

FTRANS_{1.0}

A model of contaminant transport in fracture networks

User Manual

Martin Vohralík^{1,2}

May 2, 2006

¹ Laboratoire de Mathématiques, Analyse Numérique et EDP, Université de Paris-Sud et CNRS, Bât. 425, 91 405 Orsay, France. martin.vohralik@math.u-psud.fr

² Department of Process Modeling, Faculty of Mechatronics and Interdisciplinary Engineering Studies, Technical University of Liberec, Hálkova 6, 461 17 Liberec 1, Czech Republic.
martin.vohralik@vslib.cz

Contents

1	Purpose and scope	4
2	Domain and equations	4
2.1	The fracture network	4
2.2	The contaminant transport equation	4
3	Numerical schemes	6
3.1	Space and time discretizations	6
3.2	The finite volume scheme	6
3.2.1	A posteriori error estimates	7
3.3	The mixed/condensed mixed finite element schemes	7
3.3.1	Centered scheme with vector unknowns representing full flux	8
3.3.2	Weighted upwind-mixed finite element scheme	8
3.3.3	Combination of a centered scheme with vector unknowns only representing diffusive flux and of the upwind-mixed scheme	9
3.3.4	Algebraic condensation to only scalar unknowns	10
3.3.5	A posteriori error estimates	10
3.4	The combined finite volume–finite element scheme	10
3.5	The combined finite volume–nonconforming finite element scheme	10
4	Using FTRANS	11
4.1	Requirements	11
4.2	Control file format	12
4.2.1	Section [Global]	12
4.2.2	Section [Input]	12
4.2.3	Section [Run]	13
4.2.4	Section [Solver]	13
4.2.5	Section [Output]	14
4.2.6	Section [Parameters]	15
4.2.7	Section [Particular]	16
4.3	Mesh file format	18
4.4	Time file format	19
4.5	Coefficients file format	20
4.6	Initial conditions file format	24
4.7	Boundary conditions file format	25
4.8	Output files format	25
4.8.1	Output file with results	25
4.8.2	Output file with a posteriori error estimates	26
4.9	Various remarks	26
4.9.1	Time discretization	26
4.9.2	Direct solvers	27
4.9.3	Inexact Newton method	27
4.9.4	LU factorization only at the first step	27
4.9.5	Particular cases for the different schemes	27
4.9.6	Recommendations on the choice between the different schemes	29

5	FTRANS organization	29
5.1	Structures	29
5.2	Basic code organization	30
5.3	Treatment of the data for parabolic problems	30
A	Example of a control file	32
B	Examples of input files	33
B.1	Example of a mesh file	33
B.2	Example of a time file	35
B.3	Example of a coefficients file	35
B.4	Example of initial conditions file	39
B.5	Example of boundary conditions file	40
C	Example of an output file	41

1 Purpose and scope

FTRANS is a program for numerical simulation of contaminant transport in the specific medium of rock massifs disrupted by a network of fractures, represented by a system of two-dimensional polygons placed in a three-dimensional space. It is designed for the specific geometry of fracture networks discretized to triangular three-dimensional meshes, which may include nonmatching interfaces between the different fractures. It models the evolution of a concentration of a single species taking into account the effects of convection, molecular diffusion and hydrodynamic dispersion, equilibrium adsorption, first-order irreversible reactions such as radioactive decay, hydrolysis, and some forms of biodegradation and this with various boundary conditions and sources or sinks in a variably-saturated fracture network. This problem mathematically amounts to a parabolic convection–diffusion–reaction equation which may be strongly nonlinear (of the degenerate parabolic type with a moving free boundary), where the diffusion–dispersion tensor is inhomogeneous and anisotropic (under the form of a nonconstant full matrix), and which is dominated by the convection. A dual porosity module is now being added by Milan Hokr from the Department of Process Modeling in TU Liberec, in order to include the influence of the surrounding rock matrix as well. FTRANS is a console MS DOS/WINDOWS application written in the C language under the C++ Builder 5 and based on FFLOW, a model for single-phase saturated flow in fracture networks by Otto Severýn from the Department of Process Modeling in TU Liberec.

2 Domain and equations

We describe here in more details the domain given by a fracture network and the contaminant transport equation.

2.1 The fracture network

The fracture network is given by a system \mathcal{S} ,

$$\mathcal{S} := \left\{ \bigcup_{\ell \in L} \overline{\alpha_\ell} \setminus \partial\mathcal{S} \right\}, \quad (2.1)$$

where α_ℓ is an open two-dimensional polygon placed in three-dimensional space and L is the index set of polygons. We call $\overline{\alpha_\ell}$, the closure of α_ℓ , a fracture. We suppose that each fracture is connected to the system by at least one edge in the sense that it has an intersection with at least one another fracture and that this intersection is a line segment. The boundary of \mathcal{S} , $\partial\mathcal{S}$, is the set of such edges of the fractures that do not create an intersection with another fracture. We refer to [22] for the definition of functions spaces on fracture networks.

2.2 The contaminant transport equation

The reactive miscible displacement with equilibrium adsorption of one contaminant in one fracture $\overline{\alpha_\ell}$ is given by the parabolic convection–diffusion–reaction in the form

$$\frac{\partial\beta(c)}{\partial t} - \nabla \cdot (\mathbf{S}\nabla c) + \nabla \cdot (c\mathbf{w}) + F(c) = q \text{ in } \alpha_\ell \times (0, T), \quad (2.2a)$$

$$c(\cdot, 0) = c_0 \text{ in } \alpha_\ell, \quad (2.2b)$$

$$c = u_D \text{ on } \Gamma_D \times (0, T), \quad (2.2c)$$

$$-\mathbf{S}\nabla c \cdot \mathbf{n} = u_N \text{ on } \Gamma_N \times (0, T), \quad (2.2d)$$

$$(-\mathbf{S}\nabla c + c\mathbf{w}) \cdot \mathbf{n} = u_R \text{ on } \Gamma_R \times (0, T). \quad (2.2e)$$

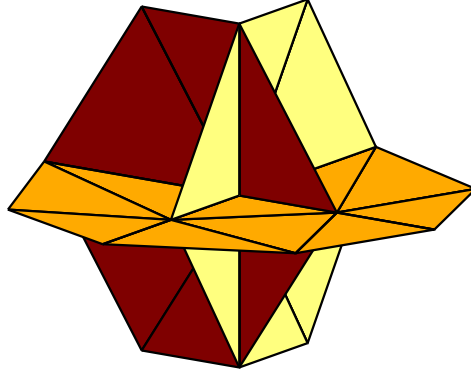


Figure 1: Example of a simple fracture network made of 3 polygons and discretized into a matching triangular mesh

Here c is the unknown concentration of the contaminant, the function $\beta(\cdot)$ represents time evolution and equilibrium adsorption reaction and is supposed to be continuous and increasing with the growth bounded from below by a positive constant, \mathbf{S} is the diffusion–dispersion tensor, \mathbf{w} is the Darcy velocity field defining the convection term, which is supposed to have a continuous normal trace over the interpolygon boundaries in the below-defined sense, the function $F(\cdot)$ represents the changes due to chemical reactions, and finally, q stands for the sources. Note that this equation is written in two-dimensional coordinates of the given fracture. In order to completely conform with contaminant transport models (cf. [1, 2, 12, 23]), the sources q can be divided to injections with prescribed concentration and pumping without prescribed concentration, i.e. $q = \max\{q, 0\}c_s + \min\{q, 0\}c$, so that the right-hand side of (2.2a) changes to $\max\{q, 0\}c_s$, whereas there is an additional reaction term $-\min\{q, 0\}c$ on the left-hand side of (2.2a). We refer to Section 4.5 for the possible choices of the functions β and F and for the definition of the parameters \mathbf{S} , \mathbf{w} , and q . Finally, Γ_D , Γ_N , and Γ_R represent respectively the Dirichlet, Neumann, and Robin boundaries, distinct sets covering the fracture system boundary $\partial\mathcal{S}$. We suppose that the measure of the set Γ_D is positive and that $\Gamma_{\text{in}} := \{\mathbf{x} \in \partial\Omega; \mathbf{w} \cdot \mathbf{n} < 0\} \subset \Gamma_D$.

The contaminant transport on the whole system \mathcal{S} (2.1) is given by (2.2a)–(2.2e) for each $l \in L$ and is closed by the requirement of the continuity of the concentration and of the normal component its flux over the interpolygon edges in the following sense: let σ be an edge such that there exist polygons α_i and α_j , $i \neq j$, such that $\sigma = \overline{\alpha_i} \cap \overline{\alpha_j}$. We require

$$c|_{\overline{\alpha_i}} = c|_{\overline{\alpha_j}} \quad \text{on } \sigma \quad \forall i, j \in I_\sigma, \quad (2.3a)$$

$$\sum_{i \in I_\sigma} -\mathbf{S} \nabla c|_{\overline{\alpha_i}} \cdot \mathbf{n}_{\sigma, \alpha_i} = 0 \quad \text{on } \sigma, \quad (2.3b)$$

where $\mathbf{n}_{\sigma, \alpha_i}$ is the unit outward normal vector of the edge σ with respect to the polygon α_i and I_σ is the set of all $i \in L$ such that $\sigma \subset \overline{\alpha_i}$. Note that by the required continuity of the normal component of \mathbf{w} , this equals to imposing the continuity of c and an equilibrated mass balance over the fracture intersections. We refer to [22] for more details.

Finally, we remark that all special cases of (2.2a)–(2.2e) are admissible as well, so that in particular elliptic or parabolic diffusion, convection–diffusion, reaction–diffusion, and convection–diffusion–reaction problems, as well as the respective hyperbolic limits for all the schemes except of the mixed ones, are comprised in FTRANS. One specifies the problem at hand by the parameters in the control and coefficients files, see respectively Sections 4.2 and 4.5 below.

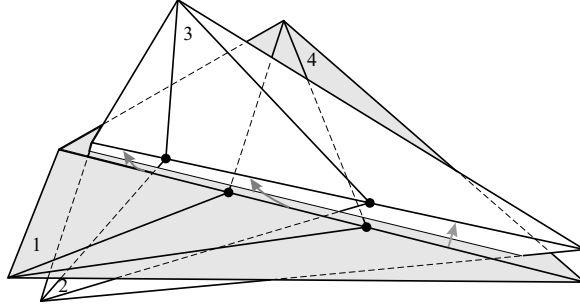


Figure 2: Example of a nonmatching interface between two fractures

3 Numerical schemes

Four different numerical schemes which we shall resume now are implemented in FTRANS; we refer to [18] for a comprehensive survey. For the sake of clarity, all the schemes are written in the fully implicit form. FTRANS however enables a weighted time discretization ranging from the fully implicit to the fully explicit for all the schemes except of the mixed ones, see Section 4.2.6 below. The Newton or inexact Newton method is used for the solution of the associated systems of nonlinear algebraic equations, see Section 4.9.3 below for more details.

3.1 Space and time discretizations

We suppose that the system \mathcal{S} is discretized to a triangular mesh in the following sense: let $\mathcal{T}_{h,\ell}$ denote a matching triangulation of $\alpha_\ell \in \mathcal{S}$, consisting of closed triangles such that $\overline{\alpha_\ell} = \bigcup_{K \in \mathcal{T}_{h,\ell}} K$ and such that if $K, L \in \mathcal{T}_{h,\ell}$, $K \neq L$, then $K \cap L$ is either an empty set or a common face, edge, or vertex of K and L . \mathcal{T}_h is then a union of triangulations of all α_ℓ , see Figure 1 for an example. We however allow as well for triangulations leading to nonmatching interfaces between the fractures. This means that the neighboring edges of two triangles in two different fractures do not have to coincide in the three-dimensional geometrical sense; an element edges correspondence is sufficient, see Figure 2 for an example of a nonmatching interface. More details on this subject are given in Section 4.3 below and in [13]. Finally, for the finite volume scheme, all the triangular elements of the mesh have to have all angles smaller than 90 degrees. We suppose the partition of the time interval $(0, T)$ such that $0 = t_0 < \dots < t_n < \dots < t_N = T$ and define $\Delta t_n := t_n - t_{n-1}$.

3.2 The finite volume scheme

The fully implicit finite volume scheme for the problem (2.2a)–(2.3b) reads (cf. [8] for the standard two- or three-dimensional case): find the values c_K^n , $K \in \mathcal{T}_h$, $n \in \{0, 1, \dots, N\}$, such that

$$c_K^0 = \frac{1}{|K|} \int_K c_0(\mathbf{x}) \, d\mathbf{x} \quad K \in \mathcal{T}_h, \quad (3.1a)$$

$$\begin{aligned} & \frac{\beta(c_K^n) - \beta(c_K^{n-1})}{\Delta t_n} |K| - \sum_{L \in \mathcal{N}(K)} \mathbb{S}_{K,L}^n (c_L^n - c_K^n) + \sum_{L \in \mathcal{N}(K)} w_{K,L}^n \overline{c_{K,L}^n} + F(c_K^n) |K| \\ & + \sum_{\sigma \subset \partial K \cap \Gamma_D} \left\{ -\mathbb{S}_{K,\sigma}^n ((u_D)_\sigma^n - c_K^n) + w_{K,\sigma}^n \overline{c_{K,\sigma}^n} \right\} + \sum_{\sigma \subset \partial K \cap \Gamma_N} \left\{ (u_N)_\sigma^n |\sigma| + w_{K,\sigma}^n c_K^n \right\} \\ & + \sum_{\sigma \subset \partial K \cap \Gamma_R} (u_R)_\sigma^n |\sigma| = q_K^n |K| \quad K \in \mathcal{T}_h, \, n \in \{1, \dots, N\}, \end{aligned} \quad (3.1b)$$

where $\mathcal{N}(K)$ is the set of all the neighboring triangles of the triangle K (they may be two and more through the same edge, cf. Figure 1),

$$\mathbb{S}_{K,L}^n = S_{K,L}^n \frac{|\sigma_{K,L}| + |\sigma_{L,K}|}{2(d_{K,\sigma_{K,L}} + d_{L,\sigma_{L,K}})},$$

where only scalar diffusion–dispersion tensors, $\mathbf{S}|_K^n = S_K^n Id$, are allowed in the finite volume scheme and where $S_{K,L}^n$ is the harmonic mean of S_K^n and S_L^n , $\sigma_{K,L} \subset \partial K$ is the edge corresponding to $\sigma_{L,K} \subset \partial L$ and $|\sigma_{K,L}|, |\sigma_{L,K}|$ are their lengths, $d_{K,\sigma_{K,L}}$ is the distance of the circumcenter of K to its edge $\sigma_{K,L}$, $w_{K,L}^n$ is the mean of the flux of \mathbf{w} from K to L over time interval $(t_{n-1}, t_n]$ (recall that by the continuity of the normal trace of \mathbf{w} over the inter-fracture boundaries, $w_{K,L}^n = w_{L,K}^n$),

$$\overline{c_{K,L}^n} := \begin{cases} c_K^n + \alpha_{K,L}^n (c_L^n - c_K^n) & \text{if } w_{K,L}^n \geq 0 \\ c_L^n + \alpha_{K,L}^n (c_K^n - c_L^n) & \text{if } w_{K,L}^n < 0 \end{cases} \quad (3.2)$$

with $\alpha_{K,L}^n$ the coefficient of the amount of upstream weighting depending on the local ratio of convection and diffusion (in order to only add the minimal necessary numerical diffusion necessary to guarantee the stability of the scheme in the convection-dominated case) by

$$\alpha_{K,L}^n := \frac{\max \left\{ \min \left\{ \mathbb{S}_{K,L}^n, \frac{1}{2} |w_{K,L}^n| \right\}, 0 \right\}}{|w_{K,L}^n|} \text{ if } w_{K,L}^n \neq 0, \quad \alpha_{K,L}^n := 0 \text{ if } w_{K,L}^n = 0,$$

$\mathbb{S}_{K,\sigma}^n = S_K^n |\sigma| / d_{K,\sigma}$, $(u_D)_\sigma^n$ is the mean value of the Dirichlet boundary condition u_D over an edge $\sigma \subset \partial K \cap \Gamma_D$ and the time interval $(t_{n-1}, t_n]$ and similarly for $(u_N)_\sigma^n$ and $(u_R)_\sigma^n$, $w_{K,\sigma}^n$ is the mean of the flux of \mathbf{w} through $\sigma \subset \partial K$ over time interval $(t_{n-1}, t_n]$, $\overline{c_{K,\sigma}^n}$ is defined similarly as $\overline{c_{K,L}^n}$ with $(u_D)_\sigma^n$ instead of c_L^n , and finally q_K^n is the mean value of the sources q over K and the time interval $(t_{n-1}, t_n]$.

3.2.1 A posteriori error estimates

A posteriori error estimates for the linear stationary version of (2.2a)–(2.2e) on a single fracture, discretized by the above finite volume scheme, have been derived in [21]. They are implemented in FTRANS. As FTRANS is not an adaptive code, the a posteriori error estimates can only be used by external applications to refine/coarsen the mesh where indicated. However, the derived estimates do not contain any undetermined multiplicative constants, so that one can immediately use them to reliably control the actual error in the approximate numerical solution.

3.3 The mixed/condensed mixed finite element schemes

In mixed finite element schemes, one approximates simultaneously the unknown scalar concentration c_h as well as its full flux $\mathbf{u}_h \approx -\mathbf{S}\nabla c_h + c_h \mathbf{w}$ or its diffusive flux $\mathbf{u}_h \approx -\mathbf{S}\nabla c_h$. We use here the lowest-order Raviart–Thomas mixed finite element method, see [15, 3, 16] for elliptic diffusion problems and [6, 7] for convection–diffusion–reaction problems. In this case c_h is constant on each $K \in \mathcal{T}_h$ and \mathbf{u}_h belongs to the Raviart–Thomas space of elementwise linear vector functions $\mathbf{RT}^0(\mathcal{T}_h)$ with one unknown per side (the normal flux of \mathbf{u}_h through this side). The space $\mathbf{RT}_{N,R}^0(\mathcal{T}_h)$ with zero normal flux through Γ_N and Γ_R (with no unknowns on the sides from Γ_N and Γ_R) is also used in the sequel. A proper definition of these spaces for fracture networks (assuring the continuity of the normal trace in the sense of (2.3b)) is given in [22]; in particular, the number of basis functions for one side is equal to the number of connected elements minus one for fracture

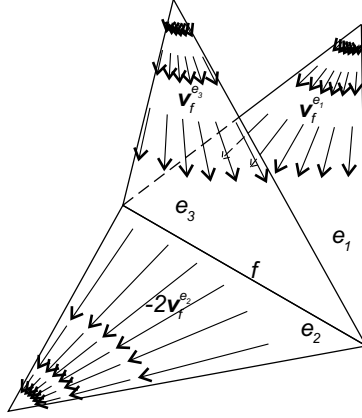


Figure 3: One of the two velocity basis function for three elements sharing the same edge

networks, see Figure 3 for an example of one of the two velocity basis function for three elements sharing the same edge.

Three different variants of this mixed finite element method are at disposal in FTRANS, differing in the discretization of the convection term. For each of these variants, local elimination of the flux unknowns \mathbf{u}_h in the standard two- or three-dimensional case is possible, leading to matrix problems with only the scalar unknowns, see [17, 18, 19]. We call the resulting schemes condensed mixed finite element schemes.

3.3.1 Centered scheme with vector unknowns representing full flux

This is the classical centered mixed finite element scheme for convection–diffusion–reaction problems where $\mathbf{u}_h \approx -\mathbf{S}\nabla c_h + c_h \mathbf{w}$. For the problem (2.2a)–(2.3b), it reads as follows (see [19, Section 4] for the standard two- or three-dimensional case): for all $n \in \{1, \dots, N\}$, find c_h^n given by c_K^n , $K \in \mathcal{T}_h$, and the vector functions $\mathbf{u}_h^n \in \mathbf{RT}^0(\mathcal{T}_h)$ satisfying $\langle \mathbf{u}_h^n \cdot \mathbf{n}, 1 \rangle_\sigma = (u_R)_\sigma^n |\sigma|$ for all $\sigma \subset \Gamma_R$ (this corresponds to the fulfillment of the Robin boundary condition (2.2e)) and $\langle \mathbf{u}_h^n \cdot \mathbf{n}, 1 \rangle_\sigma - w_{K,\sigma}^n c_K^n = (u_N)_\sigma^n |\sigma|$ for all $\sigma \subset \partial K \cap \Gamma_N$ for some $K \in \mathcal{T}_h$ (this corresponds to the fulfillment of the Neumann boundary condition (2.2d)) such that

$$(\mathbf{S}^{-n} \mathbf{u}_h^n, \mathbf{v}_h)_S - (\nabla \cdot \mathbf{v}_h, c_h^n)_S - (c_h^n \mathbf{w}^n, \mathbf{S}^{-n} \mathbf{v}_h)_S = - \sum_{\sigma \subset \Gamma_D} \langle \mathbf{v}_h \cdot \mathbf{n}, (u_D)_\sigma^n \rangle_{\sigma \cap \Gamma_D} \quad \forall \mathbf{v}_h \in \mathbf{RT}_{N,R}^0(\mathcal{T}_h), \quad (3.3a)$$

$$\left(\frac{\beta(c_K^n) - \beta(c_K^{n-1})}{\Delta t_n}, \phi_K \right)_K + (\nabla \cdot \mathbf{u}_h^n, \phi_K)_K + (F(c_K^n), \phi_K)_K = (q, \phi_h)_K \quad \forall K \in \mathcal{T}_h, \quad (3.3b)$$

where we have used the notation from the previous section and where \mathbf{S}^{-n} and \mathbf{w}^n are, respectively, the means of \mathbf{S}^{-1} and \mathbf{w} over the time interval $(t_{n-1}, t_n]$ and ϕ_K is the characteristic function of the element $K \in \mathcal{T}_h$. The initial condition is discretized by (3.1a).

3.3.2 Weighted upwind-mixed finite element scheme

The above centered scheme has the disadvantage of being unstable in the convection-dominated case. An alternative is to use a finite volume upwind-type discretization of the convection term to obtain an upwind-mixed finite element scheme, cf. [11, 4, 5]. Approximating this time $\mathbf{u}_h \approx -\mathbf{S}\nabla c_h$ and using the ideas of the local Péclet upstream weighting (3.2), we seek (cf. [19, Section 4],

[20, Section 3.3] for the standard two- or three-dimensional case), for all $n \in \{1, \dots, N\}$, the concentration c_h^n given by c_K^n , $K \in \mathcal{T}_h$, and the vector functions $\mathbf{u}_h^n \in \mathbf{RT}^0(\mathcal{T}_h)$ satisfying $\langle \mathbf{u}_h^n \cdot \mathbf{n}, 1 \rangle_\sigma = (u_N)_\sigma^n |\sigma|$ for all $\sigma \subset \Gamma_N$ (this corresponds to the fulfillment of the Neumann boundary condition (2.2d)) and $\langle \mathbf{u}_h^n \cdot \mathbf{n}, 1 \rangle_\sigma + w_{K,\sigma}^n c_K^n = (u_R)_\sigma^n |\sigma|$ for all $\sigma \subset \partial K \cap \Gamma_R$ for some $K \in \mathcal{T}_h$ (this corresponds to the fulfillment of the Robin boundary condition (2.2e)) such that

$$(\mathbf{S}^{-n} \mathbf{u}_h^n, \mathbf{v}_h)_S - (\nabla \cdot \mathbf{v}_h, c_h^n)_S = - \sum_{\sigma \subset \Gamma_D} \langle \mathbf{v}_h \cdot \mathbf{n}, (u_D)_\sigma^n \rangle_{\sigma \cap \Gamma_D} \quad \forall \mathbf{v}_h \in \mathbf{RT}_{N,R}^0(\mathcal{T}_h), \quad (3.4a)$$

$$\begin{aligned} & \left(\frac{\beta(c_K^n) - \beta(c_K^{n-1})}{\Delta t_n}, \phi_K \right)_K + (\nabla \cdot \mathbf{u}_h^n, \phi_K)_K + \sum_{L \in \mathcal{N}(K)} w_{K,L}^n \overline{c_{K,L}^n} + (F(c_K^n), \phi_K)_K \\ & + \sum_{\sigma \subset \partial K \cap \Gamma_D} w_{K,\sigma}^n \overline{c_{K,\sigma}^n} + \sum_{\sigma \subset \partial K \cap (\Gamma_N \cup \Gamma_R)} w_{K,\sigma}^n c_K^n = (q, \phi_K)_K \quad \forall K \in \mathcal{T}_h, \end{aligned} \quad (3.4b)$$

where $\overline{c_{K,L}^n}$ is given by (3.2) with either $\alpha_{K,L}^n$ given by user or by

$$\alpha_{K,L}^n := \min \left\{ \frac{\mathbb{S}_{K,L}^n}{|w_{K,L}^n|}, \frac{1}{2} \right\} \quad \text{if } w_{K,L}^n \neq 0, \quad \alpha_{K,L}^n := 0 \quad \text{if } w_{K,L}^n = 0, \quad (3.5)$$

where this time $\mathbb{S}_{K,L}^n$ is the harmonic average of the smallest eigenvalues of \mathbf{S}^n , the mean of \mathbf{S} over the time interval $(t_{n-1}, t_n]$, on K and L . For Dirichlet boundary sides, $\overline{c_{K,\sigma}^n}$ is defined similarly. The initial condition is discretized by (3.1a).

3.3.3 Combination of a centered scheme with vector unknowns only representing diffusive flux and of the upwind-mixed scheme

Another centered mixed finite element scheme for convection–diffusion–reaction problems corresponds to the choice $\mathbf{u}_h \approx -\mathbf{S}\nabla c_h$. For the problem (2.2a)–(2.3b), it reads as follows (see [20, Sections 3.2 and 5.5] for the standard two- or three-dimensional case): for all $n \in \{1, \dots, N\}$, find c_h^n given by c_K^n , $K \in \mathcal{T}_h$, and the vector functions $\mathbf{u}_h^n \in \mathbf{RT}^0(\mathcal{T}_h)$ satisfying $\langle \mathbf{u}_h^n \cdot \mathbf{n}, 1 \rangle_\sigma = (u_N)_\sigma^n |\sigma|$ for all $\sigma \subset \Gamma_N$ (this corresponds to the fulfillment of the Neumann boundary condition (2.2d)) and $\langle \mathbf{u}_h^n \cdot \mathbf{n}, 1 \rangle_\sigma + w_{K,\sigma}^n c_K^n = (u_R)_\sigma^n |\sigma|$ for all $\sigma \subset \partial K \cap \Gamma_R$ for some $K \in \mathcal{T}_h$ (this corresponds to the fulfillment of the Robin boundary condition (2.2e)) such that

$$(\mathbf{S}^{-n} \mathbf{u}_h^n, \mathbf{v}_h)_S - (\nabla \cdot \mathbf{v}_h, c_h^n)_S = - \sum_{\sigma \subset \Gamma_D} \langle \mathbf{v}_h \cdot \mathbf{n}, (u_D)_\sigma^n \rangle_{\sigma \cap \Gamma_D} \quad \forall \mathbf{v}_h \in \mathbf{RT}_{N,R}^0(\mathcal{T}_h), \quad (3.6a)$$

$$\begin{aligned} & \left(\frac{\beta(c_K^n) - \beta(c_K^{n-1})}{\Delta t_n}, \phi_K \right)_K + (\nabla \cdot \mathbf{u}_h^n, \phi_K)_K - (\mathbf{S}^{-n} \mathbf{u}_h^n \mathbf{w}^n, \phi_K)_K + (F(c_K^n) + \nabla \cdot \mathbf{w}, \phi_K)_K \\ & = (q, \phi_h)_K \quad \forall K \in \mathcal{T}_h. \end{aligned} \quad (3.6b)$$

The initial condition is again discretized by (3.1a). The advantage over the centered scheme (3.3a)–(3.3b) is that this centered discretization can easily be combined with the (full-upwind variant of the) scheme (3.4a)–(3.4b). Using the parameter (3.5) for determining the amount of the upstream weighting, we now have a scheme that smoothly changes from a centered mixed finite element one to an upwind-mixed one, guaranteeing optimal accuracy and stability in both the convection and diffusion-dominated limits.

3.3.4 Algebraic condensation to only scalar unknowns

The systems (3.3a)–(3.3b), (3.4a)–(3.4b), or (3.6a)–(3.6b), linearized by the Newton method, can be written in the matrix form as

$$\begin{pmatrix} \mathbb{A} & \mathbb{C} \\ \mathbb{B} & \mathbb{D} \end{pmatrix} \begin{pmatrix} U \\ P \end{pmatrix} = \begin{pmatrix} F \\ G \end{pmatrix}.$$

We have proposed in [17, 18, 19] an approach allowing for local elimination of the flux unknowns U from the above system for standard two- or three-dimensional domains leading to a problem of the form

$$\mathbb{E}P = H$$

with a sparse matrix \mathbb{E} . The mixed finite element schemes rewritten in this way are hereafter termed condensed mixed finite element schemes.

3.3.5 A posteriori error estimates

A posteriori error estimates for the linear stationary version of (2.2a)–(2.2e) on a single fracture, discretized by the schemes (3.4a)–(3.4b) or (3.6a)–(3.6b), have been derived in [20]. They are implemented in FTRANS. As FTRANS is not an adaptive code, the a posteriori error estimates can only be used by external applications to refine/coarsen the mesh where indicated. However, the derived estimates do not contain any undetermined multiplicative constants, so that one can immediately use them to reliably control the actual error in the approximate numerical solution.

3.4 The combined finite volume–finite element scheme

The combined finite volume–finite element scheme is defined as the combined finite volume–nonconforming finite element scheme below, with the difference that the dual volumes are associated with the vertices in place of the edges, see [9] for the details in the standard two- or three-dimensional case.

3.5 The combined finite volume–nonconforming finite element scheme

Combined finite volume–finite element schemes use a dual partition \mathcal{D}_h of \mathcal{S} in addition to the triangulation \mathcal{T}_h . In the combined finite volume–nonconforming finite element scheme, there is one dual element D associated with each side σ_D of \mathcal{T}_h . We construct it by connecting the barycentres of every $K \in \mathcal{T}_h$ that contains σ_D through the vertices of σ_D . For $\sigma_D \in \mathcal{E}_h^{\text{ext}}$, the contour of D is completed by the side σ_D itself. Note that σ_D may be multiply shared; see Figure 4 for an example of the dual mesh in this case. For nonmatching interfaces, the dual volume has the above-defined support in all the triangles connected via the element–edges correspondence. We denote by Q_D the barycentre of the side σ_D , by $\mathcal{D}_h^{\text{int}, \Gamma_{\text{N}, \text{R}}}$ the set of all interior dual volumes and of dual volumes corresponding to edges from Γ_{N} or Γ_{R} ; $\mathcal{D}_h^{\Gamma_{\text{D}}}$ are then the dual volumes corresponding to edges from Γ_{D} . Finally, as for the finite volume scheme, $\mathcal{N}(D)$ is the set of all the neighboring dual volumes to D . Note that in contrast to the finite volume case, they may never be two and more neighbors through one edge of the dual volume, since this always lies in the interior of some triangle.

The fully implicit combined finite volume–nonconforming finite element scheme for the problem (2.2a)–(2.3b) then reads (cf. [10] for standard two- or three-dimensional domains): find the

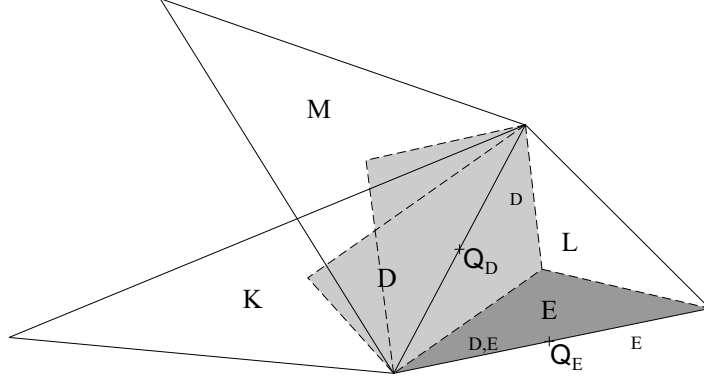


Figure 4: Triangles $K, L, M \in \mathcal{T}_h$ connected through one side and the associated dual volume $D \in \mathcal{D}_h^{\text{int}, \Gamma_{N,R}}$ together with its neighbor $E \in \mathcal{D}_h^{\Gamma_D}$

values c_D^n , $D \in \mathcal{D}_h$, $n \in \{0, 1, \dots, N\}$, such that

$$c_D^0 = \frac{1}{|D|} \int_D c_0(\mathbf{x}) \, d\mathbf{x} \quad D \in \mathcal{D}_h, \quad (3.7a)$$

$$c_D^n = (u_D)_{\sigma_D}^n \quad D \in \mathcal{D}_h^{\Gamma_D}, \, n \in \{1, \dots, N\}, \quad (3.7b)$$

$$\begin{aligned} & \frac{\beta(c_D^n) - \beta(c_D^{n-1})}{\Delta t_n} |D| - \sum_{E \in \mathcal{N}(D)} \mathbb{S}_{D,E}^n (c_E^n - c_D^n) + \sum_{E \in \mathcal{N}(D)} w_{D,E}^n \overline{c_{D,E}^n} + F(c_D^n) |D| \\ & + (u_N)_{\sigma_D \cap \Gamma_N}^n |\sigma_D \cap \Gamma_N| + w_{D, \sigma_D \cap \Gamma_N}^n c_D^n + (u_R)_{\sigma_D \cap \Gamma_R}^n |\sigma_D \cap \Gamma_R| \\ & = q_D^n |D| \quad D \in \mathcal{D}_h^{\text{int}, \Gamma_{N,R}}, \, n \in \{1, \dots, N\}, \end{aligned} \quad (3.7c)$$

where

$$\mathbb{S}_{D,E}^n := - \sum_{K \in \mathcal{T}_h} (\mathbf{S}^n \nabla \varphi_E, \nabla \varphi_D)_K \quad D, E \in \mathcal{D}_h, \, n \in \{1, \dots, N\}, \quad (3.8)$$

with \mathbf{S}^n being the mean of \mathbf{S} over the time interval $(t_{n-1}, t_n]$ and φ_D the basis functions of the nonconforming finite element method (φ_D is linear on each $K \in \mathcal{T}_h$, equal to 1 at Q_D and to 0 at all Q_E for $D \neq E$), $w_{D,E}^n$ is the mean of the flux of \mathbf{w} from D to E over time interval $(t_{n-1}, t_n]$, $\overline{c_{D,E}^n}$ is given similarly as (3.2), and finally q_D^n is the mean value of the sources q over D and the time interval $(t_{n-1}, t_n]$.

4 Using FTRANS

4.1 Requirements

WINDOWS 3.1/95/98/Millennium/NT/2000/XP with a MS DOS command line and the executable `ftrans.exe` are necessary to run FTRANS. MATLAB may be used as a computational engine namely for the solution of the resulting systems of linear equations, cf. Section 4.2.4 below, or for the CMFE scheme, see Section 4.2.7 below. In this case MATLAB 6.1 or higher has to be installed on the computer. Another choice is to rely on the gm6 solver. Libraries `Dforrt.dll` and `gm6.dll` then have to be in the same directory as `ftrans.exe`. One launches is FTRANS by typing

```
ftrans.exe control_file_name
```

on the command line. In `control_file_name`, the full name of the control file (including the extension) has to be given. The address is either relative with respect to the directory where `FTRANS.exe` is situated, or an absolute path. The control file contains all specifications for FTRANS, see Section 4.2 for the description of its format. The data for FTRANS are then given separately, see Sections 4.3–4.7. After the calculation, results are given in output files, see Section 4.8. Finally, we give in Section 4.9 various important remarks on FTRANS possibilities and limitations.

4.2 Control file format

The control file has the format of a standard windows `*.ini` file, giving always section, key, and value; the extension `*.ini` is not compulsory. For an example of a control file, see Appendix A.

4.2.1 Section [Global]

Section [Global] contains FTRANS basic parameters. The keys and their values are as follows.

Solution_method

Allowed values: Integers 1–5.

Description: Numerical method to be used.

- 1 Finite volume method (FV).
- 2 Mixed finite element method (MFE).
- 3 Condensed mixed finite element method (CMFE).
- 4 Combined finite volume–finite element method (FV–FE).
- 5 Combined finite volume–nonconforming finite element method (FV–NCFE).

Description

Allowed values: One-line text.

Description: A brief description of the given model.

4.2.2 Section [Input]

Section [Input] specifies the names of FTRANS input files (see Section 4.3–4.7 for their format). The keys and their values are as follows.

Space_mesh

Allowed values: One-line text.

Description: Name of the file containing the description of spatial discretization. The format is identical with that of FFLOW of the first generation.

Time_mesh

Allowed values: One-line text.

Description: Name of the file containing the description of the temporal discretization.

Coefficients

Allowed values: One-line text.

Description: Name of the file containing the description of the coefficients.

Initial_conditions

Allowed values: One-line text.

Description: Name of the file containing the description of initial conditions.

Boundary_conditions

Allowed values: One-line text.

Description: Name of the file containing the description of boundary conditions.

4.2.3 Section [Run]

Section [Run] contains some options for runtime behavior of FTRANS. The keys and their values are as follows.

Log_file

Allowed values: One-line text.

Description: Name of the file where FTRANS writes log messages.

Screen_verbosity

Allowed values: Integers 0–7.

Description: Intensity of messages written in the FTRANS run window (0 = no messages, ..., 7 = all messages).

Log_verbosity

Allowed values: Integers 0–7.

Description: Intensity of messages written in the log file (0 = no messages, ..., 7 = all messages).

4.2.4 Section [Solver]

Section [Solver] contains options controlling the solver of linear equations used by FTRANS. Two solvers are now at the disposition. One can either use MATLAB as a computational engine (this supposes that MATLAB is installed on your computer, see Section 4.1 for more details), or the dll-linked solver gm6. The keys and their values are as follows.

Solver

Allowed values: Integers 0–1.

Description: Type of the solver.

0 MATLAB.

1 gm6.

Solver_method

Allowed values: Integers 0–5.

Description: Specification of the solver method (see MATLAB help).

0 Direct LU-based solver.

1 BI-CG.

2 Bi-CGStab.

3 QMR.

4 GMRES.

5 CGS.

Solver_accuracy

Allowed values: Double value > 0 .

Description: Accuracy of the solution of a matrix problem for an iterative method—relative residual (see MATLAB help). Recommended values: $1e-5$ to $1e-8$; they should be of the same order as `Prec_lin` (see Section 4.2.6 below) for nonlinear problems.

Solver_max_iter

Allowed values: Positive integer.

Description: Maximal number of iterations for iterative solvers. For linear problems, the range is roughly 100–1000. For nonlinear problems solved by the inexact Newton method (see Section 4.9.3 below), 5–20 iterations should be optimal.

Preconditioning

Allowed values: Integers 0–4.

Description: Type of preconditioning (see MATLAB help).

0 No preconditioning.

1 LU.

2 LU incomplete.

3 Modified LU incomplete.

4 LU incomplete with zero level fill-in.

Precond_accur

Allowed values: Double value > 0 .

Description: Accuracy of incomplete LU preconditioning (see MATLAB help). Recommended values: $1e-1$ to $1e-5$ (the lower the value, the longer the CPU time but the higher the accuracy of LU factors and the faster solution by the preconditioned iterative method).

Save_matrix

Allowed values: Text **Yes** or **No**.

Description: Whether to save the matrix on each time and linearization step, if using the gm6 solver.

4.2.5 Section [Output]

Section [Output] contains options controlling the output of FTRANS. The keys and their values are as follows.

Running_results

Allowed values: Text **Yes** or **No**.

Description: Whether the results will be written on each intermediate time step.

Final_results

Allowed values: Text **Yes** or **No**.

Description: Whether the final results will be written at the end.

Output_digits

Allowed values: Integers 1–16.

Description: The number of significant digits of the output.

Output_file

Allowed values: One-line text.

Description: Name of the file with the results (see Section 4.8.1 below for the description of its format).

Output_file_a_post

Allowed values: One-line text.

Description: Name of the file with a posteriori error estimates for the MFE/CMFE schemes (see Section 4.8.2 below for the description of its format).

Output_file_immob

Allowed values: One-line text.

Description: Name of the file with results for the immobile zone if `Immobile = Yes`.

POS_file

Allowed values: Text `Yes` or `No`.

Description: Whether make as well `.POS` (GMSH) output files.

4.2.6 Section [Parameters]

Section [Parameters] contains various options of FTRANS. The keys and their values are as follows.

Parabolic

Allowed values: Text `Yes` or `No`.

Description: Whether the problem is parabolic or elliptic.

Evolving_BC

Allowed values: Text `Yes` or `No`.

Description: Whether the boundary conditions are evolving in time (for parabolic problems).

Evolving_sour

Allowed values: Text `Yes` or `No`.

Description: Whether the sources are evolving in time (for parabolic problems).

Evolving_coef

Allowed values: Text `Yes` or `No`.

Description: Whether the coefficients are evolving in time (for parabolic problems).

Time_discr_par

Allowed values: Double value 0–1.

Description: Parameter of the discretization in time, ranging from fully implicit (0) through Crank–Nicholson (0.5) to explicit (1), see Section 4.9.1 below for the discussion. For the MFE/CMFE schemes, only fully implicit discretization is at disposal.

Nonlinear

Allowed values: Text `Yes` or `No`.

Description: Whether the problem is nonlinear or linear.

Prec_lin

Allowed values: Double value > 0 .

Description: Precision of the Newton linearization procedure for nonlinear problems—relative

error in `Lin_error_norm`. Recommended values: 1e-5 to 1e-8; they should be of the same order as `Solver_accuracy` (see Section 4.2.4) if an iterative solver is used.

`Max_iter_lin`

Allowed values: Positive integer.

Description: Maximal number of iterations of the Newton linearization. Recommended values: virtually unlimited.

`Lin_error_norm`

Allowed values: Integers 1–2.

Description: Type of the norm for the measurement of the relative error in Newton linearization.

1 L_2 .

2 Maximum.

`Dif_disp_by_cfs`

Allowed values: Text `Yes` or `No`.

Description: Whether the diffusion–dispersion tensor is given by coefficients or directly (see Section 4.5 below).

`Sour_inj_pump_div`

Allowed values: Text `Yes` or `No`.

Description: Whether the sources are divided to injections with prescribed concentration and pumping without prescribed concentration or no, see Section 2.2.

`Immobile`

Allowed values: Text `Yes` or `No`.

Description: Whether the model will be a dual-porosity one, containing both mobile and immobile zones.

`MIM_coupled`

Allowed values: Text `Yes` or `No`.

Description: Whether there will be separate matrix representations of the mobile and immobile parts (if `Immobile` = `Yes`).

4.2.7 Section [Particular]

Section [Particular] contains some particular options for FTRANS run. The keys and their values are as follows.

`Type_of_linearization`

Allowed values: Integers 0–1.

Description: Type of the linearization of the nonlinear problem.

0 Linearization by the Newton method of the original numerical scheme. Works if $\beta' \neq \infty$, $\beta' = 0$ is allowed.

1 Linearization by the Newton method for a new unknown $u_h = \beta(c_h)$. Works even if $\beta' = \infty$ but $\beta' = 0$ is forbidden. For the moment at the disposition only for the FV–NCFE scheme if `Adsorption_function` = 11 or `Adsorption_function` = 12 and if the first of `Adsorption_parameters` is either 0.5 or 2, see Section 4.5 below.

LU_fact_only_first

Allowed values: Text **Yes** or **No**.

Description: Whether to make the LU factorization only on the first time step (linear problems) or linearization step (nonlinear problems). See Section 4.9.4 below for more details.

GMRES_restart_iter

Allowed values: Positive integer.

Description: The number of iterations after which the GMRES method will be restarted.

Mixed_conv_type

Allowed values: Integers 1–3.

Description: specification of the MFE/CMFE scheme, see Section 3.3.

- 1 Means the centered mixed finite element scheme (3.3a)–(3.3b). Precise for diffusion-dominated problems but oscillates in convection-dominated problems.
- 2 Means the weighted upwind-mixed finite element scheme (3.4a)–(3.4b). May be less precise for diffusion-dominated problems but does not oscillate in convection-dominated problems.
- 3 Means the combination of the centered and upwind-mixed finite element scheme (3.6a)–(3.6b). A compromise of the two above schemes, behaving well in both convection and diffusion-dominated limits. A recommended alternative to use.

Mixed_upstr_weigh_par

Allowed values: Double value 0–0.5 or -1.

Description: Means the amount of upstream weighting in the MFE/CMFE schemes with `Mixed_conv_type = 2` or `Mixed_conv_type = 3`.

0 Means full upstream weighting.

0.5 Means centered weighting.

-1 The amount of upstream weighting is defined by the parameter (3.5) in the dependence on the local ratio between convection and diffusion. Recommended value.

Mixed_column_ordering

Allowed values: Text **Yes** or **No**.

Description: Ordering of columns of matrices resulting from the MFE scheme so that one could use, in connection with preconditioning, some of the proposed iterative methods. Without ordering and preconditioning, the proposed iterative methods will not (sufficiently rapidly) converge because the matrices resulting from the MFE scheme (in contrast to the CMFE scheme) are of the saddle-point type. So for the MFE scheme, one should either use a direct solver, or always put `Preconditioning = 2` or `Preconditioning = 3` or `Preconditioning = 4` and `Mixed_column_ordering = 1`.

Mixed_alg_cond

Allowed values: Integers 1–4.

Description: Type of algebraic condensation for the CMFE scheme.

1–2 Works only for elliptic diffusion problems in the planar case.

3 Works for nonlinear parabolic reaction–diffusion problems in the planar case.

- 4 No restrictions on problems in the planar case. Needs however MATLAB as a computational engine. The file `matr_A_inv_J.m` has to be in some directory searched by MATLAB. This version is strongly recommended.

Mixed_alg_cond_cnd_nb

Allowed values: Text `Yes` or `No`.

Description: Whether to evaluate the condition number of the local condensation matrices. Important in order to check whether the algebraic condensation works correctly. Not to use for repeated runs on the same problem in order to save the CPU time. Only works for `Mixed_alg_cond = 4`.

Postprocessing

Allowed values: Integers 0–1.

Description: Postprocessing of the results for the FV/MFE/CMFE schemes.

0 No postprocessing, piecewise constant result.

1 Postprocessed piecewise linear (generally discontinuous) result in addition to the original piecewise constant one. This result should be more precise.

A_posteriori_err_est

Allowed values: Integers 0–1.

Description: A posteriori error estimates for the MFE/CMFE schemes.

0 No estimates will be computed.

1 A posteriori error estimates will be computed, see Sections 3.2.1 and 3.3.5. The results will be saved in the `Output_file_a_post` file, see Section 4.8.2 below for the description of its format.

4.3 Mesh file format

The mesh file format is identical with that of FFLOW of the first generation. This is also the reason why the coefficients of hydraulic resistance in each element are listed below; these values are not compulsory. All the lengths have to be given in a unique [L] unit. For an example of a mesh file, see Appendix B.1; the actual format is as follows:

First line: Compulsory heading `[FRACTURE_FLOW_MESH]`

Second line: `Mesh identification number (int)`

Third line: `Dimensions of the simulated domain (3×double)`

The data of individual fractures follow. They always begin by the line `Fracture{` and end by the line `}Fracture`. The structure between these two lines is as follows:

`id number of the fracture (int), type (int), set according to orientation (int)`

`coordinates of the center of the fracture (3×double)`

`radius of the fracture (double)`

`normal vector of the fracture (3×double)`

`hydraulic conductivity exponent (double)`

`local coordinates origin in global (3×double) and local coordinates (2×double)`

`another point in global (3×double) and local coordinates (2×double)`

direction vector of the local coordinate axis x (3×double)
direction vector of the local coordinate axis y (3×double)

Next the nodes of the mesh in the given fracture follow, beginning by the line `Nodes{` and ending by the line `}Nodes`. Each line between these two lines contains the data of one node in the following format:

`lid x2d y2d`

Description:

`lid` (int) Id of the node.

`x2d` (double) The x coordinate in the local coordinates system.

`y2d` (double) The y coordinate in the local coordinates system.

Next the elements of the mesh in the given fracture follow, beginning by the line `{Elements` and ending by the line `}Elements`. Each line between these two lines contains the data of one element in the following format:

`lid nod0 nod1 nod2 coef1 coef2 coef3`

Description:

`lid` (int) Id number of the element in the fracture.

`nod0, nod1, nod2` (3×int) Id number of the nodes of the element.

`coef1, coef2, coef3` (3×double) Coefficients of the hydraulic resistance in the element.

Next the neighboring between the elements of the fracture at hand and elements in other fractures follows, beginning by the line `Neighbors{` and ending by the line `}Neighbors`. Each line between these two lines contains the data of one neighbor in the following format:

`my_fid my_eid my_sid fid eid sid`

Description:

`my_fid` (int) Id number of the given fracture.

`my_eid` (int) Id number of the element in fracture `my_fid`.

`my_sid` (int) Number of the side of element `my_eid`.

`fid` (int) Id number of the neighboring fracture.

`eid` (int) Id number of the neighboring element in fracture `fid`.

`sid` (int) Number of the neighboring side in element `eid`.

We finally recall that the neighboring edges do not have to coincide in the three-dimensional geometrical sense; the above-given element edges correspondence is sufficient (cf. Figure 2). See [13] for more details.

4.4 Time file format

The time file is composed of a compulsory heading and three (four) blocks as follows (all times have to be given in a unique [T] unit):

First line: Compulsory heading `[FRACTURE_TRANSPORT_TIME_MESH]`

Key: `Time_interval`

Allowed values: Positive real number T .

Description: The time interval is $(0, T)$.

Key: `Number_time_steps`

Allowed values: Positive integer N .

Description: Number of time levels.

Key: `Equidistant`

Allowed values: Integers 0–1.

Description: Whether the time interval is not or is divided to N steps of equal length.

Key: `Time_mesh`

Line in the following format:

```
t_1 t_2 ... t_N
```

Description:

`t_n` (double) Length of n -th time interval. The sum must equal to T . Only compulsory when `Equidistant` = 0.

4.5 Coefficients file format

The coefficients file is composed of several blocks following a compulsory heading. We first demonstrate the default format of the blocks for sources and then describe the other blocks.

First line: Compulsory heading [`FRACTURE_TRANSPORT_COEFFICIENTS`]

Key: `Sources`

Follows a compulsory line in the following format:

```
default sources_1 ... sources_N
```

Allowed values for `sources_n`: Double value.

Description: Default source per unit volume [$\text{ML}^{-2}\text{T}^{-1}$]. For elliptic problems or in the case where `Evolving_sour` = No, only the sources on the first time level (`sources_1`) are considered. The same holds in the sequel for the other coefficients which may depend on time for elliptic problems or when `Evolving_coef` = No.

Follow optional blocks only containing the elements/sides/nodes where the sources are different from the default ones. `FTRANS` only searches for the appropriate format according to the numerical method used:

The schemes `FV/MFE/CMFE`. Compulsory first line `FV/MFE/CMFE{`, last line `}FV/MFE/CMFE`. Each line between these two lines describes the sources for one element. The format is the following:

```
fid eid sources_1 ... sources_N
```

Description:

`fid` (int) Id number of the fracture.

`eid` (int) Id number of the element.

`sources_n` (double) Sources on n -th time interval.

The scheme `FV-NCFE`. Compulsory first line `FV-NCFE{`, last line `}FV-NCFE`. Each line between these two lines describes the sources for one side. The format is the following:

```
fid eid sid sources_1 ... sources_N
```

Description:

`fid` (int) Id number of the fracture.

`eid` (int) Id number of the element.

sid (int) Id number of the side of the element (0, 1, 2).

sources_n (double) Sources on n -th time interval.

Since in this way, sources on one side may be described more than once for a side shared by two or more elements, it is necessary that in this case all the given values coincide.

The scheme FV-FE. Compulsory first line **FV-FE{**, last line **}FV-FE**. Each line between these two lines describes the sources for one node. The format is the following:

```
fid nid sources_1 ... sources_N
```

Description:

fid (int) Id number of the fracture.

nid (int) Id number of the node.

sources_n (double) Sources on n -th time interval.

Since in this way, sources on one node may be described more than once for a node shared by two or more fractures, it is necessary that in this case all the given values coincide.

We now give a complete list of all the coefficients required. Except for the diffusion–dispersion tensor and the velocity field, the format is as that for the sources, i.e. with (a) compulsory default value(s) and a list of elements/sides/nodes with different values.

Key: Adsorption_function

Next compulsory line has the following format (no time-dependency):

```
default ads_fn
```

Allowed values for **ads_fn**: Integers 0–2, 11–12.

Description: Type of the adsorption function (of the function β); the unit of $\beta(c)$ is $[\text{ML}^{-2}]$.

0 No adsorption: $\beta(c) = \theta c$, where θ is the water content **wtr_cnt**.

1 Freundlich: $\beta(c) = \theta c + \rho_b a_1 c^{a_2}$, where θ is the water content **wtr_cnt**, ρ_b is the bulk density **bulk_dens**, a_1 is the first adsorption parameter **ads_par_1**, and a_2 is the second adsorption parameter **ads_par_2**.

2 Langmuir: $\beta(c) = \theta c + \rho_b a_1 a_2 c / (1 + a_1 c)$, where θ is the water content **wtr_cnt**, ρ_b is the bulk density **bulk_dens**, a_1 is the first adsorption parameter **ads_par_1**, and a_2 is the second adsorption parameter **ads_par_2**.

11 Academic 1: $\beta(c) = c + c^{a_1}$, where a_1 is the first adsorption parameter **ads_par_1**.

12 Academic 2: $\beta(c) = c^{a_1}$, where a_1 is the first adsorption parameter **ads_par_1**.

Key: Adsorption_parameters

Next compulsory line has the following format (no time-dependency):

```
default ads_par_1 ads_par_2 ads_par_im,1 ads_par_im,2
```

Allowed values for **ads_par_n**, **ads_par_im,n**: Nonnegative double values.

Description: Parameters for the adsorption function. Parameters **ads_par_im,n** are only compulsory when **Immobile** = **Yes**. The unit for **ads_par_1**, **ads_par_im,1** is $[\text{L}^2\text{M}^{-1}]$, **ads_par_2** and **ads_par_im,2** are dimensionless [-] for the Freundlich and Langmuir adsorption functions.

Key: Reaction_function

Next compulsory line has the following format (no time-dependency):

```
default react_fn
```

Allowed values for `react_fn`: Integers 0–2, 11–14.

Description: Type of the reaction function F ; the unit of $F(c)$ is $[\text{ML}^{-2}\text{T}^{-1}]$.

0 No reaction: $F(c) = 0$.

1 Radioactive decay: $F(c) = r_1\beta(c)$, where r_1 is the first reaction parameter `react_par_1`.

2 Reaction 2: $F(c) = r_1c^{r_2}$, where r_1 is the first reaction parameter `react_par_1` and r_2 is the second reaction parameter `react_par_2`.

11 Academic 1: $F(c) = c$.

12 Academic 2: $F(c) = c^{r_1}$, where r_1 is the first reaction parameter `react_par_1`.

13 Academic 3: $F(c) = r_1c^{r_1}$, where r_1 is the first reaction parameter `react_par_1`.

14 Academic 4: $F(c) = 3c + r_1c^{r_1}$, where r_1 is the first reaction parameter `react_par_1`.

Key: Reaction_parameters

Next compulsory line has the following format (no time-dependency):

```
default react_par_1 react_par_2
```

Allowed values for `react_par_n`: Nonnegative double values.

Description: Parameters for the reaction function. For radioactive decay, the unit of `react_par_1` is $[\text{T}^{-1}]$.

The diffusion–dispersion tensor can be given in two ways: either directly by its components in block `Diffusion_dispersion_tensor` (when `Dif_disp_by_cfs = No`), or by diffusion–dispersion coefficients in section `Diffusion_dispersion_coefficients` (when `Dif_disp_by_cfs = Yes`). It is only given (as well as the following velocity field) for elements for all the schemes.

Key: Diffusion_dispersion_tensor

Next compulsory line has the following format:

```
default coef_11_1 coef_22_1 coef_12_1 ... coef_11_N coef_22_N coef_12_N
```

Allowed values for `coef_jk_n`: Double values.

Description: The diffusion–dispersion tensor \mathbf{S}^n $[\text{L}^2\text{T}^{-1}]$ on time step n has the form

$$\mathbf{S}^n = \begin{bmatrix} \text{coef}_{11_n} & \text{coef}_{12_n} \\ \text{coef}_{12_n} & \text{coef}_{22_n} \end{bmatrix}.$$

Next compulsory line is `{`, last line of the block is `}`. Each line between these two lines describes the diffusion–dispersion tensor for one element (different from the default one). The format is the following:

```
fid eid coef_11_1 coef_22_1 coef_12_1 ... coef_11_N coef_22_N coef_12_N
```

Description:

`fid` (int) Id number of the fracture.

`eid` (int) Id number of the element.

`coef_jk_n` (double) Diffusion–dispersion tensor on n -th time interval.

Key: Diffusion_dispersion_coefficients

Next compulsory line has the following format (no time-dependency):

```
default coef_1 coef_2 coef_3
```

Allowed values for `coef_j`: Nonnegative double values.

Description: The diffusion–dispersion tensor \mathbf{S} has the form

$$\begin{aligned} \mathbf{S}_{ii} &= \alpha_T |\mathbf{w}| + (\alpha_L - \alpha_T) \frac{\mathbf{w}_i^2}{|\mathbf{w}|} + \sigma \quad i = 1, 2, \\ \mathbf{S}_{ij} = \mathbf{S}_{ji} &= (\alpha_L - \alpha_T) \frac{\mathbf{w}_i \mathbf{w}_j}{|\mathbf{w}|} \quad i, j = 1, 2, i \neq j, \end{aligned}$$

where α_L is the longitudinal dispersivity `coef_1` [L], α_T is the transverse dispersivity `coef_2` [L], σ is the molecular diffusion coefficient `coef_3` [L²T⁻¹], and \mathbf{w}_i are the components of the velocity vector \mathbf{w} and $|\mathbf{w}|$ is its length. Next compulsory line is {, last line of the block is }. Each line between these two lines describes the diffusion–dispersion coefficients for one element (different from the default ones). The format is the following:

```
fid eid coef_1 coef_2 coef_3
```

Description:

`fid` (int) Id number of the fracture.

`eid` (int) Id number of the element.

`coef_j` (double) Diffusion–dispersion coefficients.

Key: `Velocity_field`

Next compulsory line has the following format:

```
default coef_11_1 coef_12_1 coef_13_1 coef_21_1 coef_22_1 coef_23_1 ...
        coef_11_N coef_12_N coef_13_N coef_21_N coef_22_N coef_23_N
```

Allowed values for `coef_jk_n`: Double values.

Description: The velocity field \mathbf{w}^n [LT⁻¹] on time step n has the form

$$\mathbf{w}^n = \begin{pmatrix} \text{coef}_{11} + \text{coef}_{12}x + \text{coef}_{13}y \\ \text{coef}_{21} + \text{coef}_{22}x + \text{coef}_{23}y \end{pmatrix}.$$

Next compulsory line is {, last line of the block is }. Each line between these two lines describes the velocity field for one element (different from the default one). The format is the following:

```
fid eid coef_11_1 coef_12_1 coef_13_1 coef_21_1 coef_22_1 coef_23_1 ...
        coef_11_N coef_12_N coef_13_N coef_21_N coef_22_N coef_23_N
```

Description:

`fid` (int) Id number of the fracture.

`eid` (int) Id number of the element.

`coef_jk_n` (double) Velocity field coefficients on n -th time interval.

Finally, the two following blocks have again the same structure as the sources:

Key: `Bulk_density`

Next compulsory line has the following format (no time-dependency):

```
default bulk_dens
```

Allowed values for `bulk_dens`: Nonnegative double values.

Description: Bulk density [ML⁻²].

Key: `Water_content`

Next compulsory line has the following format:

```
default wtr_cnt_0 wtr_cnt_1 ... wtr_cnt_N
        wtr_cnt_im,0 wtr_cnt_im,1 ... wtr_cnt_im,N
```

Allowed values for `wtr_cnt_n`, `wtr_cnt_im,n`: Nonnegative double values.

Description: Water content, immobile water content (dimensionless [-]). Note that the initial water content `wtr_cnt_0` has to be given as well when `Evolving_coef = Yes`. Coefficients `wtr_cnt_im,n` are only compulsory when `Immobile = Yes`.

4.6 Initial conditions file format

The initial conditions file has the following format:

First line: Compulsory heading [FRACTURE_TRANSPORT_INITIAL_CONDITIONS]

Follow the initial conditions for the different schemes; FTRANS always only searches for the appropriate format according to the numerical method.

The schemes FV/MFE/CMFE. Compulsory first line `FV/MFE/CMFE{`, last line `}FV/MFE/CMFE`. Each line between these two lines describes the initial concentration for one element. The format is the following:

```
fid eid conc_init conc_init_im
```

Description:

`fid` (int) Id number of the fracture.

`eid` (int) Id number of the element.

`conc_init` (double) Initial concentration on the given element.

`conc_init_im` (double) Immobile initial concentration (only read when `Immobile = Yes`).

The schemes FV-NCFE. Compulsory first line `FV-NCFE{`, last line `}FV-NCFE`. Each line between these two lines describes the initial concentration for one side. The format is the following:

```
fid eid sid conc_init conc_init_im
```

Description:

`fid` (int) Id number of the fracture.

`eid` (int) Id number of the element.

`sid` (int) Id number of the side of the element (0, 1, 2).

`conc_init` (double) Initial concentration on the given side.

`conc_init_im` (double) Immobile initial concentration (only read when `Immobile = Yes`).

Since in this way, initial concentrations on one side may be described more than once for a side shared by two or more elements, it is necessary that in this case all the given values coincide. For implicit time discretizations (`Time_discr_par = 0`) and when `Type_of_linearization = 0`, initial conditions for non-Dirichlet sides are sufficient. Otherwise initial conditions on Dirichlet sides are required as well.

The scheme FV-FE. Compulsory first line `FV-FE{`, last line `}FV-FE`. Each line between these two lines describes the initial concentration for one node. The format is the following:

```
fid nid conc_init conc_init_im
```

Description:

`fid` (int) Id number of the fracture.

`nid` (int) Id number of the node.

`conc_init` (double) Initial concentration at the given node.

`conc_init_im` (double) Immobile initial concentration (only read when `Immobile = Yes`).

Since in this way, initial concentrations on one node may be described more than once for a node

shared by two or more fractures, it is necessary that in this case all the given values coincide. For implicit time discretizations (`Time_discr_par = 0`), initial conditions for non-Dirichlet nodes are sufficient. Otherwise initial conditions on Dirichlet nodes are required as well.

4.7 Boundary conditions file format

The boundary conditions file has the following format:

First line: Compulsory heading [`FRACTURE_TRANSPORT_BOUNDARY_CONDITIONS`]

Follow the boundary conditions for the different schemes; FTRANS always only searches for the appropriate format according to the numerical method. For elliptic problems or in the case where `Evolving_BC = No`, only the boundary conditions on the first time level (`BC_1`) are considered.

The schemes FV/MFE/CMFE/FV-NCFE. Compulsory first line `FV/MFE/CMFE/FV-NCFE{`, last line `}FV/MFE/CMFE/FV-NCFE`. Each line between these two lines describes the boundary condition for one side. The format is the following:

```
type fid eid sid BC_1 ... BC_N
```

Description:

`type` (int) Type of the boundary condition: 1 Dirichlet, 2 Neumann, 3 Robin (Newton).

`fid` (int) Id number of the fracture.

`eid` (int) Id number of the element.

`sid` (int) Id number of the side of the element (0, 1, 2).

`BC_n` (double) Boundary condition on n -th time level.

The scheme FV-FE. Compulsory first line `FV-FE{`, last line `}FV-FE`. Each line between these two lines describes the boundary conditions for one node. The format is the following:

```
type fid nid BC_1 ... BC_N
```

Description:

`type` (int) Type of the boundary condition: 1 Dirichlet, 2 Neumann, 3 Robin (Newton).

`fid` (int) Id number of the fracture.

`nid` (int) Id number of the node.

`BC_n` (double) Boundary condition on n -th time interval.

Since in this way, boundary concentrations on one node may be described more than once for a node shared by two or more fractures, it is necessary that in this case all the given values coincide.

4.8 Output files format

Finally, the output files have the following format:

4.8.1 Output file with results

First line: Heading [`FRACTURE_TRANSPORT_RESULTS`]

Second line: The brief description of the given model given in `Description`.

Elliptic problems. Each line describes the results in one element and has the following form:

```
fid eid conc_e conc_s_1 conc_s_2 conc_s_3 conc_v_1 conc_v_2 conc_v_3
```

Description:

fid (int) Id number of the fracture.
eid (int) Id number of the element.
conc_e (double) Concentration in the given element (the result of the FV/MFE/CMFE schemes).
conc_s_i (double) Concentrations at the given element's sides (the result of FV–NCFE scheme).
conc_v_i (double) Concentrations at the given element's nodes (the result of FV–FE scheme).

Parabolic problems. Blocks of the form of elliptic problems, preceded by a number of the time level.

4.8.2 Output file with a posteriori error estimates

First line: Heading [FRACTURE_TRANSPORT_A_POSTERIORI_ERROR_ESTIMATES]

Second line: The brief description of the given model given in *Description*.

Each line gives the a posteriori error estimate for the MFE/CMFE schemes (cf. [20]) for one element and has the following form:

```
fid eid conc_e pol_1 ... pol_6 err_est norm est_res est_NC est_upw
      pol_int_1 ... pol_int_6
```

Description:

fid (int) Id number of the fracture.

eid (int) Id number of the element.

conc_e (double) Concentration in the given element.

pol_i (double) Coefficients for the second-order postprocessed polynomial \tilde{p}_h .

err_est (double) Error estimate in the given element.

norm (double) Estimate on the norm of \tilde{p}_h in the given element.

est_res (double) Residual estimator in the given element.

est_NC (double) Nonconformity estimator in the given element.

est_upw (double) Upwinding estimator in the given element.

pol_i (double) Coefficients for the second-order polynomial—modified Oswald interpolate of \tilde{p}_h .

4.9 Various remarks

We give in this section several important remarks on using of FTRANS.

4.9.1 Time discretization

The preferred choice for the time discretization scheme is the fully implicit (backward Euler) one (`Time_discr_par` = 0). The stability is then guaranteed in the sense of the discrete maximum principle under some conditions on the mesh geometry and the diffusion–dispersion tensor for FV, FV–FE, and FV–NCFE schemes (see [8, Lemma 18.1], [9, Theorem 4.11], and [10, Theorem 4.5], respectively). If these conditions are not verified, the oscillations shall still be very limited, which holds true also for the MFE/CMFE scheme. Approaching `Time_discr_par` = 0.5, precision may be gained, but the stability may suffer. Anytime `Time_discr_par` is different from 1, linear system(s) (on each time level) have to be solved. This suggests the use of the explicit time discretization, where however very strong stability conditions hold: when no diffusion is present, the orders of the time and space steps have to be equivalent; when there is some diffusion, the time step has to be of the order of the square of the space step, which is usually too restrictive.

4.9.2 Direct solvers

Today's direct solvers are very efficient. For linear problems with up to roughly 20 000 unknowns, in MATLAB (`Solver = 0`), we suggest the use of the combination of column minimum degree ordering and LU factorization (see the description of the backslash `\` operator in MATLAB help), which is in FTRANS specified by setting `Solver_method = 0` and `Preconditioning = 1`, see Section 4.2.4.

4.9.3 Inexact Newton method

For nonlinear problems, we suggest the use of the inexact Newton method, see e.g. [14]. It is a connection of the classical Newton method for the solution of a nonlinear matrix problem with an iterative method for the solution of resulting linear matrix problems. One limits here the maximum number of iterations of the iterative solver `Solver_max_iter` while not limiting the maximal number of iterations of the Newton linearization `Max_iter_lin`. Having an approximate solution on a k -th step of the Newton method, several iterations of the iterative method started from this solution do not give an exact result of the $(k + 1)$ -th step of the Newton method, but are usually sufficient to move the approximate solution towards the exact solution of the $(k + 1)$ -th step of the Newton method. Then assembling the Newton matrix of this approximate solution is usually sufficient to move the approximation towards the exact solution of the original nonlinear matrix problem. Bi-CGStab and GMRES iterative methods (`Solver_method = 2` and `Solver_method = 4`) work well in this respect.

4.9.4 LU factorization only at the first step

In the preconditioning technique, it is sufficient that the obtained L and U matrices represent only approximate LU decomposition of the matrix at hand. What is important is finally the condition number of the preconditioned matrix. Hence it may not be necessary to decompose the matrix at hand always if we already have "some" decomposition. So that in particular for a nonlinear problem, we may only make the LU factorization on the first linearization step and then keep the obtained L and U factors also for subsequent Newton linearization steps. This may lead to important savings in CPU time. For a linear parabolic problem with constant coefficients and time step, the system matrix in fact even does not change at all. Hence in this case, we may make a very precise incomplete LU decomposition at the first time step and then the preconditioned iterative method on each time step will converge in only one iteration. Alternatively, full LU decomposition may be made and then only triangular systems are to be solved by substitution on each time level.

4.9.5 Particular cases for the different schemes

We list here for the sake of completeness the particular cases, exceptions, and (temporary) limitations for the different schemes of FTRANS.

The finite volume scheme

Mesh We recall that all the triangular elements of the mesh for the finite volume scheme have to have all angles smaller than 90 degrees.

Diffusion–dispersion tensor We recall that only scalar diffusion–dispersion tensors, $\mathbf{S} = sId$, are allowed in the finite volume scheme.

The mixed/condensed mixed finite element schemes

Time discretization We recall that for the moment, only fully implicit time discretizations (`Time_discr_par = 0`) are possible for MFE/CMFE.

Relation MFE–CMFE We recall that the MFE and CMFE schemes are completely equivalent. The only difference is that in the MFE scheme, the matrix problems contain both scalar and flux unknowns, whereas in the CMFE scheme, the flux unknowns are locally eliminated and the final matrix problems only contain the scalar unknowns, thus using the CMFE scheme is likely to considerably reduce the CPU time. For the details, see [18, 19].

Type of algebraic condensation for CMFE We recall that the only type of algebraic condensation enabling the discretization of the full problem (2.2a)–(2.2e) is `Mixed_alg_cond = 4`. This is also the optimized variant, so that we strongly recommend its use; note however that this variant needs MATLAB as a computational engine and that the file `matr_A_inv_J.m` has to be in some directory searched by MATLAB.

Feasibility of CMFE The feasibility of the algebraic condensation is in [18, 19] only proven under some hypotheses on the mesh \mathcal{T}_h and the diffusion–dispersion tensor \mathbf{S} . This hypothesis is not too restrictive and allows for fairly general meshes and diffusion–dispersion tensors. However, an example showing a particular case where the choice `Mixed_alg_cond = 4` does not work is given in these references. In order to check whether the algebraic condensation works properly for the problem at hand, one should make a test calculation with `Mixed_alg_cond_cnd_nb = Yes`. As soon as the condition numbers of the local condensation matrices are well bounded, there is no risk at using the algebraic condensation. Shall the local condensation matrices be singular or close to, FTRANS will automatically stop the calculation. To conclude, for all test calculations made with FTRANS, we did not observe any difficulty.

CMFE on fracture networks We recall that CMFE for the moment only works for planar domains and not for fracture networks.

The combined finite volume–finite element scheme

Imposing boundary conditions In the combined finite volume–finite element scheme, the unknowns are located at the vertices. Imposing the Dirichlet/Neumann/Newton boundary conditions for this scheme amounts to specifying the mean of the functions u_D , u_N , u_R , respectively, over the set of all line segments given by one-half of the exterior edges $\subset \Gamma_D$, $\subset \Gamma_N$, or $\subset \Gamma_R$, respectively, that share the given vertex. In contrast to the other schemes, the boundaries Γ_D , Γ_N , and Γ_R are supposed to conform with such sets and not with the exterior triangle edges.

The combined finite volume–nonconforming finite element scheme

Degenerate parabolic problems We recall that as soon as $\beta' = \infty$ at some point (usually 0) and as soon as the solution may take this value (i.e. usually as soon as there may be some part of the domain with zero concentration), the problem at hand is degenerate parabolic. This is in particular the case of the Freundlich adsorption isotherm (`ads_fn = 1`) when `ads_par_2` is less than one. In this case, one has to use the option `Type_of_linearization = 1`. This option is for the moment only at the disposal for the FV–NCFE scheme.

4.9.6 Recommendations on the choice between the different schemes

A lot of factors may be comprised in a comparison of different schemes; precision, stability in the convection-dominated case, satisfaction of the discrete maximum principle, behaviour for strongly inhomogeneous and anisotropic diffusion–dispersion tensors, and memory and CPU time requirements are just some of them. Moreover, it is impossible to generally say that one scheme is better than the other for all problems. We however try to outline here the strengths of the individual schemes of FTRANS.

The numerical experiments for the contaminant transport problem (2.2a)–(2.2e) carried out in [18, Section 3.5] indicate that for problems with discontinuous coefficients and inhomogeneous and anisotropic diffusion–dispersion tensors (which is the case of contaminant transport one), mixed finite element schemes clearly give the best results. Using their condensed form, one can reduce the necessary CPU times by a factor comprised between 2 and 4, so that this variant is both the most precise and efficient (meaning the dependence of precision on the CPU time). This has been shown for planar domains in [18, Section 3.5] as well. There is no need to construct any dual mesh but the overall complexity of implementation however stays higher than that of the combined finite volume–finite element or the finite volume schemes. Recall that in the condensed form, the number of unknowns is equal to the number of triangles. The two combined schemes are based on the same ideas. In the combined finite volume–nonconforming finite element scheme, the construction of the dual volumes is straightforward and the overall structure of the scheme is much simpler. In particular, it has a more compact stencil (in planar regions, there are always at most five nonzero entries on each matrix row), a feature appealing from the parallelization viewpoint. Generalization to the three-dimensional case is equally straightforward. These advantages are however compensated by the fact that the number of unknowns is increased, equaling to the number of sides. Whereas the number of unknowns of the combined finite volume–finite element scheme is given by the number of nodes, the dual volumes are less easy to construct (especially for the potential extensions to three space dimensions) and the overall scheme structure is a bit more complicated. The two schemes give reasonable results for the contaminant transport problem in its complete form and are fully applicable without any restrictions on the triangular mesh. The use of the local Péclet upstream weighting in particular permits to reduce the excessive numerical diffusion necessary to stabilize the schemes in convection-dominated cases. Finally, when the diffusion–dispersion tensor reduces to a scalar function and when all the angles of the mesh are smaller or equal to 90 degrees (which are however quite restrictive assumptions), the finite volume method represents a vital alternative, easy to implement, not requiring the construction of any dual volumes and being well efficient.

5 FTRANS organization

We give here a short note on the organization of FTRANS so as to enable a rapid orientation for those aiming at further developments of the basic code. Since FTRANS is based on FFLOW by Otto Severýn from the Department of Process Modeling in TU Liberec, its basic organization is inherited from this program.

5.1 Structures

The basic structure of TRANS is `Problem`, containing in particular all the parameters from the control file, see Section 4.2. Structure `Mesh` then contains all the parts of the mesh, given by structures `Fracture`, `Neighbour`, `Boundary` and `Element` (for triangles), `Node`, and `Side`. These

structures are common for all the schemes of FTRANS and sufficient for the FV, MFE, and CMFE schemes. The dual volumes around nodes for the FV–FE scheme are given by `D_volume`, whereas the dual nodes by `D_node` and the dual sides by `D_side`. Similarly for the dual mesh of the FV–NCFE scheme, the structure `Covolume` is used. Error messages in FTRANS has a separate (and last structure) `Error`.

5.2 Basic code organization

The code is organized around the `FTRANS.bpr` project file. All the structures are declared in the `structs.h` file, the basic function headers are in the `header.h` file, and error messages in the `messages.h` file. Function `main` is in the file `main.c` and ensures the actual execution of the program, where `calculation.c` is responsible for the preparation of the schemes and `parabolic_cycle.c` controls the advancement of FTRANS and in particular calls `matrix_rhs.c` passing the matrix and right-hand side assembly tasks to the individual files for the different FTRANS schemes. All the communication with MATLAB is coded in the `solve_matlab.c` file; `solve_gm6.c` is responsible for calling the gm6 solver.

5.3 Treatment of the data for parabolic problems

There may be a significant amount of time-dependent data for real problems. FTRANS treats them in the following way: all the time-dependent entries are first read, controlled, and associated to elements, sides, or nodes. Then three files `__BC_time.bin`, `__coef_time.bin`, and `__sour_time.bin` are created, where all the time-dependent data are stored in the binary format, being previously sorted by time levels. In FTRANS subsequent execution, all the boundary conditions, coefficients, and sources necessary for the given time step are read from this binary file.

References

- [1] ADLER, P., AND THOVERT, J.-F. *Fractures and Fracture Networks*. Kluwer Academic Publisher, Dordrecht, Holland, 1999.
- [2] BEAR, J., AND VERRUIJT, A. *Modelling Groundwater Flow and Pollution*, vol. 2 of *Theory and Applications of Transport in Porous Media*. Kluwer Academic Publisher, Dordrecht, Holland, 1987.
- [3] BREZZI, F., AND FORTIN, M. *Mixed and hybrid finite element methods*, vol. 15 of *Springer Series in Computational Mathematics*. Springer-Verlag, New York, 1991.
- [4] DAWSON, C. Analysis of an upwind-mixed finite element method for nonlinear contaminant transport equations. *SIAM J. Numer. Anal.* 35, 5 (1998), 1709–1724.
- [5] DAWSON, C., AND AIZINGER, V. Upwind-mixed methods for transport equations. *Comput. Geosci.* 3, 2 (1999), 93–110.
- [6] DOUGLAS, JR., J., AND ROBERTS, J. E. Mixed finite element methods for second order elliptic problems. *Mat. Apl. Comput.* 1, 1 (1982), 91–103.
- [7] DOUGLAS, JR., J., AND ROBERTS, J. E. Global estimates for mixed methods for second order elliptic equations. *Math. Comp.* 44, 169 (1985), 39–52.

- [8] EYMARD, R., GALLOUËT, T., AND HERBIN, R. Finite volume methods. In *Handbook of Numerical Analysis, Vol. VII*. North-Holland, Amsterdam, 2000, pp. 713–1020.
- [9] EYMARD, R., HILHORST, D., AND VOHRALÍK, M. A combined finite volume–finite element scheme for the discretization of strongly nonlinear convection–diffusion–reaction problems on nonmatching grids. *Submitted* (2006).
- [10] EYMARD, R., HILHORST, D., AND VOHRALÍK, M. A combined finite volume–nonconforming/mixed-hybrid finite element scheme for degenerate parabolic problems. *To appear in Numer. Math.* (2006).
- [11] JAFFRÉ, J. Décentrage et éléments finis mixtes pour les équations de diffusion-convection. *Calcolo* 21, 2 (1984), 171–197.
- [12] JAVANDEL, I., DOUGHTY, C., AND TSANG, C. *Groundwater Transport: Handbook of Mathematical Models*, vol. 10. American Geophysical Union Water Resources Monograph Series, New York, 1984.
- [13] MARYŠKA, J., SEVERÝN, O., AND VOHRALÍK, M. Numerical simulation of fracture flow with a mixed-hybrid FEM stochastic discrete fracture network model. *Comput. Geosci.* 8, 3 (2004), 217–234.
- [14] QUARTERONI, A., SACCO, R., AND SALERI, F. *Numerical mathematics*, vol. 37 of *Texts in Applied Mathematics*. Springer-Verlag, New York, 2000.
- [15] RAVIART, P.-A., AND THOMAS, J.-M. A mixed finite element method for 2nd order elliptic problems. In *Mathematical aspects of finite element methods (Proc. Conf., Consiglio Naz. delle Ricerche (C.N.R.), Rome, 1975)*. Springer, Berlin, 1977, pp. 292–315. Lecture Notes in Math., Vol. 606.
- [16] ROBERTS, J. E., AND THOMAS, J.-M. Mixed and hybrid methods. In *Handbook of Numerical Analysis, Vol. II*. North-Holland, Amsterdam, 1991, pp. 523–639.
- [17] VOHRALÍK, M. Equivalence between mixed finite element and multi-point finite volume methods. *C. R. Math. Acad. Sci. Paris* 339, 7 (2004), 525–528.
- [18] VOHRALÍK, M. *Numerical Methods for Nonlinear Elliptic and Parabolic Equations. Application to Flow Problems in Porous and Fractured media*. Ph.D. dissertation, Université de Paris-Sud & Czech Technical University in Prague, December 2004.
- [19] VOHRALÍK, M. Equivalence between lowest-order mixed finite element and multi-point finite volume methods on simplicial meshes. *To appear in M2AN Math. Model. Numer. Anal.* (2006).
- [20] VOHRALÍK, M. A posteriori error estimates for lowest-order mixed finite element discretizations of convection–diffusion–reaction equations. *Submitted* (2006).
- [21] VOHRALÍK, M. Residual flux-based a posteriori error estimates for finite volume discretizations of inhomogeneous, anisotropic, and convection-dominated problems. *Submitted* (2006).
- [22] VOHRALÍK, M., MARYŠKA, J., AND SEVERÝN, O. Mixed and nonconforming finite element methods on a system of polygons. *To appear in Appl. Numer. Math.* (2006).
- [23] ZHENG, C., AND BENNETT, G. *Applied Contaminant Transport Modeling*. Van Nostrand Reinhold, New York, 1995.

A Example of a control file

```
[Global]
Solution_method      = 5
Description          = Experimental, non-real problem
```

```
[Input]
Space_mesh          = _space_mesh.txt
Time_mesh           = _time_mesh.txt
Coefficients        = _coef.txt
Initial_conditions  = _init_cond.txt
Boundary_conditions = _bound_cond.txt
```

```
[Run]
Log_file            = debug.log
Screen_verbosity    = 7
Log_verbosity       = 7
```

```
[Solver]
Solver              = 0
Solver_method       = 2
Solver_accuracy     = 1e-8
Solver_max_iter     = 1000
Preconditioning     = 2
Precond_accur       = 2e-4
Save_matrix         = No
```

```
[Output]
Running_results     = Yes
Final_results       = Yes
Output_digits       = 16
Output_file         = FTRANS_out.txt
Output_file_immob   = FTRANS_out_im.txt
POS_file            = No
```

```
[Parameters]
Parabolic           = Yes
Evolving_BC         = Yes
Evolving_sour       = Yes
Evolving_coef       = No
Time_discr_par      = 0
Nonlinear           = Yes
Prec_lin            = 1e-8
Max_iter_lin        = 50
Lin_error_norm      = 1
Dif_disp_by_cfs     = No
Sour_inj_pump_div   = No
Immobile            = No
```


MIM_coupled = No

[Particular]

Type_of_linearization = 0
LU_fact_only_first = Yes
GMRES_restart_iter = 5
Mixed_conv_type = 3
Mixed_upstr_weigh_par = -1
Mixed_column_ordering = No
Mixed_alg_cond = 4
Mixed_alg_cond_cnd_nb = No
Postprocessing = 1
A_posteriori_err_est = 0

B Examples of input files

B.1 Example of a mesh file

[FRACTURE_FLOW_MESH]

1

```
1 1 -1
Fracture{
  0 1 2
  0 0 0
  3
  0 0 1
  1
  0 0 0 0 0
  3 0 0 3 0
  3 0 0
  0 3 0
  Nodes{
    0 0 0
    1 1 0
    2 0 1
    3 0.3 0.3
    4 0.7 0.7
    5 1 1
  }Nodes
  Elements{
    0 0 1 3 1 1 0
    1 0 3 2 1 1 0
    2 3 1 2 1 1 0
    3 1 4 2 1 1 0
    4 1 5 4 1 1 0
    5 2 4 5 1 1 0
  }Elements
  Neighbours{
```

```

        0  1  2  1  1  2
        0  1  2  2  1  2
    }Neighbours
}Fracture
Fracture{
    1  1  2
    0  0  1
    3
    0  0  1
    1
    0  0  1  0  0
    3  0  1  3  0
    3  0  0
    0  3  0
    Nodes{
        0 0 0
        1 1 0
        2 0 1
        3 0.3 0.3
        4 0.7 0.7
        5 1 1
    }Nodes
    Elements{
        0  0  1  3  1  1  0
        1  0  3  2  1  1  0
        2  3  1  2  1  1  0
        3  1  4  2  1  1  0
        4  1  5  4  1  1  0
        5  2  4  5  1  1  0
    }Elements
    Neighbours{
        1  1  2  2  1  2
        1  1  2  0  1  2
    }Neighbours
}Fracture
Fracture{
    2  1  2
    0  0  2
    3
    0  0  1
    1
    0  0  2  0  0
    3  0  2  3  0
    3  0  0
    0  3  0
    Nodes{
        0 0 0
        1 1 0

```

```

    2 0 1
    3 0.3 0.3
    4 0.7 0.7
    5 1 1
  }Nodes
  Elements{
    0 0 1 3 1 1 0
    1 0 3 2 1 1 0
    2 3 1 2 1 1 0
    3 1 4 2 1 1 0
    4 1 5 4 1 1 0
    5 2 4 5 1 1 0
  }Elements
  Neighbours{
    2 1 2 0 1 2
    2 1 2 1 1 2
  }Neighbours
}Fracture

```

B.2 Example of a time file

```

[FRACTURE_TRANSPORT_TIME_MESH]
Time_interval
1
Number_time_steps
2
Equidistant
0
Time_mesh
0.4 0.6

```

B.3 Example of a coefficients file

```

[FRACTURE_TRANSPORT_COEFFICIENTS]
Adsorption_function
default 11
FV/MFE/CMFE{
}FV/MFE/CMFE
FV-NCFE{
}FV-NCFE
FV-FE{
}FV-FE

Adsorption_parameters
default 0.5 1 1 1
FV/MFE/CMFE{
}FV/MFE/CMFE
FV-NCFE{
}FV-NCFE

```

```
FV-FE{
}FV-FE
```

```
Reaction_function
```

```
default 12
```

```
FV/MFE/CMFE{
```

```
}FV/MFE/CMFE
```

```
FV-NCFE{
```

```
}FV-NCFE
```

```
FV-FE{
```

```
}FV-FE
```

```
Reaction_parameters
```

```
default 0.5 1
```

```
FV/MFE/CMFE{
```

```
}FV/MFE/CMFE
```

```
FV-NCFE{
```

```
}FV-NCFE
```

```
FV-FE{
```

```
}FV-FE
```

```
Diffusion_dispersion_tensor
```

```
default 1 1 0
```

```
{
```

```
2 0 8 20 -10
```

```
2 1 8 20 -10
```

```
2 2 8 20 -10
```

```
2 3 8 20 -10
```

```
2 4 8 20 -10
```

```
2 5 8 20 -10
```

```
}
```

```
Diffusion_dispersion_coefficients
```

```
default 1 2 3
```

```
{
```

```
}
```

```
Velocity_field
```

```
default -1 1 0 0 0 0
```

```
{
```

```
1 0 0.5 1 0 0 0 0
```

```
1 1 0.5 1 0 0 0 0
```

```
1 2 0.5 1 0 0 0 0
```

```
1 3 0.5 1 0 0 0 0
```

```
1 4 0.5 1 0 0 0 0
```

```
1 5 0.5 1 0 0 0 0
```

```
2 0 0.5 1 0 0 0 0
```

```
2 1 0.5 1 0 0 0 0
```

```

2 2 0.5 1 0 0 0 0
2 3 0.5 1 0 0 0 0
2 4 0.5 1 0 0 0 0
2 5 0.5 1 0 0 0 0
}

```

```

Bulk_density
default 2.5
FV/MFE/CMFE{
}FV/MFE/CMFE
FV-NCFE{
}FV-NCFE
FV-FE{
}FV-FE

```

Sources

```
default 0 0
```

```

FV/MFE/CMFE{
0 0 0.9845701523385473 0.8339430542264101
0 1 0.4894619093631114 0.1388254781939984
0 2 0.9845701523385473 0.8339430542264101
0 3 1.295159780384606 1.23343476110585
0 4 2.922014831292236 3.558853454266147
0 5 1.157184814468891 0.6491406663594344
1 0 4.107583778456231 5.982922048292974
1 1 2.727198955825017 3.828230144929424
1 2 4.107583778456231 5.982922048292974
1 3 5.372582523073173 7.955968366612946
1 4 10.86374990699728 16.65256099984635
1 5 9.098919890173935 13.74284821193963
2 0 -8.384470726014506 -14.61299392797328
2 1 -6.223749230022607 -10.92938852201228
2 2 -8.384470726014506 -14.61299392797328
2 3 -10.93710844768109 -18.93416605541544
2 4 -20.9031903958229 -35.72226918247446
2 5 -22.66802041264624 -38.63198197038118
}FV/MFE/CMFE

```

```
}FV/MFE/CMFE
```

```
FV-NCFE{
```

```

0 0 0 1.121724969154648 1.07311973921749
0 0 1 1.521683266089526 1.606908850300689
0 0 2 0.5246250449365031 0.162542351860004
0 1 1 0.1625423518600035 -0.6339356848683435
0 2 1 1.07311973921749 0.7724965820253864
0 3 0 2.412349066179013 2.875680322631878
0 3 1 0.3847490827566764 -0.4672668985077557
0 4 0 3.509470277888987 4.50624903591965
0 4 1 2.875680322631878 3.326698927008579
0 5 2 0.7724965820253864 -0.006257713797412023

```

1	0	0	4.142354030360363	6.053295123322311
1	0	1	5.599106008778094	8.329442455807786
1	0	2	2.997706950986695	4.239965094548571
1	1	1	4.239965094548571	6.088597920638755
1	2	1	6.053295123322313	8.983417669616188
1	3	0	8.495149016446025	12.90452198605078
1	3	1	6.467549033023687	9.561574764911146
1	4	0	11.72039136547979	18.04376928507083
1	4	1	12.90452198605078	19.86146349797098
1	5	2	8.983417669616188	13.53126253535377
2	0	0	-7.940162214462496	-13.86740641309698
2	0	1	-10.71058496197618	-18.56069196622061
2	0	2	-6.894620673214073	-12.0697258762057
2	1	1	-12.0697258762057	-20.80153650138964
2	2	1	-13.86740641309697	-23.86026668074702
2	3	0	-15.83605078462202	-27.21084466762483
2	3	1	-17.86365076804436	-30.55379188876446
2	4	0	-21.12329298488342	-36.10631171153389
2	4	1	-27.21084466762483	-46.27759478587863
2	5	2	-23.86026668074702	-40.61881846125095
0	1	2	-1.898904092051067	-3.907055644887332

}FV-NCFE

FV-FE{

0	1	2.733178200585764	3.509470277888987
0	3	0.6798583965171781	0.3048035761873731
0	4	1.852401665420167	1.836625703324449
0	5	4.50624903591965	5.786138296045463
1	1	7.713353584690585	11.72039136547979
1	3	4.01816978925588	5.808748577616241
1	4	9.28195030201284	14.08588057217593
1	5	18.04376928507084	28.10573588335472
2	1	-12.2073479517287	-21.12329298488342
2	3	-9.335075781698929	-16.20703142809923
2	4	-20.43624424435786	-34.91113890322998
2	5	-36.10631171153391	-61.1726544658823
0	0	-0.7850491392068685	-1.898904092051067
0	2	-3.907055644887331	-7.438424505565415

}FV-FE

Water_content

default 1

FV/MFE/CMFE{

}FV/MFE/CMFE

FV-NCFE{

}FV-NCFE

FV-FE{

}FV-FE

B.4 Example of initial conditions file

[FRACTURE_TRANSPORT_INITIAL_CONDITIONS]

FV/MFE/CMFE{

0	0	1.704604865322754
0	1	1.22140275816017
0	2	1.704604865322754
0	3	2.225540928492467
0	4	4.334761826185209
0	5	4.334761826185209
1	0	1.704604865322754
1	1	1.22140275816017
1	2	1.704604865322754
1	3	2.225540928492467
1	4	4.334761826185209
1	5	4.334761826185209
2	0	1.704604865322754
2	1	1.22140275816017
2	2	1.704604865322754
2	3	2.225540928492467
2	4	4.334761826185209
2	5	4.334761826185209

}FV/MFE/CMFE

FV-NCFE{

0	0	0	1.648721270700128
0	0	1	2.225540928492468
0	0	2	1.349858807576003
0	1	1	2.225540928492468
0	1	2	1.648721270700128
0	2	1	2.718281828459046
0	3	0	3.320116922736547
0	3	1	3.320116922736547
0	4	0	4.481689070338065
0	4	1	5.473947391727199
0	5	2	4.481689070338065
1	0	0	1.648721270700128
1	0	1	2.225540928492468
1	0	2	1.349858807576003
1	1	1	2.225540928492468
1	1	2	1.648721270700128
1	2	1	2.718281828459046
1	3	0	3.320116922736547
1	3	1	3.320116922736547
1	4	0	4.481689070338065
1	4	1	5.473947391727199
1	5	2	4.481689070338065
2	0	0	1.648721270700128
2	0	1	2.225540928492468

```

2 0 2 1.349858807576003
2 1 1 2.225540928492468
2 1 2 1.648721270700128
2 2 1 2.718281828459046
2 3 0 3.320116922736547
2 3 1 3.320116922736547
2 4 0 4.481689070338065
2 4 1 5.473947391727199
2 5 2 4.481689070338065

```

}FV-NCFE

FV-FE{

```

0 0 1
0 1 2.718281828459046
0 2 2.718281828459046
0 3 1.822118800390509
0 4 4.055199966844675
0 5 7.389056098930652
1 0 1
1 1 2.718281828459046
1 2 2.718281828459046
1 3 1.822118800390509
1 4 4.055199966844675
1 5 7.389056098930652
2 0 1
2 1 2.718281828459046
2 2 2.718281828459046
2 3 1.822118800390509
2 4 4.055199966844675
2 5 7.389056098930652

```

}FV-FE

B.5 Example of boundary conditions file

[FRACTURE_TRANSPORT_BOUNDARY_CONDITIONS]

FV/MFE/CMFE/FV-NCFE{

```

1 0 0 0 2.45960311115695 4.481689070338065
3 0 4 0 -6.68589444227927 -12.18249396070348
1 0 5 2 6.68589444227927 12.18249396070348
1 1 0 0 2.45960311115695 4.481689070338065
3 1 4 0 3.342947221139635 6.091246980351738
1 1 5 2 6.68589444227927 12.18249396070348
1 2 0 0 2.45960311115695 4.481689070338065
3 2 4 0 23.40063054797745 42.63872886246217
1 2 5 2 6.68589444227927 12.18249396070348

```

}FV/MFE/CMFE/FV-NCFE

FV-FE{

```

1 0 0 1.49182469764127 2.718281828459046
1 0 2 4.055199966844675 7.389056098930652

```



```

1 0 1 4.055199966844675 7.389056098930652
1 0 5 11.02317638064161 20.08553692318768
1 1 0 1.49182469764127 2.718281828459046
1 1 2 4.055199966844675 7.389056098930652
1 1 1 4.055199966844675 7.389056098930652
1 1 5 11.02317638064161 20.08553692318768
1 2 0 1.49182469764127 2.718281828459046
1 2 2 4.055199966844675 7.389056098930652
1 2 1 4.055199966844675 7.389056098930652
1 2 5 11.02317638064161 20.08553692318768

```

}FV-FE

C Example of an output file

[FRACTURE_TRANSPORT_RESULTS]

Experimental, non-real problem

1

```

0 0 0 2.45960311115695 3.446763136931259 1.64745796908299 0 0 0
0 1 0 1.64745796908299 3.292085509024633 2.227765940077287 0 0 0
0 2 0 3.446763136931259 4.184159977409498 3.292085509024633 0 0 0
0 3 0 5.05026714798563 5.12083659952068 4.184159977409498 0 0 0
0 4 0 6.603613597321377 7.179774257085176 5.05026714798563 0 0 0
0 5 0 5.12083659952068 7.179774257085176 6.68589444227927 0 0 0
1 0 0 2.45960311115695 3.341416487306315 1.7129871756627 0 0 0
1 1 0 1.7129871756627 3.118863603991642 2.227765940077287 0 0 0
1 2 0 3.341416487306315 3.991703642623759 3.118863603991642 0 0 0
1 3 0 4.646600899867071 5.12214024867777 3.991703642623759 0 0 0
1 4 0 6.229371424920504 7.15906679246168 4.646600899867071 0 0 0
1 5 0 5.12214024867777 7.15906679246168 6.68589444227927 0 0 0
2 0 0 2.45960311115695 3.62505187623178 1.564758936395174 0 0 0
2 1 0 1.564758936395174 3.269992989753237 2.227765940077287 0 0 0
2 2 0 3.62505187623178 4.379237881395313 3.269992989753237 0 0 0
2 3 0 4.601767834201824 5.483412271213194 4.379237881395313 0 0 0
2 4 0 5.953932895329543 7.059921442415086 4.601767834201824 0 0 0
2 5 0 5.483412271213194 7.059921442415086 6.68589444227927 0 0 0

```

2

```

0 0 0 4.481689070338065 6.343621860381246 2.920214566890867 0 0 0
0 1 0 2.920214566890867 6.040676050394491 4.04283907631766 0 0 0
0 2 0 6.343621860381246 7.692305284285964 6.040676050394491 0 0 0
0 3 0 9.268709598765152 9.400691600122482 7.692305284285964 0 0 0
0 4 0 12.04268076512371 12.94860744023239 9.268709598765152 0 0 0
0 5 0 9.400691600122482 12.94860744023239 12.18249396070348 0 0 0

```

1	0	0	4.481689070338065	6.057096000851955	3.086524209806753	0	0	0
1	1	0	3.086524209806753	5.619689778927603	4.04283907631766	0	0	0
1	2	0	6.057096000851955	7.195380185908584	5.619689778927603	0	0	0
1	3	0	8.150136170013463	9.420212765761335	7.195380185908584	0	0	0
1	4	0	10.92788504625798	12.68746591500146	8.150136170013463	0	0	0
1	5	0	9.420212765761335	12.68746591500146	12.18249396070348	0	0	0
2	0	0	4.481689070338065	6.649729808896692	2.811656138997174	0	0	0
2	1	0	2.811656138997174	5.996228254106752	4.04283907631766	0	0	0
2	2	0	6.649729808896692	8.030303521310053	5.996228254106752	0	0	0
2	3	0	8.435272097122487	10.03284281046946	8.030303521310053	0	0	0
2	4	0	10.85210455908557	12.8413350214933	8.435272097122487	0	0	0
2	5	0	10.03284281046946	12.8413350214933	12.18249396070348	0	0	0