{3} \left. \frac{\partial c_p}{\partial x} \right|_{x=\Delta x}$$

$$\frac{\partial c_p}{\partial t} = \frac{\phi D_p}{b} \left. \frac{\partial c_p}{\partial x} \right|_{x=0} = \frac{\phi D_p}{b} \left( \frac{8}{3} \frac{c_{NN} - c_P}{\Delta x_{NN}} - \frac{2}{3} \frac{c_{NNN} - c_{NN}}{\Delta x_{NN} + \Delta x_{NNN}} \right)$$

### A3.5.3 Colloidal exchange

The exchange term in the advection-dispersion equation

$$\int_{\Omega} (\kappa_1 + \beta \kappa_2) c_m - (\kappa_1 k_1 + \kappa_2 k_2) c_s dV$$

$$Q = (\kappa_1 + \beta \kappa_2) c_m - (\kappa_1 k_1 + \kappa_2 k_2) c_s$$

colloidal transport occurs at J = 2, hence concentrations are stored in the northern nodes

$$\frac{\partial c_p}{\partial t} = (\kappa_1 + \beta \kappa_2) c_N - (\kappa_1 k_1 + \kappa_2 k_2) c_P$$

The exchange term in the equation for colloid facilitated transport

$$\int_{\Omega} \frac{\kappa_1 + \beta \kappa_2}{1 + \beta} c_m - \frac{\kappa_1 k_1 + \kappa_2 k_2}{1 + \beta} c_s dV$$

colloidal transport occurs at J = 2, hence concentrations are stored in the northern nodes

$$\frac{\partial c_p}{\partial t} = \frac{\kappa_1 + \beta \kappa_2}{1 + \beta} c_P - \frac{\kappa_1 k_1 + \kappa_2 k_2}{1 + \beta} c_N$$

### A3.6 Comparison between the variables used in the FARF31 code and the Collage II Plus code

Advection-dispersion equation used in the COLLAGE II code

$$\frac{\partial c_s}{\partial t} = D_s \frac{\partial^2 c_s}{\partial z^2} - u_s \frac{\partial c_s}{\partial z} + \frac{\phi D_p}{b} \frac{\partial c_p}{\partial x} \Big|_{x=0} - \lambda c_s + (\kappa_1 + \beta \kappa_2) c_m - (\kappa_1 k_1 + \kappa_2 k_2) c_s$$

Advection-dispersion equation used in the FARF model /Norman and Kjellbert 1990/

$$\frac{\partial c_s(\zeta, t)}{\partial t} = - \frac{\partial c_s(\zeta, t)}{\partial \zeta} + \frac{\partial}{\partial \zeta} \frac{D_s(\zeta)}{u_s(\zeta)^2} \frac{\partial c_s(\zeta, t)}{\partial \zeta} + a(\zeta) D_e(\zeta, x) \frac{\partial c_p(x, \zeta, t)}{\partial x} \Big|_{x=0} - \lambda c_s(z, t)$$

If constant  $D_s$  and  $u_s$  are used the following relations exist (as shown in the FARF31 manual)

$$u_s = \frac{L}{t_w} \text{ and } \frac{D_s}{u_s^2} = \frac{t_w}{Pe}$$

Identification of variables in the transverse diffusion equation

$$\frac{\partial c_p}{\partial t} = \frac{D}{R} \frac{\partial^2 c_p}{\partial x^2} - \lambda c_p$$

yields,

$$\frac{D_{Collage}}{R_{Collage}} = \frac{D_{FARF}}{R_{FARF}}$$

Where the following retardation coefficients have been used

$$R_{FARF} = \varepsilon + K_d \rho \text{ and } R_{COLLAGE} = 1 + \frac{1 - \varepsilon}{\varepsilon} K_d \rho$$

Hence, the two codes use a different definition for the diffusivities and FARF diffusivities cannot be used in the COLLAGE code etc.

Identifying variables in the sink for transverse diffusion in the advection-dispersion equation yields

$$\frac{\phi D_{Collage}}{b} = a D_{FARF}$$

$$\frac{\phi R_{Collage}}{b R_{FARF}} = a$$