Available online at www.sciencedirect.com

SCIENCE DIRECT®

Journal of Computational Physics 208 (2005) 384–402

JOURNAL OF COMPUTATIONAL PHYSICS

www.elsevier.com/locate/jcp

Solving the advection-diffusion equation with the Eulerian–Lagrangian localized adjoint method on unstructured meshes and non uniform time stepping

A. Younes [a,b], P. Ackerer [b,*]

[a] *Laboratoire de Génie Industriel, Université de la Réunion, 15 Avenue René Cassin, BP 7151 – 97715, St. Denis Cedex 09, La Réunion, France*

[b] *Institut de Mécanique des Fluides et des Solides, Université Louis Pasteur, CNRS – UMR 7507, 2 Rue Boussingault, 67000 Strasbourg, France*

Received 24 August 2004; received in revised form 17 February 2005; accepted 17 February 2005
Available online 14 April 2005

Abstract

Eulerian–Lagrangian localized adjoint method (ELLAM) is used to solve the advection diffusion equation (ADE) which is a very common mathematical model in physics. In this work, ELLAM is extended to triangular meshes. Standard integration schemes, which perform well for rectangular grids, are improved to reduce oscillations with unstructured triangulations. Numerical experiments for grid Peclet numbers ranking from 1 to 100 show the efficiency of the developed scheme.

A new algorithm is also developed in order to avoid excessive numerical diffusion when using many time steps with the ELLAM. The basic idea of this approach is to keep the same characteristics for all time steps and to interpolate only the concentration variations due to the dispersion process at the end of each time step.

Although ELLAM requires a lot of integration points for unstructured meshes, it remains a competitive method when using a single or many time steps compared to explicit discontinuous Galerkin finite element method.
© 2005 Elsevier Inc. All rights reserved.

Keywords: ELLAM; ADE; Forward tracking; Triangular meshes; Unstructured meshes

1. Introduction

The advection-diffusion equation (ADE) is a very common mathematical model used in physics to simulate mass or energy transported by a fluid in movement. Its general formulation is:

* Corresponding author. Tel.: +33 390 242 909; fax: +33 388 614 300.
E-mail address: ackerer@imfs.u-strasbg.fr (P. Ackerer).

0021-9991/$ - see front matter © 2005 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2005.02.019

$$L(C) = \frac{\partial C}{\partial t} + \nabla \cdot (\boldsymbol{V}C) - \nabla \cdot (\boldsymbol{D} \cdot \nabla C) = 0, \quad \boldsymbol{x} \in \Omega, \ t \in [0, T] \tag{1}$$

with its associated initial and boundary conditions:

$$\begin{aligned}
&C(\boldsymbol{x}, 0) = C_0(\boldsymbol{x}) \quad (\boldsymbol{x} \in \Omega),\\
&C(\boldsymbol{x}, t) = g_1(\boldsymbol{x}, t) \quad (\boldsymbol{x} \in \partial\Omega^1, \ t > 0),\\
&(-\boldsymbol{D} \cdot \nabla C) \cdot \boldsymbol{n}_{\partial\Omega} = g_2(\boldsymbol{x}, t) \quad (\boldsymbol{x} \in \partial\Omega^2, \ t > 0),\\
&(\boldsymbol{V}C - \boldsymbol{D} \cdot \nabla C) \cdot \boldsymbol{\eta}_{\partial\Omega} = g_3(\boldsymbol{x}, t) \quad (\boldsymbol{x} \in \partial\Omega^3, \ t > 0),
\end{aligned} \tag{2}$$

where $C(\boldsymbol{x}, t)$ is the unknown state variable which in this work corresponds to the solute concentration, $\boldsymbol{V}$ the fluid velocity, $\boldsymbol{D}$ the diffusion/dispersion tensor, Ω a bounded, polygonal open set of $\mathbb{R}^2$, $\partial\Omega^1$, $\partial\Omega^2$ and $\partial\Omega^3$ are partitions of the boundary $\partial\Omega$ of Ω corresponding to Dirichlet, Neumann and total flux boundary conditions and $\boldsymbol{\eta}_{\partial\Omega}$ the unit outward normal to the boundary $\partial\Omega$.

In theory, the three types of boundary conditions can be applied at the inflow or the outflow boundary. However, in practice, Dirichlet boundary conditions or total flux boundary conditions at the inflow are commonly used. At the outflow, the Neumann condition with no dispersive flux is almost exclusively used. Therefore, only these configurations will be detailed in this work.

Classical Eulerian methods such as finite element (FE) or finite volume (FV) methods are not suitable since they often generate numerical solutions with artificial diffusion and/or non-physical oscillations for convective dominated transport.

Characteristic methods use a Lagrangian treatment of the advective part by tracking particles along the characteristics and an Eulerian treatment on the fixed grid of the dispersive part of the transport equation. Characteristic methods generate accurate solutions even if large time steps are used. However, they fail to conserve mass and are unable to rigorously treat boundary conditions [11].

The Eulerian–Lagrangian localized adjoint method (ELLAM), introduced by [8], is well adapted to transport problems. The ELLAM uses space-time test functions. It preserves the performance of characteristic methods and treats general boundary conditions naturally in their formulations.

Different ELLAM schemes for one-dimensional ADE were investigated [2,7,11,20,24,27]. ELLAM has been extended to include two spatial dimensions and variable coefficients [5,12,25], reaction terms [9,24], nonlinear equation [22,24] and point sources or sinks [13,26,28]. The method has also been extended to three-dimensions [6,18]. A review of the researches done on the ELLAM is presented by Russell and Celia [17].

In the literature, all numerical results are presented on uniform rectangular or block meshes. However, in two-dimensions and for practical problems, triangular meshes are preferred since they can handle domains with complex geometry. Moreover, triangular meshes are often unstructured and can contain some subdomains with a much finer discretization than the rest of the domain.

Hence, the first objective of this work is to develop an efficient implementation of the ELLAM for a general triangulation.

It is also known that Eulerian–Lagrangian methods perform well for problems in which they can successfully use a large time step, but when using several time steps, they suffer from numerical dispersion introduced by interpolation at each time step [16]. This phenomenon was avoided for the one-dimensional problem by Younes [27] since ELLAM was combined with a moving grid procedure. This approach cannot be extended in the same way to two- or three-dimensions. Indeed, characteristics in one dimension are not necessarily straight lines but are nevertheless ordered. This property has no meaning for multidimensional problems. Therefore, the second objective of this work is to develop an efficient implementation of the ELLAM that avoids problems due to interpolation.

In the first part of this paper, we recall the mathematical developments of the ELLAM for solving the two-dimensional ADE on triangular meshes. Numerical integration and improved implementation methods are described in the second part of the paper. Their accuracy and performance are compared to standard implementations in the third part of the paper. In the last part of this work, the robustness of the suggested new methods is evaluated using different Peclet numbers and different time and space discretizations. The ELLAM are also compared to another numerical model based on a combination of discontinuous Galerkin/mixed finite element in order to evaluate the efficiency of the developed ELLAM for unstructured meshes and grid Peclet numbers ranking from 1 to 100.

2. The Eulerian–Lagrangian localized adjoint method for ADE

2.1. ELLAM formulation

The weak formulation of Eq. (1), using space–time test function $\omega = \omega(\boldsymbol{x}, t)$ gives:

$$\int_0^{\mathrm{T}} \int_\Omega L(C)\omega = \int_0^{\mathrm{T}} \int_\Omega \left[\frac{\partial C}{\partial t}\omega + \nabla \cdot (\boldsymbol{V}C)\omega - \nabla \cdot (\boldsymbol{D} \cdot \nabla C)\omega\right] \mathrm{d}\boldsymbol{x}\,\mathrm{d}t = 0 \tag{3}$$

which can be written as following

$$\int_0^{\mathrm{T}} \int_\Omega L(C)\omega = \int_0^{\mathrm{T}} \int_\Omega \left[\frac{\partial (C\omega)}{\partial t} - C\frac{\partial \omega}{\partial t} + \nabla \cdot (\boldsymbol{V}C\omega) - \boldsymbol{V}C \cdot \nabla\omega - \nabla.(\boldsymbol{D} \cdot \nabla C\omega) + (\boldsymbol{D} \cdot \nabla C) \cdot \nabla\omega\right] \mathrm{d}\boldsymbol{x}\,\mathrm{d}t = 0. \tag{4}$$

The concentration C is defined in a finite element framework with piecewise linear basis functions

$$C = \sum_{i=1}^{N} C_i \psi_i \quad \text{with} \quad \psi_i = \sum_{A \supset i}^{NE} \varphi_i^A, \tag{5}$$

where N is the total number of nodes and NE the number of elements on Ω.

For the triangle A formed by the three vertices (i,j,k), we have

$$\varphi_i^A = \begin{cases} [(x_j y_k - x_k y_j) + (y_j - y_k)x + (x_k - x_j)y]/(2S^A) & \text{for } (x,y) \in A, \\ 0 & \text{elsewhere,} \end{cases} \tag{6}$$

where S^A is the area of the element A.

The ELLAM selects $\omega(\boldsymbol{x}, t)$ so that for $t \in [t^n, t^{n+1}]$,

$$\frac{D\omega}{Dt} = \frac{\partial \omega}{\partial t} + \boldsymbol{V} \cdot \nabla\omega = 0, \tag{7}$$

where $\omega(\boldsymbol{x}, t)$ is discontinuous at $t = t^n$, and $\omega\ (\boldsymbol{x},t) = 0$ for $t \notin [t^n, t^{n+1}]$.

With this choice of ω, the ADE to solve becomes (for details, see Binning and Celia, [6] among others)

$$\int_\Omega C^{n+1}\omega^{n+1}(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} + \int_0^{\mathrm{T}} \int_\Omega (\boldsymbol{D} \cdot \nabla C) \cdot \nabla\omega\,\mathrm{d}\boldsymbol{x}\,\mathrm{d}t + \int_0^{\mathrm{T}} \int_{\partial\Omega} [(\boldsymbol{V}C - \boldsymbol{D} \cdot \nabla C)\omega] \cdot \boldsymbol{n}_{\partial\Omega}\,\mathrm{d}\boldsymbol{x}\,\mathrm{d}t = \int_\Omega C^n \omega^n(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}. \tag{8}$$

The test functions ω_i are constant along characteristics since they satisfy [7] and are defined only for $t \in [t^n, t^{n+1}]$. At $t = t^{n+1}$, following Russel and Celia [17] and Binning and Celia [6], who found that the linear functions should be preferred since they are less sensitive to oscillations, ω_i is a piecewise linear function (chapeau function) defined by

$$\omega_i(\boldsymbol{x}_j, t^{n+1}) = \begin{cases} 1 & \text{if} \quad i = j, \\ 0 & \text{if} \quad i \neq j, \end{cases} \tag{9}$$

where $\boldsymbol{x}_j$ is the location of the vertice j.

Notice that at $t = t^{n+1}$, we have $\psi_i = \omega_i^{n+1}$.

Due to Eq. (7), $\omega_i(\boldsymbol{x}, t^n)$ can be computed analytically for simple one dimensional configuration and numerically by particle tracking for more general configurations.

Backtracking or forward tracking can be used to solve (7). Let $\boldsymbol{x}(t) = X(t; \tilde{\boldsymbol{x}}, \tilde{t})$ be the characteristic passing through a given point $(\tilde{\boldsymbol{x}}, \tilde{t})$, with $\tilde{t} \in [t^n, t^{n+1}]$, and determined by the following system of equations:

$$\begin{cases} \mathrm{d}\boldsymbol{x}(t)/\mathrm{d}t = V(\boldsymbol{x}(t), t), \\ \boldsymbol{x}(\tilde{t}) = \tilde{\boldsymbol{x}}. \end{cases} \tag{10}$$

This notation can refer to tracking forward or backward in time. In particular, we define

$$\begin{aligned} \boldsymbol{x}^* &= X(t^n; \boldsymbol{x}, t^{n+1}), \\ \bar{\boldsymbol{x}} &= X(t^{n+1}; \boldsymbol{x}, t^n), \end{aligned} \tag{11}$$

so that $(\boldsymbol{x}, t^{n+1})$ backtracks to $(\boldsymbol{x}^*, t^n)$ and $(\boldsymbol{x}, t^n)$ tracks forward to $(\bar{\boldsymbol{x}}, t^{n+1})$.

ELLAM is usually implemented with the backtracking approach (see [4,5,10] among others). With this approach, the right-hand term of Eq. (8) is evaluated with standard value of $\omega_i(\boldsymbol{x}, t^{n+1})$ but backtracked to evaluate $C(\boldsymbol{x}^*, t^n)$, where $\boldsymbol{x}^*$ is the point at the foot corresponding to $\boldsymbol{x}$ at the head of the characteristic. For multidimensional problems, the backtracking algorithm requires significant computational effort, due to the need to define the geometry at t^n which requires mapping of points along the boundary of the element and subsequent interpolation and mapping onto the fixed grid at the previous time level [23]. This approach is considered impractical in two- and three-dimensions [5].

The ELLAM can also be used with the forward-tracking approach. This approach is more practical; it was proposed by Russell and Trujillo [20] and was implemented by Healy and Russell [11]. This would enforce the integration quadrature at t^n with respect to the fixed spatial grid. To evaluate the test function $\omega_i(\boldsymbol{x}, t^n)$, discrete quadrature points chosen in the fixed grid at t^n in a regular fashion are forward tracked to t^{n+1}, where evaluation of $\omega_i(\boldsymbol{x}, t^{n+1})$ is straightforward [23].

2.2. Evaluation of the integrals

The integrals in Eq. (8) are discretized following [5]. The formulations are summarized in Table 1 with the notations:

- E: number of edges e, of length l_e, for which the concentration or the total mass flux is prescribed,
- $\overline{TQ}_e^{n+1}$: prescribed mass flux through edge e,
- $\bar{C}_e^{n+1}$: prescribed concentration at the edge e,
- $\bar{Q}_e^{n+1}$: fluid flux through e,
- N_s: number of 1D spatial integration points, each point $\boldsymbol{x}^p$ having a weight W_e^p,
- N_t: number of integration points in time, each point t^q having a weight W_e^q,
- S^A: area of the element A,
- N_A: number of 2D spatial integration points, each point $\boldsymbol{x}^p$ having a weight W_A^p.

Table 1
Continuous and discrete form of the integrals

Continuous formulation	Discrete formulation
Storage term at time step $n + 1$	
$\int_{\Omega} C^{n+1}\omega_i^{n+1}\,\mathrm{d}\boldsymbol{x}$	$\sum_{A\supset i} C_j^{n+1}\varphi_j^A\omega_i^{n+1} = \sum_{A\supset i}\left(\frac{S^A}{3}C_i^{n+1}\right)$ (12)
Storage term at time step n	
$\int_{\Omega} C^{n}\omega_i^{n}\mathrm{d}\boldsymbol{x}$	$\int_{\Omega} C^{n}\omega_i^{n}d\boldsymbol{x} = \sum_{A}\left[\sum_{p=1}^{N_A}(C^n(\boldsymbol{x}^p,t^n)\omega_i(\boldsymbol{x}^p,t^n)W_A^p)\right]$ (13)
Dispersion term	
$\int_0^{\mathrm{T}}\int_{\Omega}(\boldsymbol{D}\cdot\nabla C)\cdot\nabla\omega_i\mathrm{d}\boldsymbol{x}\mathrm{d}t$	$\Delta t\int_{\Omega}[(\boldsymbol{D}\cdot\nabla C)\cdot\nabla\omega_i]^{n+1}\mathrm{d}\boldsymbol{x}$ (14)
Boundary condition	
$\int_0^{\mathrm{T}}\int_{\partial\Omega}[(\boldsymbol{V}C-\boldsymbol{D}\cdot\nabla C)\omega_i]\boldsymbol{n}_{\partial\Omega}\mathrm{d}l\mathrm{d}t$	
Flux type boundary	
	$\sum_{e=1}^{E}\left\{\overline{TQ}_e^{n+1}\sum_{q=1}^{N_\mathrm{s}}\sum_{p=1}^{N_t}(\omega_i(\boldsymbol{x}^p,t^q)W_e^pW_e^q)\right\}$ (15)
Dirichlet boundary	
	$\sum_{e=1}^{E}\bar{Q}_e^{n+1}\bar{C}_e^{n+1}\left\{\sum_{q=1}^{N_s}\sum_{p=1}^{N_t}\left(\omega_i(\boldsymbol{x}^p,t^q)W_e^pW_e^q\right)\right\}$ $-\sum_{e=1}^{E}\sum_{q=1}^{N_s}\sum_{p=1}^{N_t}\boldsymbol{n}_e\cdot\left\{[(\boldsymbol{D}\cdot\nabla C)\omega_i]_{\tilde{\boldsymbol{x}}^p}^{t^{n+1}}W_e^pW_e^q\right\}$ (16)

To perform the discrete form of the integrals, following assumptions or implementations are used:

- mass lumping for the storage term (the summation in (12) appoints all elements A surrounding the node i);
- the dispersion integral is evaluated using a one-step backward Euler approximation in time;
- at the outflow, since the diffusive part is specified as zero, the outflow boundary does not have any contribution on the distribution of concentration over the domain at the new time level. In practice, all spatial integration points which reach the outflow boundary at time less than Δt when forward tracked will have zero as integration weighting factor;
- to evaluate $\omega_i(\boldsymbol{x}^p, t^q)$, the integration point $\boldsymbol{x}^p$ at time t^q is tracked forward to $\tilde{\boldsymbol{x}}^p$ at t^{n+1} and we use $\omega_i(\boldsymbol{x}^p,t^q)=\omega_i(\tilde{\boldsymbol{x}}^p,t^{n+1})$. Contributions of Eq. (15) and of the first term of Eq. (16) are placed in the right-hand side vector of the final system to solve;

- the second term of the Dirichlet boundary condition (Eq. (16)) is approximated by forward-tracking the boundary values to the new time level [6]. This unknown term is placed in the left-hand side of the final system to solve and introduces an asymmetry into the system matrix.

The method of numerical integration and the number of spatial and time integration points must be chosen carefully. The interpolation of C^n ($\boldsymbol{x}^p$, t^n) can introduce numerical diffusion and/or non-physical oscillations when many time steps are used. This phenomenon as well as the choice of the method of integration and the number of spatial and time integration points will be studied in the next sections.

3. Accurate numerical implementations of the ELLAM on triangles

ELLAM has been applied on rectangular grids only. The implementation on triangles requires new methods to approximate the boundary integral (16) and the mass integral at the old time level (13). These methods are described and compared to more usual methods using a standard test problem.

3.1. Test problem definition

ELLAM implementations will be verified by solving a transport problem in the spatial domain Ω of rectangular shape (0,100) × (0,40) which is partitioned into triangular elements (Fig. 1). Time is divided into discrete intervals $\Delta t = t^{n+1} - t^n$.

Accurate fluxes at each element edge are required to track the particles. Therefore, the flow equation is solved with the mixed finite element approximation as detailed in [29]. With the mixed formulation, the velocity is defined everywhere in the field and is continuous across the inter-element boundaries. Integration over each element gives the corresponding streamlines using the technique developed by Pollok [15].

Fluxes at each edge are obtained by solving the stationary flow problem:

$$\begin{cases} \Delta P = 0 & \text{in } \Omega, \\ P = 1 & \text{in } \partial\Omega_L^1, \\ P = 0 & \text{in } \partial\Omega_R^1, \\ (\nabla P) \cdot \boldsymbol{\eta}_{\partial\Omega} = 0 & \text{in } \partial\Omega^2, \end{cases} \tag{17}$$

where $\partial\Omega_L^1$ and $\partial\Omega_R^1$ are, respectively, the left- and the right-hand perpendicular sides of the domain and $\partial\Omega^2 = \partial\Omega/\partial\Omega^1$.

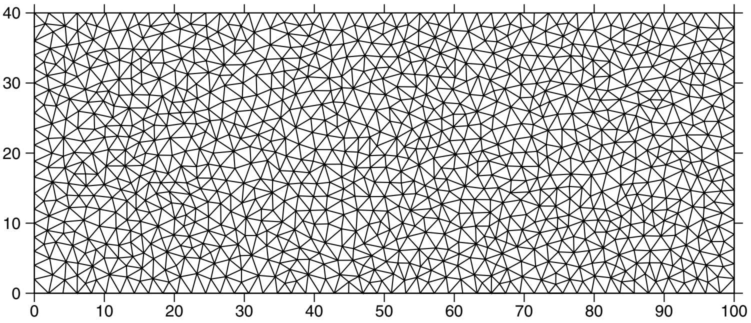


Fig. 1. The domain Ω discretized with non-uniform triangular elements.

The boundary conditions for the transport problem are of Dirichlet type at the inflow (left boundary of Ω), with

$$\begin{cases} C = 0 & \text{for } x = 0 \text{ and } 0 \leqslant y < 12, \\ C = 1 & \text{for } x = 0 \text{ and } 12 \leqslant y \leqslant 28, \\ C = 0 & \text{for } x = 0 \text{ and } 28 < y \leqslant 40. \end{cases} \tag{18}$$

A zero diffusive flux is imposed at the outflow boundary.

This problem is related to mass transport in porous media and the dispersion tensor is defined by

$$\boldsymbol{D}_{ij} = (\alpha_T|\boldsymbol{V}| + D_m)\delta_{ij} + (\alpha_L - \alpha_T)\frac{v_i v_j}{|\boldsymbol{V}|} \quad i,j = 1,2, \tag{19}$$

where $\boldsymbol{V}$ is the fluid velocity vector of components v_i, α_{L} and α_{T} are the longitudinal and transverse dispersivities, δ_{ij} is the Kronecker delta function, and D_{m} is the molecular diffusion coefficient.

The corresponding analytical solution for an infinite domain is [14]

$$C(x,y,t) = \frac{x}{(16\pi\alpha_{\mathrm{L}})^{1/2}} \int_0^{\mathrm{T}} \tau^{-3/2} \left\{ erf\left[\frac{y-12}{(4\alpha_{\mathrm{T}}\tau)^{1/2}}\right] + erf\left[\frac{28-y}{(4\alpha_{\mathrm{T}}\tau)^{1/2}}\right] \right\} \exp\left[-\frac{(x-)^2}{4\alpha_{\mathrm{L}}\tau}\right] \mathrm{d}\tau \tag{20}$$

with $erf(x) = \frac{2}{\sqrt{\pi}}\int_0^x \exp(-\tau^2)\mathrm{d}\tau$.

Solution of (17) leads to fluid velocity components equal to $v_x = 1.0$ and $v_y = 0.0$ over the whole domain.

3.2. The accurate numerical integration for the boundary integral (16)

The test problem is simulated with a single time step of 60 days. Since this problem corresponds to a constant injection into a domain originally free of contaminant, the spatial integration at the old time level (13) is not needed. Integrals (12) and (14) are standard as with finite element method. Boundary integration points in (16) are only used over Dirichlet boundary edges where prescribed concentration is nonzero.

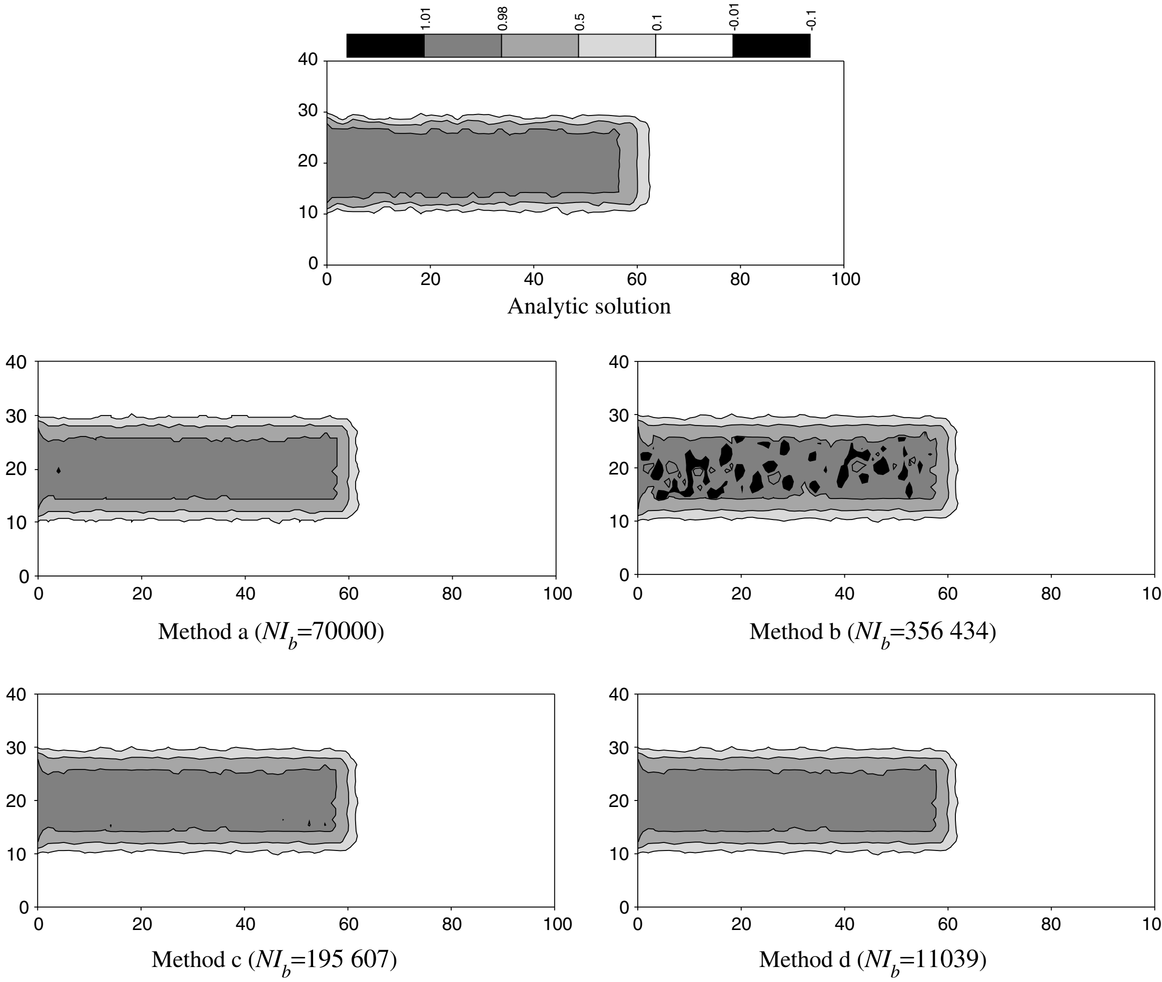
To approximate *the boundary integral* (16), the following methods are used:

- *Method a*: the trapezoid rule is used on each edge e for which the prescribed concentration is nonzero. This is a quite standard scheme for the ELLAM. Intermediate integration points are included for both space and time. We use a uniform space-time density of (100 × 100) intermediate integration points for each edge e.
- *Method b*: Strategic time integration points (STIP) obtained from strategic spatial integration points (SSIP) when tracked to the boundary as defined by Healy and Russell [12], are used in this method. The location and time of their arrival at the boundary become intermediate integration points for space and time, respectively. The trapezoid rule is used to obtain the weight of the integration points. For this method, we use 4 SSIPs per triangular element located respectively at the three nodes and at the center of each triangle.
- *Method c*: This method corresponds to a combination of the previous methods. An uniform space-time density of (100 × 100) intermediate integration points for each edge e is used and the corresponding STIPs obtained from one SSIP per element (located at the center of each triangle) are added. Again, the trapezoid rule is used to obtain the weight of the integration points.
- *Method d*: for this method, we use N_{s} spatial integration points corresponding to Gauss points on each edge e. W_e^q correspond to standard Gauss weights. As in Binning and Celia [6], the number of time integration points N_t, is computed from the number of spatial integration points and the Courant number $CFL = \frac{1}{2S^A}\sum_{i=1}^{3}\left|\bar{Q}_{i,A}^{n+1}\right|\Delta t$ ($\bar{Q}_{i,A}^{n+1}$ is the fluid flux across the edge i of the element A, S^A the area of the element A). Since we use non-uniform meshes, we define $CFL^{\max}$ the greatest CFL over all elements (not only for boundary elements as in Binning and Celia [5]) for which a particle located at the center crosses the edge e in a time less than Δt when tracked backward. N_{t} is defined by $N_{\mathrm{t}} = 2 \times CFL^{\max} \times N_{\mathrm{s}}$. Simpson's rule is used for the numerical integration in time.

If $\bar{C}_e^{n+1} = 0$, the forward tracking of the corresponding integration points is not necessary.

Results obtained with the four suggested integrations (Methods a–d) and the total number of integration points NI_b are given in Fig. 2. The dispersion parameters are $\alpha_L = 10^{-3}$, $\alpha_T = 10^{-4}$and $D_m = 10^{-5}$. The corresponding analytical solution is also given in Fig. 2.

From this figure, we can see that the use of SSIP and STIP (Method b) provides less accurate results than an uniform space-time distribution of the integration points. Indeed, using SSIP and STIP leads to better results for rectangular meshes [12]. More accurate results are obtained with significantly less integration points with Method d. Therefore, in the rest of this work, the boundary integral will always be computed with the numerical integration Method d. The total number of integration points is $NI_b = 2 \times CFL^{\max} \times N_s^2$, where N_s is the number of spatial gauss integration points for each edge with nonzero prescribed concentration.



Analytic solution

Method a (NI_b=70000)

Method b (NI_b=356 434)

Method c (NI_b=195 607)

Method d (NI_b=11039)

Fig. 2. Distribution of the concentration after a simulation time of 60 days with different numerical boundary integration methods.

ELLAM is suspected to provide solutions with much more oscillations for unstructured meshes than for uniform meshes. The effect of grid orientation on ELLAM results was considered in [18] for rectangular grids. In [18], flow orientated with 45° to the grid leads to results with overshoot and undershoot. This phenomenon can be reduced by using finer grid [18] or increasing the number of boundary integration points.

The mesh orientation effect is studied here with three different triangulations: a uniform triangulation (a) (Fig. 3), a structured triangulation (b) (Fig. 1) and an unstructured triangulation (c) (Fig. 3). For this last triangulation,the mesh refinement is located close to transverse concentration fronts in order to increase the sensitivity of the spatial resolution.

The test problem is simulated for a single time step of 60 days with the same dispersion parameters as previously. For the three meshes, Method d is used with three different numbers of spatial boundary integration points:

- *Method_d1*: corresponds to Method d with $N_s = 3$ and $N_t = 2 \times CFL^{\max} \times N_s$.
- *Method_d2*: corresponds to Method d with $N_s = 7$ and $N_t = 2 \times CFL^{\max} \times N_s$.
- *Method_d3*: corresponds to Method d with $N_s = 15$ and $N_t = 2 \times CFL^{\max} \times N_s$.

Results of the simulations and the total number of integration points NI_b are given in Fig. 4 for the three meshes.

These experiments show that:

- For a given mesh, the accuracy of the results depends strongly on the number of integration points used to evaluate the boundary integral.

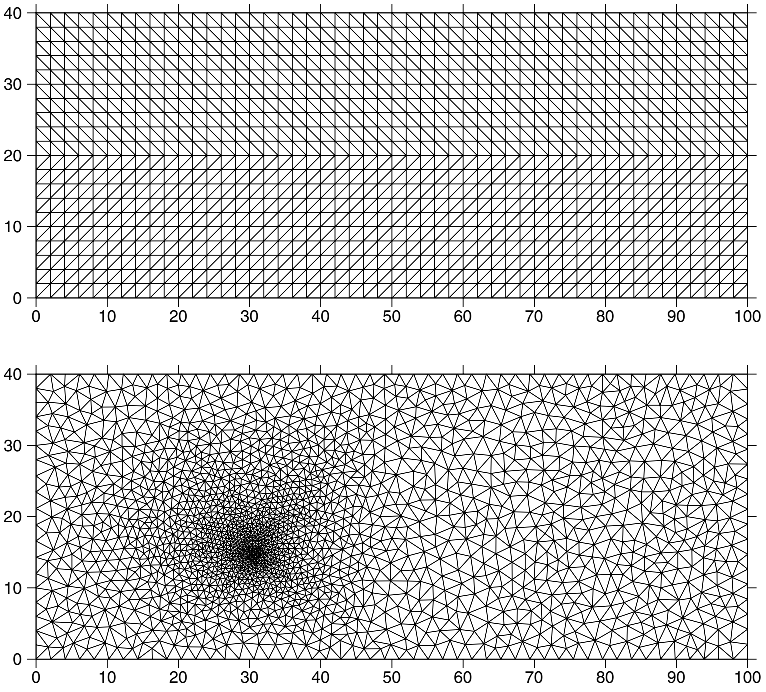


Fig. 3. The domain Ω discretized with uniform and unstructured meshes.

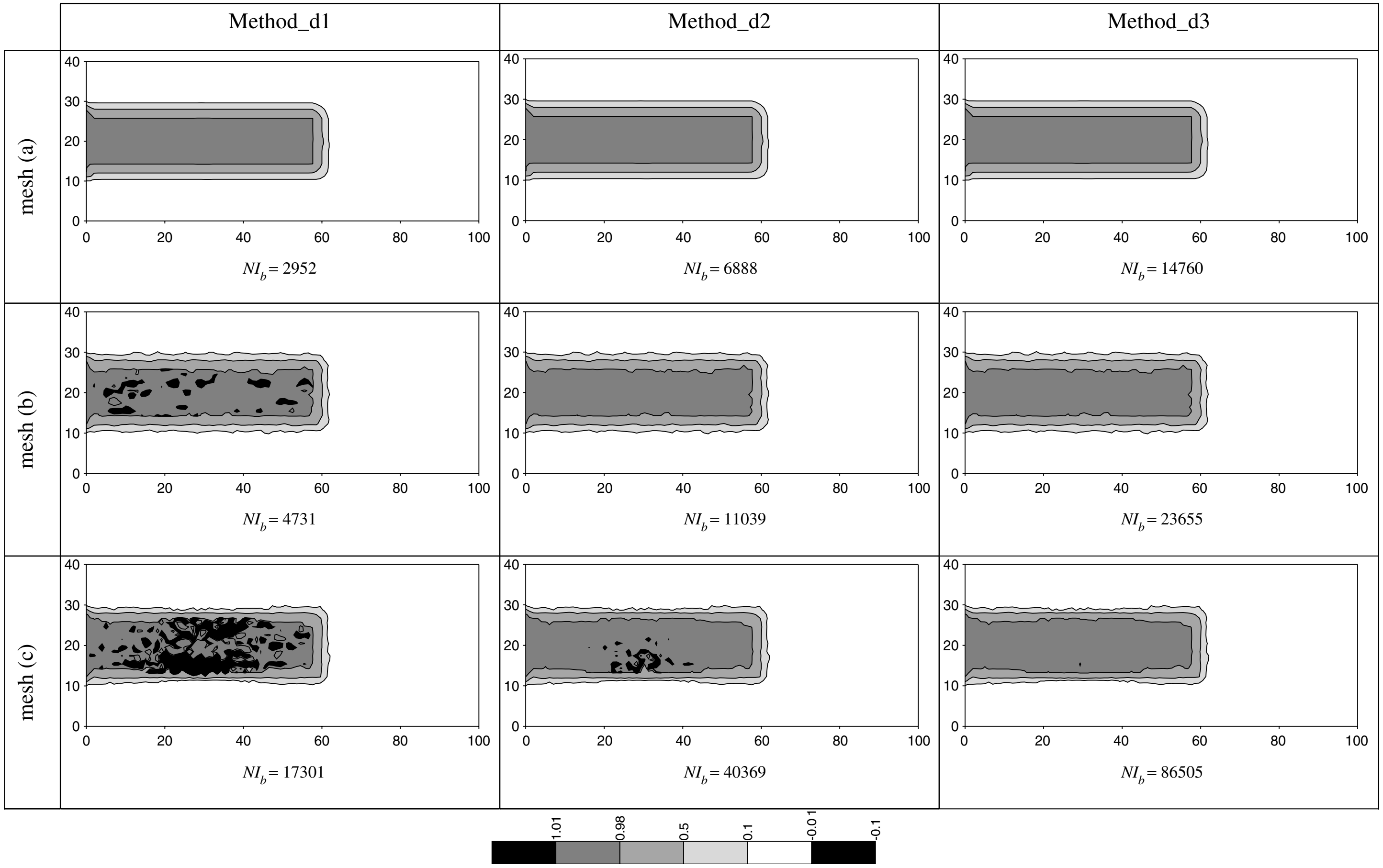


Fig. 4. Concentration distribution with different numbers of boundary integration points for the three meshes.

- To obtain accurate results, the number of integration points used to evaluate integrals is depending on the mesh. For the uniform mesh (a), Method_d1 with $N_s = 3$ per Dirichlet boundary edge is sufficient to obtain a good solution. For mesh (b), more integration points are required. A good solution is obtained with $N_s = 7$. The last unstructured mesh (c) needs however much more integration points ($N_s = 15$) to avoid oscillations. Increasing the number N_s and so the number of particles at the boundary edge allows to achieve sufficient mass for each element at the new time level which reduces nonphysical oscillations in the solution. This will be required particularly when working with unstructured meshes and/or for more complicated flow problems (with divergence of streamlines in the solution, for example).
- Since the computational effort of the method is depending on the total number of particles to be tracked, the solution for the unstructured mesh ($NI_b = 86505$) is much more CPU consuming than for the uniform mesh ($NI_b = 2952$).
- It is important to note that the number of boundary integration points is not the same for all Dirichlet boundary edges. Indeed, for each edge, this number is driven by the smallest element size not among the whole grid but among all elements which receive mass from that boundary edge. In this way, the scheme remains efficient when employed for unstructured grids.

3.3. *Spatial integral at the old time level (13)*

To study the effect of the *spatial integral at the old time level* (13), we test the following methods:

- SPINT_6GP: Over each triangular element, we use 6 Gauss integration points with the corresponding coordinates and weights. Concentration $C^n(\boldsymbol{x}^p, t^n)$ at these points is obtained by linear interpolation of concentrations at grid nodes. Each point is then forward tracked to the new time level to evaluate $\omega_i(\boldsymbol{x}^p, t^n) = \omega_i(\tilde{\boldsymbol{x}}^p, t^{n+1})$.
- SPINT_12GP: similar to Method 1, but with 12 Gauss integration points.
- SPINT_24GP: in this case, each triangular element is divided into 4 equal sub-elements (by joining edge centers). Over each sub-element, 6 Gauss integration points are used, i.e., the number of integration points over the whole triangle is equal to 24.

To evaluate the accuracy of the suggested 3 methods used to compute integral (13), the simulation time T is divided into two equal intervals of 30 days. Results of the transport problem for mesh (b) with $\alpha_L = 10^{-3}$, $\alpha_T = 10^{-4}$ and $D_m = 10^{-5}$ are given in Fig. 5. In this case, the boundary integral (16) is computed with sufficient numerical integration points as explained previously. Oscillations occur after the second time step only and are due to the computation of the spatial integral (13). After the first time step, the solute would reach $x = 30$ if transported by advection only. After the second time step, oscillations appear in the concentration distribution at $x > 30$ when not enough integration points are used. Method SPINT_24GP, which uses 24 Gauss points per element, leads to better results than the other methods. Numerical experiments show also that the number of spatial numerical integration points required to obtain the accurate solution depends on the mesh. Indeed, for the uniform mesh (a) only 6 spatial integration points per triangle are required. However for the unstructured mesh (c), we need at least 24 integration points to obtain the good solution.

4. Numerical diffusion with the ELLAM

ELLAM may suffer from numerical dispersion [16] due to interpolation and numerical approximations. However, this numerical dispersion can be strongly reduced by an accurate evaluation of the dispersion

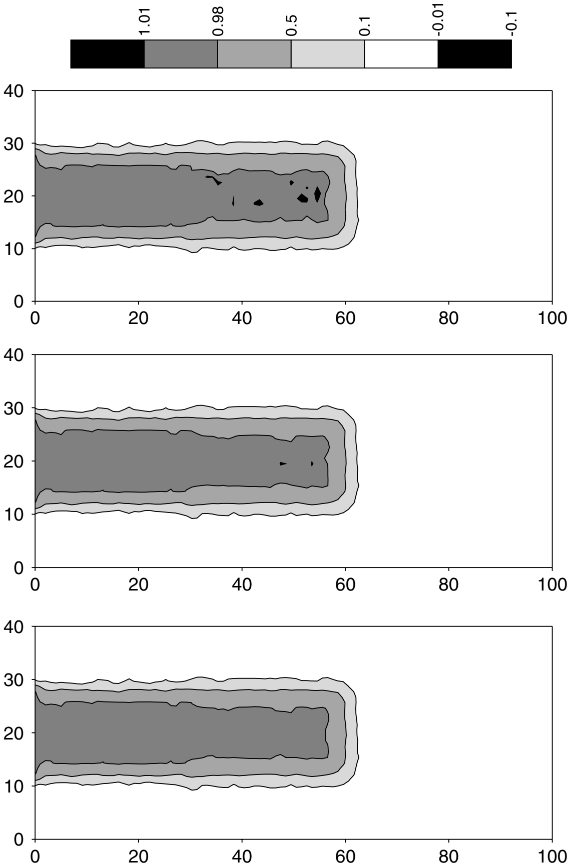


Fig. 5. Concentration distribution with different number of spatial integration points.

integral and a good approximation of the concentration at the foot of the characteristic, as it is shown in this section.

4.1. Integration of the dispersion term (14)

As stated in some works in the literature ([1,3,23] among others), the time step size Δt used in the dispersion integral (14) has to be reduced for elements near the inflow boundary.The discretization of the *dispersion term* (14) has been modified, as following

$$\int_0^{\mathrm{T}} \int_\Omega (\boldsymbol{D} \cdot \nabla C) \cdot \nabla \omega_i \,\mathrm{d}x\,\mathrm{d}y\,\mathrm{d}t \approx \sum_{A \supset i} \min(\Delta t, \Delta t^A) \int_A [(\boldsymbol{D} \cdot \nabla C) \cdot \nabla \omega_i]^{n+1} \,\mathrm{d}x\,\mathrm{d}t, \tag{21}$$

where Δt^A is the arrival time at the inflow boundary, when backtracking the particle which starts at $t = 0$ at the center of the element A. The equivalent time step in Eq. (21) will be smaller for elements which backtrack to the inflow boundary during the time period Δt.

Because the ELLAM allows using large time steps, significant differences can be found between formulations (14) and (21). To show this effect, the previous test problem is simulated in the case of dispersive dominant transport ($\alpha_L = 2$, $\alpha_T = 0.5$ and $D_m = 0$). Results with both formulations as well the analytical solution are given in Fig. 6 in the case of a single time step simulation of 60 days. The standard approach, given by Eq. (14), leads to excessive dispersion (Fig. 6(a)). Using the scheme described by Eq. (21) leads to significantly more accurate results (Fig. 6(b)). The differences are due to the weight of the dispersion term in Eq. (8). This weight is excessive when formula (14) is used compared to formula (21).

4.2. Numerical diffusion due to interpolation

To study the effect of the number of time steps on the behavior of the solution, the previous simulation is done with a final simulation time of $T = 60$ days divided into:

- *Case 1:* a single time step of 60 days.
- *Case 2:* 6 time steps of 10 days.
- *Case 3:* 20 time steps of 3 days.

Simulation results (Fig. 7) show that numerical diffusion becomes important when several time steps are used. With Eulerian–Lagrangian methods, we need the concentration at the foot of the characteristic for

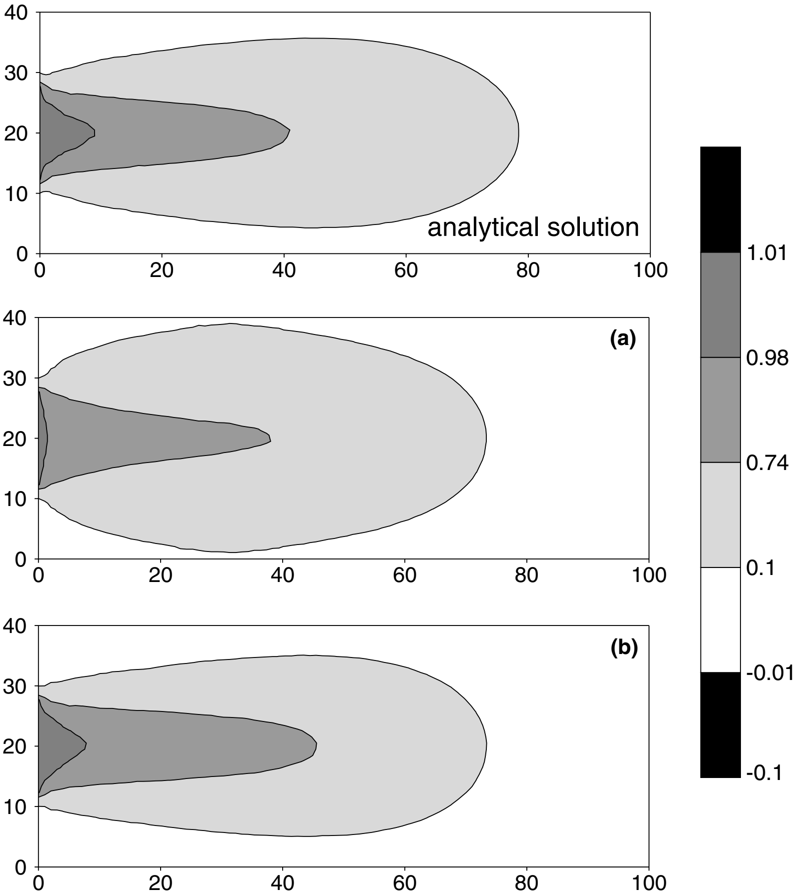


Fig. 6. A single time step ELLAM results with different dispersion formulations.

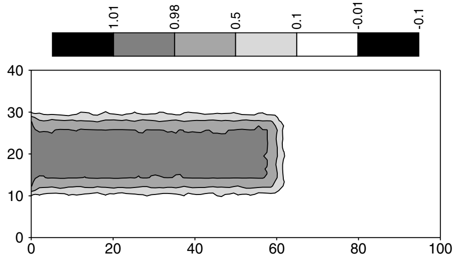
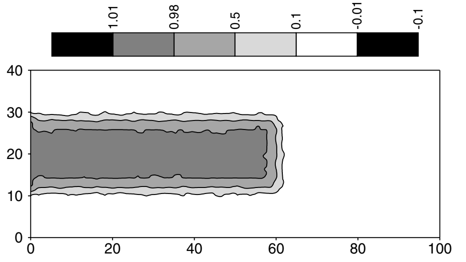
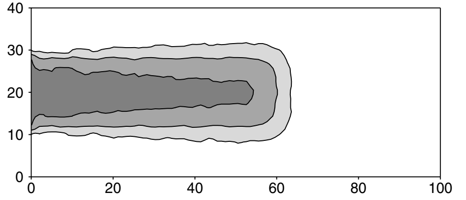
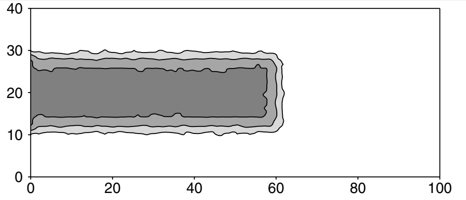
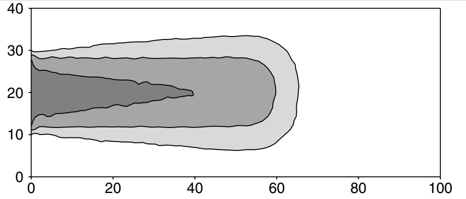
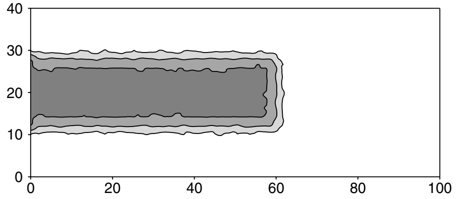
	Number of time steps	Time step	Standard approach	New method
Method 1	1	60 days		
Method 2	6	10 days		
Method 3	20	3 days		

Fig. 7. Numerical dispersion with ELLAM when using several time steps.

each time step. Because the location of the foot of the characteristic does not coincide with a grid point, interpolation is necessary. When many interpolations are used (many time steps), numerical diffusion becomes significant. With higher order interpolation, negative weights will be necessary to avoid numerical dispersion, creating the potential of spurious oscillations [16].

To reduce the problem of numerical diffusion with ELLAM, we suggest the following new scheme:

- At the inflow boundary, a particle of weight $\bar{Q}_e^{n+1} W_e^p W_e^q$ and concentration $\bar{C}_e^{n+1}$ is associate to each integration point $(\boldsymbol{x}^p, t^q)$ of the edge e.
- Each particle keeps the same weight for all time steps along the simulation.
- To avoid interpolation, we use the same particles for all time steps. For the first time step, each boundary particle of concentration $\bar{C}_e^{n+1}$ and weight $\bar{Q}_e^{n+1} W_e^p W_e^q$ is forward tracked into the domain. At the second time step, this particle has a weight $W_A^p = \bar{Q}_e^{n+1} W_e^p W_e^q$ and is forward tracked again until the end of the second time step and so on for all time steps. We use the same characteristics along the simulation. For each time step, the foot of each characteristic corresponds to the head of a characteristic from the previous time step.
- For each time step, new particles are added at the inflow boundary.
- When dispersion is zero, the value of the concentration $\bar{C}_e^{n+1}$ for the particles is constant during the simulation and for all time steps. When dispersion is taken into account, this value at the foot of each characteristic has to be adapted. This is done as follows:
  - At the end of each time step, we compute the value C_i^{adv} at each node i due to the advection process only. This is done using Eq. (8) with zero dispersion. Because we use mass lumping, this procedure is not CPU time consuming. We have to perform a local (at the element level) computation and no linear system has to be solved.
  - On the other hand, at the end of each time step, we obtain $C_i^{n+1} = C_i^{\text{adv+disp}}$ at each node of the mesh. Theses values of the concentration are due to both advection and dispersion phenomena. Therefore, for each node of the mesh, we compute C_i^{disp} which is only due to the dispersion phenomenon using $C_i^{\text{disp}} = C_i^{\text{adv+disp}} - C_i^{\text{adv}}$.
  - At the foot of each characteristic, the value of the concentration $C_l^{\text{f}}\big|^{n+1}$ of the particle l is adapted from $C_l^{\text{h}}\big|^{n}$ at the head of the characteristic of the previous time step using: $C_l^{\text{f}}\big|^{n+1} = C_l^{\text{h}}\big|^{n} + C_l^{\text{h}}\big|^{n,\text{disp}}$, where $C_l^{\text{h}}\big|^{n,\text{disp}}$ is obtained by interpolating C_i^{disp} of the three nodes of the arrival element of the particle.

In this way, only the dispersive part of the transport is interpolated at each time step. Since dispersion process is much smoother than the advection-dispersion process, numerical diffusion due to the interpolation of the concentration is strongly reduced.

However, the two following problems remain:

- Since all particles arrive from the inflow boundary, the distribution of the concentration at $t = 0$ is not taken into account.
- At the end of each time step, some elements receive mass by dispersion only and no particle arrives at these elements. For the following time steps, the mass at these elements has to be transported by advection. In the current scheme, no particle will start from these elements.

To avoid the previous problems, we proceed as following:

- At $t = 0$, we use spatial integration points with fixed weights for all elements in the domain. In this work, 24 integration points per triangle are used. Each particle keeps the same weight along the simulation and the value of the concentration is adapted at the end of each time step by only interpolating the dispersion part. These particles will be progressively replaced by particles arriving from the inflow boundary.

- We use integration points for all inflow boundaries and not only for inflow boundary where the prescribed value of the concentration is not zero. In this way, particles arriving with no mass at the end of a time step can start with a non zero value of concentration for the next time step because of dispersion.

Simulation results (Fig. 7) show that, contrarily to the standard approach, the new scheme does not suffer from excessive artificial dispersion even when many time steps are used. Simulations are done with the transport parameters given in Section 3.2. The corresponding analytical solution is shown in Fig. 2. Similar results are obtained with non uniform time stepping.

As detailed in Eq. (12), the new scheme uses the standard mass lumping procedure to evaluate the mass at the new time level. Mass-lumping is often used to avoid oscillations with numerical methods and is known to add excessive numerical diffusion with Eulerian–Lagrangian methods [19]. To avoid this problem, one dimensional ELLAM scheme with a selective lumping has been developed in [19]. This procedure is expected to be much more complicated for bi-dimensional problems with unstructured meshes.

With the suggested implementation, only variations due physical dispersion are added to the concentration at the foot of each characteristic. Results of Fig. 7 show that, although using mass lumping procedure, the new scheme does not lead to excessive numerical diffusion when many time steps are used.

5. Efficiency of ELLAM compared to high order Eulerian method

The developed ELLAM requires an important number of integration points, which increases its CPU cost, especially for unstructured meshes. The main question is whether ELLAM remains competitive when compared to other numerical methods.

To this aim, ELLAM will be compared to a numerical code based on explicit Discontinuous Finite Element Method (DFEM). An explicit scheme is used for the convective part of the solute transport equation, an implicit scheme for its dispersive part [21]. The dispersive part of the transport equation is solved with mixed hybrid finite element method. Due to the explicit scheme, the Courant criterion has to be fulfilled which can lead to very small time steps for unstructured meshes. Compared to other more standard methods like finite volumes, this approach was found more accurate for high Peclet numbers [21].

The following cases are simulated:

- $Pe \approx 100$ ($\alpha_L = 0.02$, $\alpha_T = 0.002$, $D_m = 0$),
- $Pe \approx 10$ ($\alpha_L = 0.2$, $\alpha_T = 0.02$, $D_m = 0$),
- $Pe \approx 1$ ($\alpha_L = 2$, $\alpha_T = 0.2$, $D_m = 0$),

where Pe is the grid Peclet number with grid characteristic length of about 2. Each of the three cases is simulated with a single time step, 6 time steps and 15 time steps on the three meshes: the uniform mesh (a) (Fig. 3), the structured mesh (b) (Fig. 1) and the unstructured mesh (c) (Fig. 3).

Simulations are done with both codes (DFEM & ELLAM). For the ELLAM, we use Method_d1 for mesh (a), Method_d2 for mesh (b) and Method_d3 for the unstructured mesh (c).

The root-mean-squared (RMS) error and the CPU cost of each method are compared (Tables 2–4).

The RMS error is defined by

$$\text{RMSError} = \frac{1}{N}\sum_{i=1}^{N}\sqrt{\left[\left(C_i^{\text{numerical}} - C_i^{\text{analytical}}\right)\right]^2}, \tag{22}$$

where N is the total number of grid nodes.

Table 2
ELLAM versus DFEM for $Pe = 100$ ($\alpha_L = 0.02$, $\alpha_T = 0.002$, $D_m = 0$)

		Mesh (a)		Mesh (b)		Mesh (c)	
		RMSError	CPU	RMSError	CPU	RMSError	CPU
DFEM		1×10^{-2}	1.63	$2.8\ 10^{-2}$	6.12	3.3×10^{-2}	91.4
ELLAM	$1\Delta t$	1.7×10^{-3}	1.1	$1.2\ 10^{-2}$	1.6	9×10^{-3}	6.8
	$6\Delta t$	2.2×10^{-3}	1.56	1.2×10^{-2}	2.23	8.9×10^{-3}	8.07
	$15\Delta t$	5.7×10^{-3}	2.6	8×10^{-3}	3.6	8.3×10^{-3}	11.8

Table 3
ELLAM versus DFEM for $Pe = 10$ ($\alpha_L = 0.2$, $\alpha_T = 0.02$, $D_m = 0$)

		Mesh (a)		Mesh (b)		Mesh (c)	
		RMSError	CPU	RMSError	CPU	RMSError	CPU
DFEM		6.6×10^{-3}	1.66	1.47×10^{-2}	6.1	1.7×10^{-2}	60.7
ELLAM	$1\Delta t$	5.3×10^{-3}	1.1	8.7×10^{-3}	1.65	7.2×10^{-3}	6.89
	$6\Delta t$	4×10^{-3}	1.59	8.2×10^{-3}	2.2	7.8×10^{-3}	8.2
	$15\Delta t$	5.7×10^{-3}	2.6	8.2×10^{-3}	3.6	8.3×10^{-3}	11.8

Table 4
ELLAM versus DFEM for $Pe = 1$ ($\alpha_L = 2$, $\alpha_T = 0.2$, $D_m = 0$)

		Mesh (a)		Mesh (b)		Mesh (c)	
		RMSError	CPU	RMSError	CPU	RMSError	CPU
DFEM		2.8×10^{-3}	1.43	2.7×10^{-3}	5.12	3.7×10^{-3}	72.4
ELLAM	$1\Delta t$	1.7×10^{-2}	1.1	1.6×10^{-2}	1.68	2×10^{-2}	6.9
	$6\Delta t$	1×10^{-2}	1.65	1.1×10^{-2}	2.3	1.1×10^{-2}	8.4
	$15\Delta t$	1.3×10^{-2}	2.6	1.1×10^{-2}	3.6	1×10^{-2}	12.2

From these tables, we can see that

- Whatever the number of time step, the RMS Error is similar for both methods.
- The CPU time of ELLAM is not strictly proportionate to the number of time steps. Indeed, ELLAM with 15 time steps needs about 50% more CPU time than ELLAM with a single time step.
- Table 2 shows that for convective dominated transport, ELLAM is more accurate than DFEM. In one dimensional problems, it was shown in [27] that numerical diffusion generated by DFEM is more important than with ELLAM.
- For the uniform mesh (a), the single time step ELLAM needs 50% less CPU time than DFEM.
- For the uniform mesh (a), when working with many time steps, DFEM becomes less CPU consuming than ELLAM.
- For the structured mesh (b), the single time step ELLAM needs 80% less CPU time than DFEM.
- For the unstructured mesh (c), we have a strong reduction of the computation time with ELLAM since the single time step ELLAM needs between 6 and 9 times less CPU time than DFEM.

For all grid Peclet numbers and for all meshes, the single time step ELLAM is found to be much more efficient than DFEM.

6. Conclusion

In the literature, ELLAM has been only applied on rectangular grids. We developed here the ELLAM for unstructured triangular meshes.

Numerical experiments show that the usual numerical integrations which performed well for rectangular grids are not accurate enough for such meshes. The adequate numerical integration method for the boundary integral uses the Gauss integration method in space and Simpson's rule for the numerical integration in time.

Unstructured meshes require increasing the number of particles to achieve sufficient mass for each element at the new time level in order to reduce non-physical oscillations. In this work, the number of boundary integration points for each edge is driven by the smallest element size among all elements which receive mass from that boundary edge.

Numerical diffusion due to interpolation when using the ELLAM with many time steps is also addressed. A new algorithm is developed in order to avoid excessive numerical diffusion. The basic idea of this approach is to keep the same characteristics for all time steps. To obtain the concentration at the foot of the characteristic, we keep the advective part of the concentration (since it is constant along the characteristic) and we interpolate only the concentration variations due to the dispersion process.

Numerical experiments for grid Peclet numbers ranking from 1 to 100 show the efficiency of the developed scheme. Although ELLAM requires a lot of integration points for unstructured meshes, it remains a competitive method when using a single or many time steps, compared to a high order Eulerian scheme based on Discontinuous and Mixed finite element methods.

The suggested implementations make the ELLAM very attractive, especially because large time steps can be used without fulfilling the Courant criterion. These implementations are not restricted to 2D and can be applied to 3D.

References

[1] M. Al-Lawatia, R.C. Sharpley, H. Wang, Second-order characteristic methods for advection-diffusion equations and comparison to other schemes, Adv. Water Res. 22 (1999) 741–768.

[2] T. Arbogast, M.F. Wheeler, A characteristics-mixed finite element method for advection-dominated transport problems, SIAM J. Numer. Anal. 32 (1995) 404–424.

[3] L.S.J. Bell, P.J. Binning, A split operator approach to reactive transport with the forward particle tracking Eulerian–Lagrangian Localized Adjoint Method, Adv. Water Res. 27 (4) (2004) 323–334.

[4] P. Binning, Modeling Unsaturated Zone Flow and Contaminant in the Air and Water Phases, Ph.D. Thesis, Department of Civil Engineering and Operational Research, Princeton University, Princeton, NJ, 1994.

[5] P. Binning, M.A. Celia, A finite volume Eulerian–Lagrangian localized adjoint method for solution of the contaminant transport equations in two-dimensional multi-phase flow systems, Water Resour. Res. 32 (1996) 103–114.

[6] P. Binning, M.A. Celia, A forward particle tracking Eulerian–Lagrangian localized adjoint method for solution of the contaminant transport equation in three dimensions, Adv. Water Res. 25 (2002) 147–157.

[7] M.A. Celia, I.A. Ferrand, A comparison of ELLAM formulations for simulation of reactive transport in groundwater, in: Wang (Ed.), Advances in Hydro-science and Engineering, 1(B), University of Mississippi Press, 1993, pp. 1829–1836.

[8] M.A. Celia, T.F. Russell, I. Herrera, R.E. Ewing, An Eulerian–Lagrangian localized adjoint method for the advection-diffusion equation, Adv. Water Res. 13 (1990) 187–206.

[9] M.A. Celia, S. Zisman, An Eulerian–Lagrangian localized adjoint method for reactive transport in groundwater, in: G. Gambolati et al. (Eds.), Conference on Computational Methods in Water Resources VIII, vol. I., Springer-Verlag, Berlin, 1990, pp. 383–392.

[10] R.E. Ewing, H. Wang, Eulerian–Lagrangian localized adjoint methods for variable-coefficient advective-diffusive-reactive equations in groundwater contaminant transport, in: Gomez, Hennart (Eds.), Advances in Optimization and Numerical Analysis, Mathematics and Its Applications, Kluwer Academic Publishers, Dordrecht, Netherlands, 1994, pp. 185–205.

[11] R.W. Healy, T.F. Russell, A finite-volume Eulerian–Lagrangian localized adjoint method for solution of the advection-dispersion equation, Water Resour. Res. 29 (1993) 2399–2413.

[12] R.W. Healy, T.F. Russell, Solution of the advection-dispersion equation in two dimensions by a finite-volume Eulerian–Lagrangian localized adjoint method, Adv. Water Res. 21 (1998) 11–26.

[13] R.W. Healy, T.F. Russell, Treatment of internal sources in the finite volume ELLAM, in: L.R. Bentley, J.F. Sykes, C.A. Brebbia, W.G. Gray, G.F. Pinder (Eds.), XIII International Conference on Computational Methods in Water Resources, A.A. Balkema, Calgary, 2000.

[14] F.J. Leij, J.H. Dane, Analytical solution of the one-dimensional advection equation and two or three-dimensional dispersion equation, Water Resour. Res. 26 (1990) 1475–1482.

[15] D.W. Pollock, Semianlytical computation of path lines for finite-diffrence models, Ground Water 26 (1988) 743–750.

[16] T.F. Russell, Numerical dispersion in Eulerian–Lagrangian methods, in: S.M. Hassanizadeh, et al. (Eds.), Computational Methods in Water Resources, vol. 2, Elsevier, Amsterdam, 2002, pp. 963–970.

[17] T.F. Russell, M.A. Celia, An overview of research on Eulerian–Lagrangian Localized adjoint methods (ELLAM), Adv. Water Res. 25 (2002) 1215–1231.

[18] T.F. Russell, C.I. Heberton, L.F. Konikow, G.Z. Hornberger, A finite-volume ELLAM for three-dimensional solute-transport modeling, Groud Water 41 (2) (2003) 258–272.

[19] T.F. Russell, P. Binning, Oh No, not the Wiggles Again! A Revisit of an Old Problem and a New Approach, in: C.T. Miller et al. (Eds.), Computational Methods in Water Resources, vol. 1, Elsevier, Amsterdam, 2004, pp. 483–494.

[20] T.F. Russell, R.V. Trujillo, Eulerian–Lagrangian localized adjoint methods with variable coefficients in multiple dimensions, Computational methods in surface hydrology, in: Proceedings of the Eighth International Conference on Computational Methods in Water Resources. Italy, Venice, 1990, pp. 357–363.

[21] P. Siegel, R. Mosé, Ph Ackerer, J. Jaffré, Solution of the advection dispersion equation using a combination of discontinuous and mixed finite elements, Int. J. Numer. Meth. Fluids 24 (1997) 595–613.

[22] J.E. Vag, H. Wang, H.K. Dahle, Eulerian–Lagrangian localized adjoint methods for systems of nonlinear advective-diffusive-reactive equations, Adv. Water Res. 19 (1996) 297–315.

[23] H. Wang, H.K. Dahle, R.E. Ewing, M.S. Espedal, R.C. Sharpley, S. Man, An ELLAM scheme for advection-diffusion equations in two dimensions, SIAM J. Sci. Comput. 20 (6) (1999) 2160–2194.

[24] H. Wang, R.E. Ewing, M.A. Celia, Eulerian–Lagrangian localized adjoint method for reactive transport with biodegradation, Numer. Meth. Partial Differ. Eq. 11 (1995) 229–254.

[25] H. Wang, R.E. Ewing, G. Qin, S.L. Lyons, M. Al-Lawatia, S. Man, A family of Eulerian–Lagrangian localized adjoint methods for multi-dimensional advection-reaction equations, J. Comput. Phys. 152 (1999) 120–163.

[26] H. Wang, D. Liang, R.E. Ewing, S.K. Lyons, G. Qin, An accurate simulator of compressible flow in porous media with wells, in: L.R. Bentley, J.F. Sykes, C.A. Brebbia, W.G. Gray, G.F. Pinder (Eds.), XIII International Conference on Computational Methods in Water Resources, A.A. Balkema, Calgary, 2000.

[27] A. Younes, An accurate moving grid Eulerian–Lagrangian localized adjoint method for solving the one-dimensional variable-coefficient ADE, Int. J. Numer. Meth. Fluids 45 (2004) 157–178.

[28] A. Younes, Modélisation du transport réactif en milieux poreux non saturés avec une méthode ELLAM en maillage variable, C.R. Geoscience 336 (2004) 547–552.

[29] A. Younes, P.h. Ackerer, R. Mosé, G. Chavent, A new formulation of the mixed finite element method for solving elliptic and parabolic PDE with triangular elements, J. Comp. Phys. 149 (1999) 148–167.