Journal of Computational Physics **157,** 181–198 (2000)
doi:10.1006/jcph.1999.6370, available online at http://www.idealibrary.com on IDEAL®

A Time-Splitting Technique for the Advection-Dispersion Equation in Groundwater

Annamaria Mazzia, Luca Bergamaschi, and Mario Putti

Dipartimento di Metodi e Modelli Matematici per le Scienze Applicate, Università degli Studi di Padova, via Belzoni 7, 35131 Padova, Italy
E-mail: mazzia@dmsa.unipd.it, berga@dmsa.unipd.it, putti@dmsa.unipd.it

Received March 22, 1999; revised August 30, 1999

In this paper a time-splitting technique for the two-dimensional advection-dispersion equation is proposed. A high resolution in space Godunov method for advection is combined with the RT0 Mixed Finite Element for the discretization of the dispersion term. Numerical tests on an analytical one-dimensional example ascertain the convergence properties of the scheme. At different Peclet numbers, the choice of optimal time step size used for the two equations is discussed, showing that with accurate selection of the time step sizes, the overall CPU time required by the simulations can be drastically reduced. Results on a realistic test case of groundwater contaminant transport confirm that the proposed scheme does not suffer from Peclet limitations and always displays only small amounts of numerical diffusion across the entire range of Peclet numbers. © 2000 Academic Press

Key Words: time-splitting; transport equation; finite volume; mixed hybrid finite element; triangular grid.

1. INTRODUCTION

In this paper we concentrate on the advection-dispersion equation, which, for example, can be used to describe solute transport in two-dimensional porous media. This equation is difficult to approximate when advection dominates because sharp concentration fronts tend to develop and move without changing form. It is well known that standard finite difference and finite element methods may not work well for problems with sharp fronts, showing non-physical oscillations. To overcome these phenomena, numerical schemes try to combine numerical stability with minimal artificial diffusion. Two approaches are generally used in these situations. One is based on the definition of a proper control volume where upwind techniques can be used for approximating the advective flux. In this case the stability of the scheme is obtained by adding an amount of numerical diffusion that is dependent on the approach used [15]. The other class of methods originates from the splitting of the dispersion and advection fluxes into two separate partial differential equations (PDEs) containing one the dispersive and the other the advective term, respectively. These two equations are then discretized, each with the technique deemed most appropriate. Splitting allows the combination of explicit time-stepping for advective fluxes with implicit time-stepping for dispersive fluxes. This approach lessens the stability constraint connected with explicit discretization of the dispersion term but maintains the possibility of using efficient explicit schemes for advection. Belonging to this class are the Eulerian–Lagrangian schemes [14, 3] or the fully Eulerian Godunov-mixed methods (GMM) [4–7]. In this latter approach, a time-explicit, spatially second-order accurate Godunov method is used to treat advection, and a time-implicit, spatially second order accurate mixed finite element method is used for modeling dispersion.

In this work we investigate the numerical behavior of a time-splitting technique in two spatial dimensions, similar in spirit to the GMM approach. Using Euler time-stepping, the advective term is discretized by a triangle-based, high resolution finite volume (FV) scheme [16, 9, 13], while the dispersive flux is discretized using a mixed hybrid finite element (MHFE) technique. The choice of these two schemes is dictated, on one hand, by their accuracy, robustness, and efficiency in handling nonuniform meshes and highly variable coefficients. On the other hand, both FV and MHFE are based on the weak formulation of the governing equation and use similar functional spaces for the approximation of the dependent variable, making them ideally suited for combination in a time-splitting approach. The main difference between the GMM approach and the proposed one lies in the unstructured character of the spatial discretizations of the latter. This implies, in contrast with the original GMM approach, the use of triangular meshes together with fully multidimensional slope limiters in the FV phase. More precisely, the time splitting scheme employs explicit and implicit Euler time-stepping for FV and MHFE, respectively, while piecewise constant basis functions are used by both techniques to approximate concentration. Second order accuracy in space is obtained by MHFE at special superconvergence points (the centroids of the triangles) [8]. The FV approach achieves spatial second order accuracy (away from sharp fronts) by employing linear reconstruction plus slope limiting, combined in such a way as to locally satisfy the maximum principle [13]. The resulting numerical scheme is first order accurate in time and second order accurate in space. Both MHFE and FV are locally (at the element level) conservative and monotone. The combination of the two methods in the time splitting approach should maintain these two properties as long as stability requirements are met, as confirmed also by numerical results. In principle there are no difficulties in extending our technique to three dimensions. This is done by merely employing three-dimensional versions of MHFE and FV. Proper implementation of FV requires the development of tetrahedra based slope limiters, a field that is still subject of active research.

Numerical tests on a one-dimensional sample problem (movement of a tracer in a semi-infinite column) are used to validate theoretical results for different Peclet numbers. A heuristic analysis on the relative role of the two discretization schemes in the convergence behavior of the proposed approach is aimed at determining the best time-stepping strategy for the explicit and the implicit schemes. At each implicit step, a number of explicit time steps can be performed, according to accuracy and stability requirements. In this way the proposed approach can be viewed also as a sub-stepping technique for the solution of the advective phase. In addition, we present a realistic test case—the Gureghian test—and compare the numerical results obtained by the proposed scheme with those available in the literature [10]. These test cases show that the proposed approach does not suffer from Peclet limitations and always displays small amounts of numerical diffusion, maintaining high order of accuracy across the entire spectrum of Peclet numbers and high computational efficiency.

2. THE NUMERICAL SCHEME

Subsurface contaminant transport is governed by an advection-diffusion equation of the form

$$\begin{aligned} \frac{\partial \phi c}{\partial t} + \vec{\nabla}\cdot(\vec{v}c - D\vec{\nabla}c) &= f && \text{on } \Omega\times(0,T], \\ c &= c^0 && \text{on } \Omega\times 0, \\ c &= b_D && \text{on } \Gamma_D\times(0,T], \\ -D\vec{\nabla}c\cdot\vec{n} &= b_N && \text{on } \Gamma_N\times(0,T] \\ (\vec{v}c - D\vec{\nabla}c)\cdot\vec{n} &= b_C && \text{on } \Gamma_C\times(0,T], \end{aligned} \tag{1}$$

where c is the concentration of the solute, $\phi(t)$ is the porosity of the medium, $\vec{v}(\vec{x},t)$ is Darcy's velocity, $D = D(\vec{v})$ is the tensor accounting for mechanical dispersion and molecular diffusion, and f is a source or sink term ($\Omega \in \mathbb{R}^2$ and $\Gamma = \partial\Omega$).

Denoting by $\vec{F}$ and $\vec{G}$ the advective and dispersive fluxes, respectively, Eq. (1) may be written as

$$\frac{\partial \phi c}{\partial t} + \vec{\nabla}\cdot(\vec{F}+\vec{G}) = f \qquad \text{on } \Omega\times(0,T] \tag{2}$$

$$\vec{F} = \vec{v}c \tag{3}$$

$$\vec{G} = -D\vec{\nabla}c. \tag{4}$$

As the geometry of the physical domain Ω is often complex when dealing with real world applications, we choose to work with unstructured meshes, and thus Ω is discretized into m triangles, T_l, $l = 1,\dots,m$. Concentration c can be approximated by

$$c \simeq \sum_{l=1}^{m} c_l \psi_l, \tag{5}$$

where ψ_l are $P_0(T_l)$ scalar basis functions, taking on the value one on triangle T_l and zero elsewhere. Multiplying Eq. (2) by ψ_l and integrating in space and time, with time-step Δt over the time interval $[t^k, t^{k+1}]$, the following semidiscrete equations are obtained,

$$\phi_l^{k+1}c_l^{k+1} = \phi_l^k c_l^k - \frac{\Delta t}{|T_l|}\int_{T_l}[\vec{\nabla}\cdot(\vec{F}(c^{k+1-\theta}) + \vec{G}(c^{k+\theta})) - f^{k+\theta}]\,\mathrm{d}\Delta, \qquad l = 1,\dots,m, \tag{6}$$

where c_l^k is the volume average over T_l defined by

$$c_l^k = \frac{\int_{T_l} c(\cdot,t^k)\,\mathrm{d}\Delta}{|T_l|}, \tag{7}$$

$|T_l|$ is the area of T_l, ϕ is considered constant within each triangle, and a weighted scheme is used for the time quadrature with weighting parameter $\theta \in [0.5, 1]$ and $c^{k+\theta} = \theta c(\cdot, t^{k+1}) + (1-\theta)c(\cdot, t^k) = \theta c^{k+1} + (1-\theta)c^k$.

Denoting by L_d the spatial discretization operator for dispersion, where we also include the source term f, and by L_a the spatial discretization operator for advection, the fully discretized equations become

$$\phi_l^{k+1}c_l^{k+1} = \phi_l^k c_l^k + \Delta t\left[L_d(c^{k+\theta}) + L_a(c^{k+1-\theta})\right], \qquad l = 1, \ldots, m, \tag{8}$$

where for $\theta = 1$ we have the implicit Euler scheme for L_d and explicit Euler scheme for L_a. For $\theta = 0.5$ the above equation reduces to the midpoint rule for both L_d and L_a. In the following we consider only Euler schemes, i.e., $\theta = 1$.

The numerical fluxes L_a and L_d are evaluated by means of the discretization methods that are deemed more appropriate to solve, respectively, the advection and diffusion equations. For advection, we consider a high resolution triangular finite volume (FV) discretization by following the scheme developed in [13]. It is a second-order total-variation-diminishing (TVD) type scheme satisfying the maximum principle. It is accurate in the presence of steep fronts and introduces minimal numerical diffusion without oscillations. This method requires explicit time-stepping and thus stability is guaranteed by a CFL restriction on Δt. Possible nonlinearities can also be resolved without iteration. This can be seen as a disadvantage of the time-splitting method with respect to fully implicit schemes, which are not impaired by stability constraints. However, in many problems of practical importance, CFL numbers less than unity are required to maintain accuracy. Thus stability constraints are automatically satisfied and do not pose limitations. The dispersive flux is discretized by an implicit MHFE method. This approach has been chosen because of its intrinsic compatibility with the FV method. Since it is implicit in time, there is no stability restriction on the time-step connected with MHFE.

In the following we describe first the splitting scheme and then the two space discretization methods.

2.1. *The Time-Splitting Technique*

The time-splitting technique can be viewed as a predictor-corrector approach and can be described by the following algorithm:

For each time step do:

- advection step: for each T_l solve n_a times with the explicit FV scheme, with Δt_a as the time step, determining the predictor concentration $\hat{c}_l^{k+1}$

    1. $\phi_l^{(0)} := \phi_l^k \; c_l^{(0)} := c_l^k$
    2. DO $i_a = 0, n_a - 1$

    $$\phi^{(i_a+1)}c_l^{(i_a+1)} = \phi^{(i_a)}c_l^{(i_a)} + \Delta t_a\left[L_a\left(c_l^{(i_a)}\right)\right] \tag{9}$$

    END DO

    3. $\hat{c}_l^{k+1} := c_l^{(n_a)}$

- dispersion step: for each T_l solve with implicit MHFE method using $\hat{c}_l^{k+1}$ as initial condition

$$\phi^{k+1} c_l^{k+1} = \phi^k \hat{c}_l^{k+1} + \Delta t_d \left[L_d \left(c_l^{k+1} \right) \right] \tag{10}$$

with $\Delta t_d = n_a \Delta t_a$, obtaining the final approximation c_l^{k+1}.

Because the stability of the advection step is determined by the CFL constraint, while the dispersive time step is not subject to stability restrictions, we use two different time steps for advection and diffusion, Δt_a and Δt_d, respectively. Therefore a finer advection time step together with a coarser diffusive time step can be employed. The convergence rate of the scheme is influenced by the convergence rates of the two spatial discretization methods employed. Namely, first order accuracy in time and second order accuracy in space is expected at the centroids of the triangles, as both FV and MHFE are spatially second order accurate. Accuracy is also influenced by the different time step sizes that can be used in the advection and dispersion discretizations. A heuristic analysis trying to find the optimal n_a in different situations will be reported in a later section.

2.1.1. Finite volume discretization. In the advection step, Eq. (9) can be explicitly written as

$$\phi_l^{k+1} c_l^{k+1} = \phi_l^k c_l^k - \frac{\Delta t}{|T_l|} \int_{T_l} \vec{\nabla} \cdot \left(\vec{F}\left(c_l^k \right) \mathrm{d}\Delta, \qquad l = 1, \ldots, m \tag{11}$$

and is solved using as initial condition the solution calculated at the end of the previous time step. The discretization of (11) is obtained by means of the finite volume scheme on the unstructured triangular grid, as developed by [9] and then modified by [13]. The technique can be described as follows.

Application of the divergence theorem to the right-hand side of (11) yields

$$\phi_l^{k+1} c_l^{k+1} = \phi_l^k c_l^k - \frac{\Delta t}{|T_l|} \sum_{j=1}^{3} \int_{e_{lj}} \vec{F}\left(c_l^k \right) \cdot \vec{n}_{lj} \, \mathrm{d}\Gamma, \tag{12}$$

where e_{lj}, $j = 1, 2, 3$, are the edges of triangle T_l and $\vec{n}_{lj}$ is the corresponding outward unit normal. The approximation of the three line integrals in the above equation is obtained by a two step procedure. The reconstruction step approximates the values of concentration c_l^k over each triangle. Second order accurate reconstruction is achieved by linear interpolation in combination with a limiting procedure that explicitly prevents the formation of overshoots and undershoots. The reconstructed values are used in the second step to build a two-point Lipschitz monotone flux approximating $\vec{F}(c_l^k) \cdot \vec{n}_{lj}$.

The reconstruction step, following [13], proceeds as follows: for the triangle T_l with centroid x_l, three linear interpolants are built using the values of the nearby triangles, say T_p, T_q, T_r of Fig. 1. Denoting by x_j the pair of coordinates of the centroid of T_j, $j = l, r, p, q$, we construct L_l^1 as the linear interpolant of the points $\{(x_l, c_l^k), (x_p, c_p^k), (x_q, c_q^k)\}$, while L_l^2 is the linear interpolant of $\{(x_l, c_l^k), (x_q, c_q^k), (x_r, c_r^k)\}$ and L_l^3 is the linear interpolant of $\{(x_l, c_l^k), (x_r, c_r^k), (x_p, c_p^k)\}$. If an edge of T_l is on the boundary, the value of c on the midpoint of that edge is used instead of the centroid value in the linear interpolation.

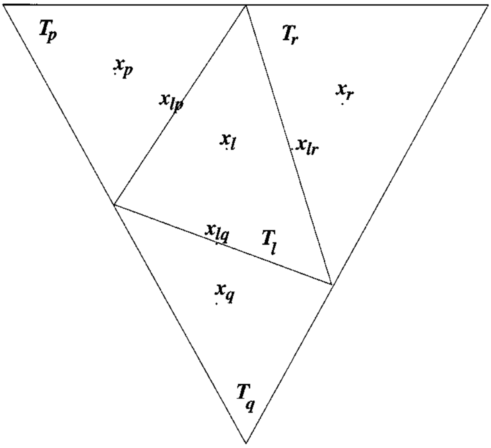

FIG. 1. Sample triangulation grid.

Once the three interpolants are calculated, the magnitude of the gradient of L_l^j can be expressed as

$$\left|\nabla L_l^j\right| = \sqrt{\left(\frac{\partial}{\partial x} L_l^j\right)^2 + \left(\frac{\partial}{\partial y} L_l^j\right)^2}, \qquad j = 1, 2, 3. \tag{13}$$

Starting from the L_l^j with maximum gradient and going toward the L_l^j with minimum gradient, we choose the first j such that

$$\begin{aligned} &L_l^j(x_{lp}) \text{ is between } c_l^k \text{ and } c_p^k\\ &L_l^j(x_{lq}) \text{ is between } c_l^k \text{ and } c_q^k\\ &L_l^j(x_{lr}) \text{ is between } c_l^k \text{ and } c_r^k, \end{aligned}$$

where x_{lp} is the midpoint of the edge sharing T_l and T_p, and so on. If no L_l^j satisfies these three requirements, we compute the local upper bound UB_l and the local lower bound LB_l of triangle T_l. They are defined, respectively, as the maximum and the minimum of the concentration values at the centroids of the triangles that have at least a common point with the triangle T_l. Starting again from L_l^j with maximum gradient we choose the first j such that L_l^j satisfies

$$\begin{aligned} UB_l &\geq \max\bigl(L_l^j(x_{lp}), L_l^j(x_{lq}), L_l^j(x_{lr})\bigr)\\ LB_l &\leq \min\bigl(L_l^j(x_{lp}), L_l^j(x_{lq}), L_l^j(x_{lr})\bigr). \end{aligned}$$

If no interpolant satisfies these inequalities, we choose as interpolant a piecewise constant reconstruction, that is, L_l assumes a constant value equal to c_l^k.

Once the linear interpolation L_l is obtained, the reconstructed values at the midpoints of each edge of T_l from inside and outside the triangle, i.e., $L_l(x_{lj}^{\text{in}})$ and $L_l(x_{lj}^{\text{out}})$, respectively, are the boundary conditions for the local Riemann problem. The line integral $\int_{e_{lj}} \vec{F}(c_l^k) \cdot \vec{n}_{lj}\, \mathrm{d}\Gamma$ is approximated using the midpoint formula by $h_j(L_l(x_{lj}^{\text{in}}), L_l(x_{lj}^{\text{out}}))|e_j|$, where h_j is the Godunov flux, and $|e_j|$ is the length of e_{lj}. The spatially second order accurate discrete approximation to (11) is then

$$\phi_l^{k+1} c_l^{k+1} = \phi_l^k c_l^k - \frac{\Delta t}{|T_l|} \sum_{j=1}^{3} h_j \big(L_l\big(x_{lj}^{\text{in}}\big), L_l\big(x_{lj}^{\text{out}}\big)\big)|e_j|. \tag{14}$$

2.1.2. MHFE discretization. In the dispersion step, our implementation of the MHFE applied to the discretization of (10) produces the following system of linear equations,

$$\int_{T_l} D_l^{-1} \vec{G}_l \cdot \vec{w}_{il}\, \mathrm{d}\Delta - \int_{T_l} c\vec{\nabla} \cdot \vec{w}_{il}\, \mathrm{d}\Delta + \int_{\partial T_l} \lambda \vec{w}_{il} \cdot \vec{n}_l\, \mathrm{d}\Gamma = 0 \tag{15}$$

$$\frac{\phi_l^{k+1}|T_l|c_l^{k+1}}{\Delta t_d} + \int_{T_l} \vec{\nabla} \cdot \vec{G}_l\, \mathrm{d}\Delta = \frac{\phi_l^k|T_l|c_l^k}{\Delta t_d} + \int_{T_l} f_l\, \mathrm{d}\Delta \tag{16}$$

$$\int_{e_j} \vec{G}_l \cdot \vec{n}_l\, \mathrm{d}\Gamma + \int_{e_j} \vec{G}_r \cdot \vec{n}_r\, \mathrm{d}\Gamma = 0 \qquad \text{if } e_j \in T_l \cap T_r \tag{17}$$

$$\int_{e_j} \vec{G}_l \cdot \vec{n}_l\, \mathrm{d}\Gamma = b_N \qquad \text{if } e_j \in \Gamma_N \cap T_l \tag{18}$$

$$\lambda_j = b_D \qquad \text{if } e_j \in \Gamma_D, \tag{19}$$

where $i = 1, 2, 3$, $l = 1, \ldots, m$, $j = 1, \ldots, n$, n being the number of edges. The quantities with subscript l are defined over element T_l, while $|T_l|$ denotes the area of T_l. The dispersive flux $\vec{G}$ is approximated for each T_l by

$$\vec{G}_l = \sum_{j=1}^{3} g_{jl}\vec{w}_{jl}, \qquad l = 1, \ldots, m, \tag{20}$$

where the $\vec{w}_{jl}$ are the discontinuous RT0 vector basis functions. The unknown Lagrange multiplier λ is expressed as

$$\lambda = \sum_{j=1}^{n} \lambda_j \mu_j, \tag{21}$$

where the μ_j are piecewise constant basis functions defined on edge e_j and λ_j represents the trace of the concentration on e_j.

In the above system, Eq. (15) is the MHFE discretization of the dispersive flux (4); Eq. (16) represents the discretized version of (2) without advective terms, as required by the time-splitting approach; Eq. (17) guarantees continuity of the normal flux across interelement edges, while Eqs. (18)–(19) are the explicitly imposed Neumann and Dirichlet boundary conditions, respectively.

The final hybrid formulation can be written in matrix notation as

$$\begin{pmatrix} A & -B & Q \\ B^T & P^{k+1} & 0 \\ Q^T & 0 & 0 \end{pmatrix} \begin{pmatrix} \mathbf{g}^{k+1} \\ \mathbf{c}^{k+1} \\ \boldsymbol{\lambda}^{k+1} \end{pmatrix} = \begin{pmatrix} 0 \\ \mathbf{f} + P^k\mathbf{c}^k \\ \mathbf{b}_N \end{pmatrix}, \tag{22}$$

where $A = \text{diag}[A_1, \ldots, A_m]$, $B = \text{diag}[B_1, \ldots, B_m]$, $P^k = \text{diag}[p_1^k, \ldots, p_m^k]$, and

$$\begin{aligned} A_l &= (a_{ik}) = \int_{T_l} D_l^{-1} \vec{w}_{il} \cdot \vec{w}_{kl}\, \mathrm{d}\Delta, \qquad B_l = (b_i) = \int_{T_l} \vec{\nabla} \cdot \vec{w}_{il}\, \mathrm{d}\Delta \\ Q &= (q_{rj}) = \int_{\partial T_l} \mu_j \vec{w}_{il} \cdot \vec{n}_l\, \mathrm{d}\Gamma, \qquad p_l^k = \phi_l^k |T_l| / \Delta t_d \\ \mathbf{g}^{k+1} &= (g_r) = g_{il}, \qquad \mathbf{f} = (\hat{f}_l) = \int_{T_l} f_l\, \mathrm{d}\Delta, \end{aligned} \tag{23}$$

where $i, k = 1, 2, 3$, $j = 1, \ldots, n$, $r = 3(l-1) + i$, and $\mathbf{c}^{k+1} = (c_l)$, $\boldsymbol{\lambda}^{k+1} = (\lambda_j)$, and $\mathbf{b}_N = (b_{Nj})$ where b_{Nj} assumes a non-vanishing value only if there is a Neumann condition on a boundary edge e_j.

$$\begin{pmatrix} T & W \\ W^T & 0 \end{pmatrix} \begin{pmatrix} \boldsymbol{\alpha} \\ \boldsymbol{\lambda}^{k+1} \end{pmatrix} = \begin{pmatrix} \boldsymbol{\beta} \\ \mathbf{b}_N \end{pmatrix}, \tag{24}$$

where

$$T = \begin{pmatrix} A & -B \\ B^T & P^{k+1} \end{pmatrix}, \qquad \boldsymbol{\alpha} = \begin{pmatrix} \mathbf{g}^{k+1} \\ \mathbf{c}^{k+1} \end{pmatrix}, \qquad \boldsymbol{\beta} = \begin{pmatrix} 0 \\ \mathbf{f} + P^k \mathbf{c}^k \end{pmatrix}, \qquad W = \begin{pmatrix} C \\ 0 \end{pmatrix}. \tag{25}$$

Matrix T is easily invertible because submatrices A and B are block-diagonal with 3×3 and 3×1 blocks, respectively, and P^{k+1} is diagonal. The solution of (24) can be obtained by means of the Schur complement with respect to T:

$$\begin{pmatrix} I & T^{-1} \\ 0 & -W^T T^{-1} W \end{pmatrix} \begin{pmatrix} \boldsymbol{\alpha} \\ \boldsymbol{\lambda}^{k+1} \end{pmatrix} = \begin{pmatrix} T^{-1} \boldsymbol{\beta} \\ \mathbf{b}_N - W^T T^{-1} \boldsymbol{\beta} \end{pmatrix}. \tag{26}$$

The second equation defines a system of linear equations in the multipliers only:

$$W^T T^{-1} W \boldsymbol{\lambda}^{k+1} = W^T T^{-1} \boldsymbol{\beta} - \mathbf{b}_N. \tag{27}$$

Its expression in terms of the original matrices and vectors can be obtained by noting that $W^T T^{-1} W = Q^T T_{11}^{-1} Q$ and $W^T T^{-1} \boldsymbol{\beta} = Q^T T_{12}^{-1} (\mathbf{f} + P^k \mathbf{c}^k)$. Denoting by

$$H = P^{k+1} + B^T A^{-1} B, \qquad S = A^{-1} B, \qquad M = A^{-1} - S H^{-1} S^T \tag{28}$$

so that

$$T^{-1} = \begin{pmatrix} M & S H^{-1} \\ -H^{-1} S^T & H^{-1} \end{pmatrix} \tag{29}$$

the final system for the multipliers reads

$$Q^T M Q \boldsymbol{\lambda}^{k+1} = Q^T S H^{-1} (\mathbf{f} + P^k \mathbf{c}^k) - \mathbf{b}_N. \tag{30}$$

The matrix $Q^T M Q$ is symmetric and positive definite and thus Eq. (30) can be solved using a preconditioned conjugate gradient scheme to obtain $\boldsymbol{\lambda}^{k+1}$. The unknown $\mathbf{c}^{k+1}$ can be computed in view of (26) by

$$\mathbf{c}^{k+1} = -T_{21}^{-1} Q\boldsymbol{\lambda} + T_{22}^{-1}(\mathbf{f} + P^k \mathbf{c}^k) = H^{-1} S Q \boldsymbol{\lambda}^{k+1} + H^{-1}(\mathbf{f} + P^k \mathbf{c}^k). \tag{31}$$

2.2. *Boundary Conditions*

The intrinsic nature of the time-splitting approach requires careful implementation of boundary conditions. In this respect, however, we are facilitated by the fact that in groundwater contaminant transport problems only a limited variety of boundary conditions is physically admissible (e.g., in general no boundary layers occur). To better describe how the boundary conditions are implemented in the proposed approach we distinguish between inflow and outflow boundaries. Inflow boundaries are characterized by having flow velocity normal components $\vec{v} \cdot \vec{n}$ directed inside the domain. Dirichlet or Cauchy boundary conditions may be of use in this situation. In the special case of $\vec{v} \cdot \vec{n} = 0$, Neumann type boundary conditions can also be employed. Implementation of these types of boundary conditions in the time splitting algorithm is obtained by specifying Dirichlet-type boundary conditions in the advective step and Neumann-type boundary conditions in the dispersive step. For example, inflow from a distributed source of contaminant can be specified as

$$\vec{v}c \cdot \vec{n} = \vec{v}c_1 \cdot \vec{n} \Rightarrow \text{i.e., Dirichlet b. c. } c = c_1 \text{ for the advection step}$$
$$-D\vec{\nabla}c \cdot \vec{n} = 0 \Rightarrow \text{i.e., zero Neumann flux for the dispersion step.}$$

Outflow boundaries are characterized by outgoing velocities and are easily implemented by imposing in the dispersion equation zero Neumann fluxes, as the outgoing advective flux is governed only by the velocity field.

Other type of conditions that may occur concern the presence of internal injection or extraction wells. Also in this case inflow or outflow are governed by the flow field and possible boundary conditions are easily implemented by Dirichlet plus zero Neumann and by zero Neumann conditions for injection and extraction wells, respectively.

3. NUMERICAL RESULTS

The behavior of the proposed numerical scheme can be characterized as a function of two grid related dimensionless numbers, the Courant–Friedrichs–Lewy (CFL) number and the Peclet (Pe) number. The CFL number can be defined for each triangle T_l as [13]

$$\text{CFL} = \Delta t_a \sup \frac{\overline{T_l}}{|T_l|} \sup \left| \frac{d\vec{F}}{dc} \right|, \tag{32}$$

where $\overline{T_l}$ and $|T_l|$ denote the perimeter and the area of T_l, respectively. Stability of the FV scheme requires that $\text{CFL} \le \frac{1}{3}$. The Peclet number represents the ratio between the advective and the dispersive term and can be defined in our case as [16]

$$\text{Pe} = \frac{\text{CFL}}{\gamma}, \tag{33}$$

where the dispersion number γ is given by

$$\gamma = |D| \Delta t_a \sup \frac{1}{|T_l|} \tag{34}$$

and $|D|$ is the norm of tensor D. Low Peclet numbers indicate that dispersion is predominant over advection, and vice versa.

3.1. *One-Dimensional Tests*

The numerical convergence rate of the time-splitting technique is tested on a one-dimensional model problem solved in a two-dimensional grid system. We consider the partial differential equation describing the movement of a tracer in a semi-infinite column and simulate it on a rectangular domain of unit length, with $\vec{v} = (v, 0)$ and $D = \mathrm{diag}(D_1, D_1)$. The boundary conditions $c = 1$ at $x = 0$ and $c = 0$, for $x = \infty$ are imposed. Zero concentration is used as the initial condition. This situation is simulated numerically by employing a grid of unitary length and making sure that at the time at which the relative error is evaluated the solution vanishes naturally at the right boundary. The analytical solution to this problem is [1]

$$c(x,t) = \frac{1}{2}\left(\mathrm{erfc}\frac{x - vt}{2\sqrt{D_1 t}} + \exp\frac{vx}{D_1}\cdot \mathrm{erfc}\frac{x + vt}{2\sqrt{D_1 t}}\right). \tag{35}$$

The numerical convergence behavior of the scheme is evaluated by calculating errors at different grid levels. For a given level l, the error norm is calculated as

$$|e_l| = \frac{\sqrt{\sum_{l=0}^{m}\left(c\left(x_l, t^k\right) - c_l^k\right)^2}}{\sqrt{\sum_{l=0}^{m} c\left(x_l, t^k\right)^2}}, \tag{36}$$

where $c(x_l, t^k)$ is the analytical solution on the centroid of T_l at time t^k and c_l^k is the corresponding numerical solution. For all the subsequent test runs we consider the solution at $t^k = 0.1$ s.

Five grid levels are used and defined as follows. At the coarsest level ($l = 1$) the rectangular domain is discretized into three layers of rectangular elements that are further subdivided into two triangles. The refined triangulations ($l = 2, \ldots, 5$) are obtained by connecting the midpoints of the three edges of each triangle. To reduce the dimensionality of the mesh, the height of the domain is always halved in passing from a coarser to the next finer level, in such a way that the shape of the triangles at the different levels is preserved. The coarsest mesh is defined on the rectangle $[0, 1] \times [0, 0.1]$ and is characterized by 300 triangles and 204 edges, while the finest level ($l = 5$) is defined on the rectangle $[0, 1] \times [0, 6.25 \times 10^{-3}]$ and is characterized by 4800 triangles and 3204 edges. In the case of constant coefficients, Pe decreases by a factor of 2 in passing from a coarser to a finer level.

The first set of simulations is aimed at numerically verifying the theoretical convergence rate of the time splitting scheme under different Pe and CFL numbers. Second order convergence rate can be observed on a problem with smooth solution (small Pe number) employing $\Delta t_d = \Delta t_a = \Delta x^2$. Table I reports the errors and convergence rates at the different levels for a case with $D_1 = D_2 = 1 \times 10^{-2}$ m^2/s and $v = 1$ m/s. Correspondingly the grid Peclet number varies between 9.22 ($l = 1$) to 0.58 ($l = 5$), while CFL goes to zero. First order convergence rate is instead achieved when $\Delta t = \mathcal{O}(\Delta x)$. Table II reports the results to the same problem obtained with constant CFL numbers (0.28 and 0.14, respectively) using a constant $\Delta t_d = \Delta t_a$. Convergence is still superlinear but seems to tend asymptotically to first order, as predicted by the theory.

TABLE I
One-Dimensional Example: Convergence Behavior for $\Delta t_d = \Delta t_a = \Delta x^2$

l	Pe	$\lvert e_l \rvert$	Rate
1	9.22	1.24e-2	
2	4.61	3.67e-3	1.76
3	2.31	8.01e-4	1.98
4	1.15	2.07e-4	1.97
5	0.58	7.13e-5	1.86

The second set of simulations is aimed at determining the best time stepping strategy, i.e., the number n_a of advective time steps per dispersive time step, for which the error remains reasonably small and CPU time is minimal. It is intuitive to think that the behavior of the time splitting approach depends on the given Peclet number. For small Pe, i.e., dominant dispersion, one expects convergence to be mainly driven by the MHFE technique discretizing the dispersion terms. The transient behavior of the solution should be well captured even for $n_a = 1$, i.e., $\Delta t_a = \Delta t_d$. On the other hand, for large Pe, the advective terms become important and thus the advective transient has to be accurately captured. We expect for this case the best accuracy when $\Delta t_a < \Delta t_d$, or $n_a > 1$. Verification of this behavior is obtained for a given Peclet number by comparing errors $|e_l|$, as given in Eq. (36), and CPU times for different values of n_a on a fixed mesh level. For this purpose we chose the mesh with 1200 triangles ($l = 3$). The dispersion coefficient varies in the range $D = 2 \times 10^{-2} \div 0.5 \times 10^{-4}$ m^2/s, while velocity is kept constant at $v = 0.5$ m/s. These values correspond to Peclet numbers varying from 0.28, a dispersion dominated problem, to 115, a convection dominated case. The results of the different simulations are reported in Tables III to VI. Each column of the tables contains the results ($|e_3|$, and CPU times in second) for a fixed Δt value and for the different n_a values tested. Subsequent columns (rows) are characterized by double Δt_d (n_a) values. The CFL number as well as Δt_a are thus constant along the main diagonals of the tables. For example, in Table III the advective time step is the same ($\Delta t_a = 0.25 \times 10^{-1}$ s) for the three cases $\Delta t_d = 0.25 \times 10^{-3}$ s and $n_a = 1$, $\Delta t_d = 0.5 \times 10^{-3}$ s and $n_a = 2$, and $\Delta t_d = 1 \times 10^{-3}$ s and $n_a = 4$.

When dispersion dominates, i.e., Pe $= 0.28$, the accuracy of the scheme is mainly influenced by the size of Δt_d, as can be seen in Table III. First order convergence rate can be seen in every row where the values increase linearly with Δt_d. It is worth noting that the relative

TABLE II
One-Dimensional Example: Convergence Behavior for $\Delta t_d = \Delta t_a$

l	Pe	$\lvert e_l \rvert$	Rate	l	Pe	$\lvert e_l \rvert$	Rate
1	9.22	1.47e-2		1	9.22	1.27e-2	
2	4.61	5.40e-3	1.44	2	4.61	4.25e-3	1.58
3	2.31	1.84e-3	1.50	3	2.31	1.14e-3	1.74
4	1.15	7.67e-4	1.42	4	1.15	4.02e-4	1.66
5	0.58	3.54e-4	1.34	5	0.58	1.65e-4	1.57

Note. CFL $= 0.28$ (left) and CFL $= 0.14$ (right).

TABLE III

One-Dimensional Example: Relative Error Norm $\lvert e_3\rvert$ and CPU Times for Pe = 0.28

Δt_d	0.25×10^{-3}		0.5×10^{-3}		1×10^{-3}	
n_a	$\lvert e_3\rvert \times 10^3$	CPU (s)	$\lvert e_3\rvert \times 10^3$	CPU (s)	$\lvert e_3\rvert \times 10^3$	CPU (s)
1	0.69	35.94	1.42	20.20	2.87	12.07
2	0.56	38.20	1.13	21.51	2.30	12.71
4	0.50	44.16	1.00	22.60	2.01	13.95

TABLE IV

One-Dimensional Example: Relative Error Norm $\lvert e_3\rvert$ and CPU Times for Pe = 2.88

Δt_d	0.25×10^{-3}		0.5×10^{-3}		1×10^{-3}		2×10^{-3}	
n_a	$\lvert e_3\rvert \times 10^3$	CPU (s)	$\lvert e_3\rvert \times 10^3$	CPU (s)	$\lvert e_3\rvert \times 10^3$	CPU (s)	$\lvert e_3\rvert \times 10^3$	CPU (s)
1	2.21	29.28	4.06	14.77	7.91	7.72		
2	1.55	32.17	2.46	16.06	4.42	8.43	8.57	4.67
4	1.23	38.09	1.76	19.08	2.86	9.90	5.25	5.44
8	1.10	50.16	1.43	25.10	2.13	12.83	3.52	6.78
16			1.31	36.84	1.80	18.48	2.62	9.65
32					1.62	30.25	2.30	15.18

TABLE V

One-Dimensional Example: Relative Error Norm $\lvert e_3\rvert$ and CPU Times for Pe = 28.8

	1×10^{-3}		2×10^{-3}		4×10^{-3}		8×10^{-3}		16×10^{-3}		32×10^{-3}	
Δt_d	$\lvert e_3\rvert$	CPU	$\lvert e_3\rvert$	CPU	$\lvert e_3\rvert$	CPU	$\lvert e_3\rvert$	CPU	$\lvert e_3\rvert$	CPU	$\lvert e_3\rvert$	CPU
n_a	$\times 10^2$	(s)	$\times 10^2$	(s)	$\times 10^2$	(s)	$\times 10^2$	(s)	$\times 10^2$	(s)	$\times 10^2$	(s)
1	4.62	7.19										
2	2.25	7.98	4.70	3.98								
4	1.20	9.54	2.24	4.70	4.75	2.35						
8	0.78	12.55	1.19	6.21	2.26	3.08	4.47	1.66				
16	0.66	18.60	0.79	9.25	1.21	4.59	2.29	2.42	4.11	1.23		
32	0.61	37.80	0.63	15.37	0.79	7.59	1.28	3.94	2.30	2.02	3.76	0.99
64	0.60	55.56	0.59	27.40	0.63	13.52	0.86	7.17	1.36	3.50	2.38	1.73
128			0.59	51.01	0.59	25.50	0.70	13.48	0.99	6.65	1.54	3.08

TABLE VI

One-Dimensional Example: Relative Error Norm $\lvert e_3\rvert$ and CPU Times for Pe = 115

	1×10^{-3}		2×10^{-3}		4×10^{-3}		8×10^{-3}		16×10^{-3}		32×10^{-3}	
Δt_d	$\lvert e_3\rvert$	CPU	$\lvert e_3\rvert$	CPU	$\lvert e_3\rvert$	CPU	$\lvert e_3\rvert$	CPU	$\lvert e_3\rvert$	CPU	$\lvert e_3\rvert$	CPU
n_a	$\times 10^2$	(s)	$\times 10^2$	(s)	$\times 10^2$	(s)	$\times 10^2$	(s)	$\times 10^2$	(s)	$\times 10^2$	(s)
1	6.06	7.28										
2	2.67	8.03	5.62	4.00								
4	1.78	9.51	2.64	4.71	5.85	2.32						
8	1.90	12.33	1.74	6.16	2.92	3.06	5.48	1.58				
16	2.10	18.39	1.88	9.03	1.60	4.50	2.66	2.33	4.46	1.16		
32			2.10	14.90	1.80	7.45	1.44	3.82	2.39	1.91	3.57	0.97
64					2.02	13.27	1.65	6.75	1.63	3.41	1.88	1.71
128									1.66	6.33	1.56	3.20
256											1.81	6.19

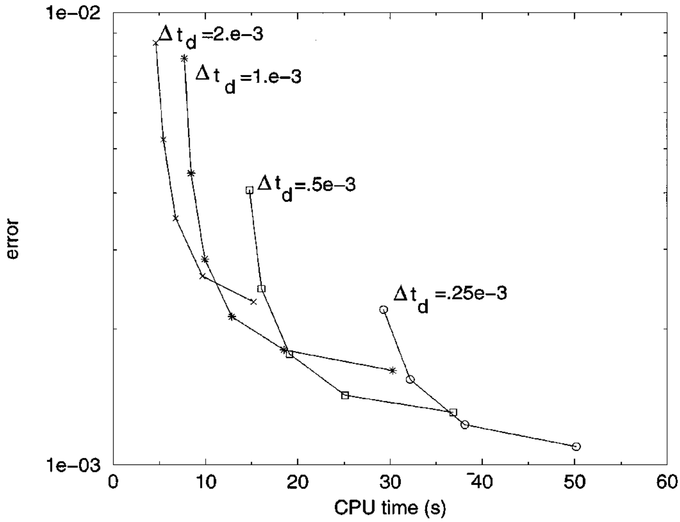

FIG. 2. Relative accuracy vs CPU time for different Δt_d and fixed Peclet number = 2.88.

error significantly decreases from the case with $n_a = 1$ to $n_a = 4$ (by about 30%), whereas the CPU time increases only by 15% on the average. This fact suggests using $n_a > 1$ also for very small Peclet numbers.

At Pe = 2.88, an intermediate value of the Peclet number (Table IV), we still recognize first order convergence rate along the rows. However, the error decrease along the diagonals is now much less pronounced, indicating that in this case small Δt_a are needed to maintain accuracy. These results are exemplified graphically in Fig. 2 where four plots of $|e_3|$ vs CPU time corresponding to four values of Δt_d are reported. For each curve, the data points refer to the n_a values of Table IV. Hence, the optimal $(\Delta t_d, n_a)$ combination can be found on the intersection of the envelope of the curves with the horizontal line corresponding to the desired accuracy. Obviously, there is no unique strategy to choose the optimal Δt values, however, a few observations can be helpful for this purpose. Efficiency reasons demand that Δt_d be not too small, so as to minimize the number of linear system solutions. On the other hand, accuracy at this level of Peclet numbers already requires small Δt_a, and thus large $n_a (>4)$. For example, a reasonable choice for our test case could be $\Delta t_d = 10^{-3}$ and $n_a = 16$.

At Pe = 28.8 advection starts to dominate over dispersion. The value of n_a is now important, as can be seen from the significant error decrease in the columns of Table V. We also note that the error remains almost constant along the diagonals, i.e., for constant Δt_a and increasing Δt_d. Note that, for each Δt_d value, the increase in accuracy tends to be smaller as n_a increases, suggesting that after a certain value of n_a, convergence tends to stagnate. From this observation, we may argue that the truncation error of the scheme is proportional to Δt_d and to Δt_a,

$$|\epsilon_T| \approx O\left(\Delta t_d, \Delta x^2\right) + O\left(\Delta t_a, \Delta x^2\right) = \left|\epsilon_{T_d}\right| + \left|\epsilon_{T_a}\right|. \tag{37}$$

With this model we have that, for constant Δt_d, $\lim_{n_a \to \infty} |\epsilon_T| = \lim_{\Delta t_a \to 0} |\epsilon_T| = |\epsilon_{T_d}|$ explaining the experimented numerical behavior.

At even higher Peclet numbers (Table VI, Pe = 115) the same behavior can be observed with one notable exception. For constant Δt_d the error attains a minimum value for a specific

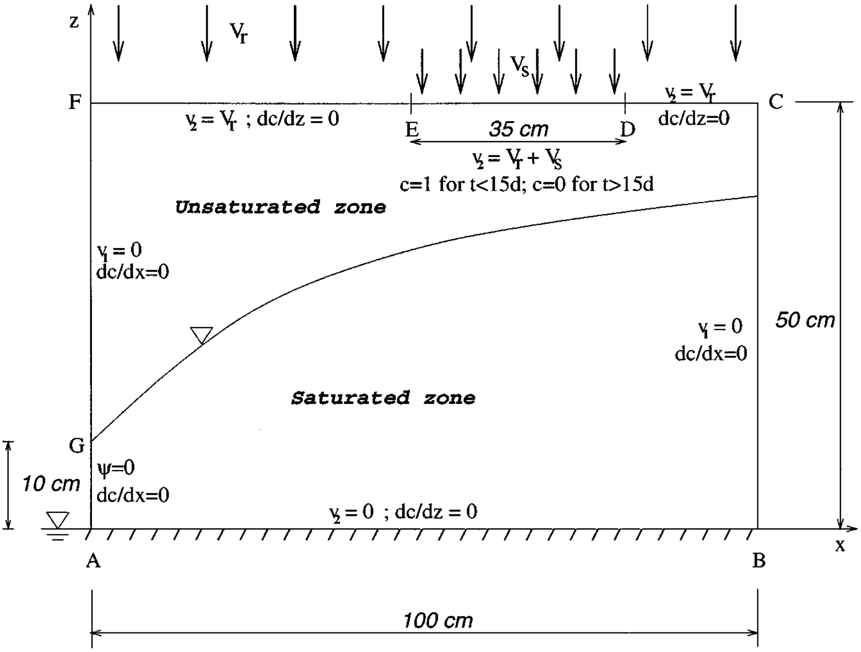

FIG. 3. Schematic description of the domain and boundary conditions for the two-dimensional test case.

n_a. This behavior suggests that errors due to the operator splitting technique accumulate with the advective time step and thus $\epsilon_{T_a} = \epsilon_{T_a}(n_a)$.

3.2. *Two-Dimensional Infiltration of Chloride Ion in a Surface Aquifer*

The applicability of the proposed approach is shown on a realistic two-dimensional problem of infiltration of a conservative contaminant into a saturated–unsaturated surface aquifer. The test case considers a ditched-drained aquifer with incident steady rainfall and trickle infiltration of chloride ion [11]. The geometry of the domain and the boundary conditions employed in the solution of the flow and transport problems are described in Fig. 3. The boundary conditions imposed along face AG correspond to the presence of a seepage

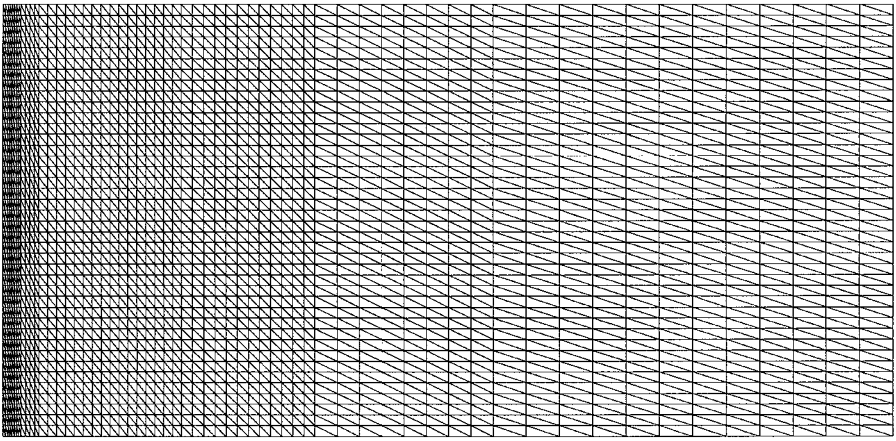

FIG. 4. Computational mesh used in the two-dimensional test case.

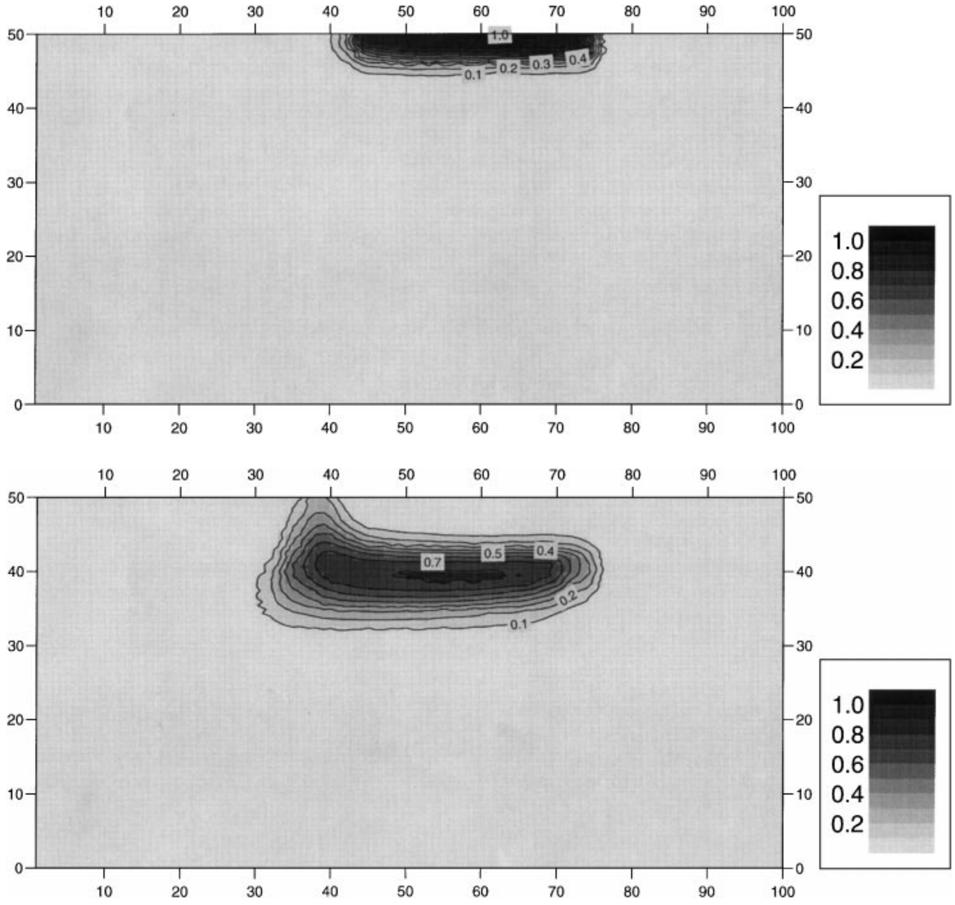

FIG. 5. Two-dimensional test-case: concentration contours at 14.7 days (top) and 45.2 days (bottom).

face and thus to an outflow condition from which the aquifer is drained. The saturated–unsaturated flow equation is solved in steady state conditions by means of a Richards' equation solver based on the mixed hybrid finite element method [2]. The physical parameters of the simulation assume the following values: $V_r = 0.1$ cm/d, $V_s = 0.05$ cm/d (Fig. 3), $K_s = 1$ cm/d. The moisture retention curves of [12] are used with the following parameter values: $\alpha = 0.015$, $\beta = 2$, $\gamma = 3$, $a = 2$, $b = 3.5$, $\psi_s = -10$ cm, $S_{wr} = 0.01$. The Darcy velocity field $\vec{v} = (v_x, v_y)^T$ and water saturation S_w values as calculated from the solution of the flow problem are used in a 120-day simulation of the transport of the chloride ion. For this latter problem we have used a dispersion tensor $D = \mathrm{diag}(D_1, D_2)$ as given by [1]: $D_1 = \alpha_L|\vec{v}| + nS_w D_0$ and $D_2 = \alpha_T|\vec{v}| + nS_w D_0$ where $|\vec{v}| = \sqrt{v_x^2 + v_y^2}$, $\alpha_L = 0.5$ cm is the longitudinal dispersivity, $\alpha_T = 0.1$ cm is the transverse dispersivity, $\phi = 0.30$ is the porosity of the medium, and $D_0 = 1.e - 06$ cm^2/s is the molecular diffusion coefficient.

The mesh employed is made up of 2501 nonuniform triangles and 4800 edges (Fig. 4). The time step sizes are $\Delta t_d = 0.5$ days together with $n_a = 50$. The simulations are characterized by $\mathrm{CFL} = 0.44$ and $\mathrm{Pe} = 5.55$. With this choice of parameters the advective phase of the time-splitting algorithm is approximately twice as expensive than the dispersive phase (0.73 and 0.41 s per time step, respectively, on a DEC Alpha-600 workstation).

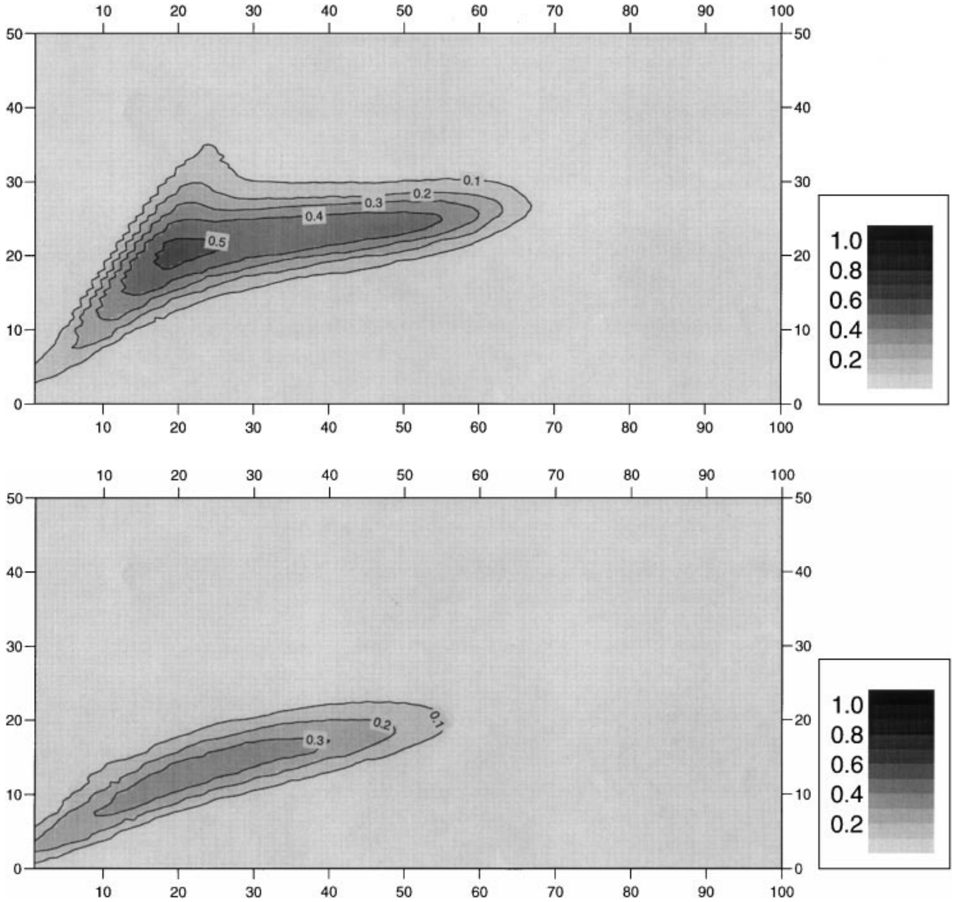

FIG. 6. Two-dimensional test case: concentration contours at 90.2 days (top) and 120 days (bottom).

In Figs. 5 and 6 the solute concentration contours at 14.7, 45.2, 90.2, and 120 days of simulation are shown. At the beginning the concentration plume infiltrates downwards with the unsaturated flow. Once it reaches the water table, it finds a more pronounced horizontal velocity in the saturated zone and starts moving towards the seepage face and exits the domain.

A few observations from the numerical standpoint are worth mentioning. The solution obtained with the proposed approach does not present oscillations in any part of the domain and at any time. This verifies that the property of the FV scheme of being TVD is retained in the time-splitting algorithm, which maintains monotonicity in all our simulations. The plume shape in Figs. 5 and 6 shows a front that is slightly steeper than the corresponding front calculated by standard Galerkin finite elements (GFE) with no upwind, as can be seen from Fig. 7 that shows the solution to the same problem at 45.2 days as obtained by GFE [10]. This confirms the fact that the combination of MHFE and FV introduces less numerical diffusion than standard FE. Finally, the mass balance in all simulations is satisfied within the accuracy of the linear system solution in the dispersive phase (i.e., 10^{-10}), in contrast with FE applications where mass balance errors of few percents are commonly observed. These characteristic features of the time-splitting algorithm developed in this paper do not change at larger Peclet numbers as long as the stability criteria for the advection step are satisfied.

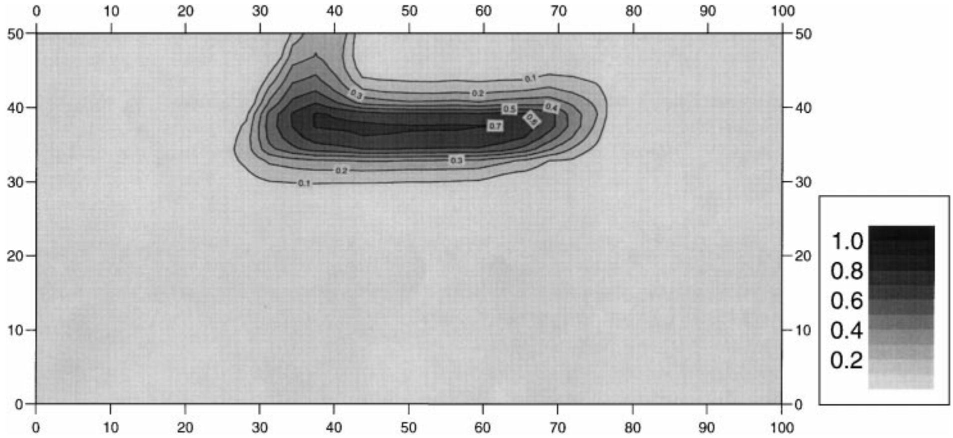

FIG. 7. Two-dimensional test-case: concentration contours at 45.2 days as obtained by the standard Galerkin FEM scheme.

4. CONCLUSIONS

We have presented and studied a first order accurate in time, second order accurate in space, time-splitting approach for the numerical solution of the two-dimensional advection-dispersion equation typically arising in groundwater contaminant transport simulations. The technique is based on the separate discretization of the dispersive and advective terms of the governing equation by means of the discontinuous RT0 mixed hybrid finite element technique and a high resolution Godunov-type finite volume method, respectively. Both these techniques are defined on triangles and use consistent functional approximations of concentrations. The final solution is obtained by means of a two-step time-splitting procedure employing explicit in time finite differences for the FV scheme and implicit time stepping for MHFE. The system of linear equations resulting from the implicit time stepping of the MHFE scheme is solved using a Schur complement decomposition to reduce the original system to a smaller positive definite linear system. The explicit character of the Godunov-type method is well suited for capturing sharp fronts with introduction of minimal artificial diffusion. Different time stepping strategies can be used for the two steps. In the dispersion step, the most computationally demanding, the time step size is not limited by stability constraints and its choice is dictated only by accuracy considerations. On the other hand, the advective step has to satisfy a typical CFL stability condition but remains stable across the entire range of Peclet numbers. To minimize the differences in the computational costs, several advective time steps can be performed in one single dispersive step.

Numerical tests on a one-dimensional sample problem are used to verify the theoretical properties of the proposed scheme and to experiment different time-stepping strategies for various Peclet numbers. The results show that the scheme is first order accurate in time while global second order accuracy in space is achieved for reasonably smooth solutions. Test cases run at different Peclet numbers show that an effective compromise between accuracy and computing time can be achieved when the ratio between the number of advective time steps per dispersive time step is always larger than one. This ratio increases at larger Peclet numbers.

A two-dimensional advection-dominated groundwater transport problem is presented to show the applicability of the proposed approach to realistic problems. The results of the simulation confirm that the time-splitting approach does not suffer from Peclet limitations, displaying only small amounts of numerical diffusion.

ACKNOWLEDGMENTS

This work has been supported by the EC Inco-Copernicus program (contract ERBIC15CT96-0211).

REFERENCES

1. J. Bear, *Hydraulics of Groundwater* (McGraw–Hill, New York, 1979).
2. L. Bergamaschi and M. Putti, Mixed finite elements and Newton-like linearization for the solution of Richard's equation, *Int. J. Numer. Methods Eng.* **45**(8), 1025 (1999).
3. V. Casulli, Eulerian–Lagrangian methods for hyperbolic and convection dominated parabolic problems, in *Computational Methods for Nonlinear Problems* (Pineridge, Swansea, 1987), p. 239.
4. C. N. Dawson, Godunov-mixed methods for immiscible displacement, *Int. J. Numer. Methods Fluids* **11**(7), 835 (1990).
5. C. N. Dawson, Godunov-mixed methods for advective flow problems in one space dimension, *SIAM J. Numer. Anal.* **28**(5), 1282 (1991).
6. C. N. Dawson, Godunov-mixed methods for advection-diffusion equations in multidimensions, *SIAM J. Numer. Anal.* **30**(5), 1315 (1993).
7. C. N. Dawson, High resolution upwind-mixed finite element methods for advection diffusion equations with variable time-stepping, *Numer. Methods PDE* **11**, 525 (1995).
8. J. Douglas, Jr., and J. E. Roberts, Global estimates for mixed methods for second order elliptic equations, *Math. Comp.* **44**, 39 (1985).
9. L. J. Durlofsky, B. Engquist, and S. Osher, Triangle based adaptive stencils for the solution of hyperbolic conservation laws, *J. Comput. Phys.* **98**, 64 (1992).
10. G. Gambolati, G. Pini, M. Putti, and C. Paniconi, Finite element modeling of the transport of reactive contaminants in variably saturated soils with LEA and non-LEA sorption, in *Environmental Modeling. Vol. II. Computer Methods and Software for Simulating Environmental Pollution and Its Adverse Effects*, edited by P. Zannetti (Computational Mechanics Publications, Southampton, UK, 1994), Chap. 7, p. 173.
11. A. B. Gureghian, *TRIPM, a Two-Dimensional Finite Element Model for the Simultaneous Transport of Water and Reacting Solutes through Saturated and Unsaturated Porous Media*, Technical Report ONWI-465, Off. of Nuclear Water Isolation, Columbus, 1983.
12. P. S. Huyakorn, S. D. Thomas, and B. M. Thompson, Techniques for making finite elements competitive in modeling flow in variably saturated porous media, *Water Resour. Res.* **20**(8), 1099 (1984).
13. X.-D. Liu, A maximum principle satisfying, modification of triangle based adaptive stencils for the solution of scalar hyperbolic conservation laws, *SIAM J. Numer. Anal.* **30**(3), 701 (1993).
14. S. P. Neuman, A. Eulerian–Lagrangian numerical scheme for the dispersion convection equation using conjugate space time grids, *J. Comput. Phys.* **41**, 270 (1981).
15. R. Peyret and T. D. Taylor, *Computational Methods for Fluid Flow* (Springer-Verlag, New York, 1983).
16. M. Putti, W. W.-G. Yeh, and W. A. Mulder, A triangular finite volume approach with high resolution upwind terms for the solution of groundwater transport equations, *Water Resour. Res.* **26**(12), 2865 (1990).