SIAM J. SCI. COMPUT.
Vol. 20, No. 6, pp. 2160–2194

AN ELLAM SCHEME FOR ADVECTION-DIFFUSION EQUATIONS IN TWO DIMENSIONS*

HONG WANG†, HELGE K. DAHLE‡, RICHARD E. EWING§, MAGNE S. ESPEDAL‡, ROBERT C. SHARPLEY†, AND SHUSHUANG MAN†

Abstract. We develop an Eulerian–Lagrangian localized adjoint method (ELLAM) to solve two-dimensional advection-diffusion equations with all combinations of inflow and outflow Dirichlet, Neumann, and flux boundary conditions. The ELLAM formalism provides a systematic framework for implementation of general boundary conditions, leading to mass-conservative numerical schemes. The computational advantages of the ELLAM approximation have been demonstrated for a number of one-dimensional transport systems; practical implementations of ELLAM schemes in multiple spatial dimensions that require careful algorithm development are discussed in detail in this paper. Extensive numerical results are presented to compare the ELLAM scheme with many widely used numerical methods and to demonstrate the strength of the ELLAM scheme.

Key words. characteristic methods, comparison of numerical methods, Eulerian–Lagrangian methods, numerical solutions of advection-diffusion equations

AMS subject classifications. 65M25, 65M60, 76M10, 76S05

PII. S1064827596309396

1. Introduction.

Many difficult problems arise in the numerical simulation of advection-diffusion equations, which describe the transport of solutes in groundwater and surface water, the displacement of oil by fluid injection in oil recovery, the movement of aerosols and trace gases in the atmosphere, and miscible fluid flow processes in many other applications. In industrial applications, these equations are commonly discretized via finite difference methods (FDM) or finite element methods (FEM) in large-scale simulators. Because of the enormous size of many field-scale applications, large grid-spacings must be used in the simulations. When physical diffusion dominates the transport process, these methods perform fairly well. However, when advection dominates the transport process, these methods suffer from serious numerical difficulties. Centered FDM (in space or time) and corresponding FEM often yield numerical solutions with excessive oscillations. The classical space-upwinded (or backward-in-time) schemes can greatly suppress the oscillations, but they tend to generate numerical solutions with severe damping or a combination of both. Recent developments in effectively solving advection-diffusion equations have generally been along one of two approaches: Eulerian or characteristic methods. Eulerian methods use a fixed spatial grid such as the optimal test function methods of Christie et al. [16], Barrett and Morton [5], Celia et al. [13], and Bank et al. [3]. These methods attempt to minimize the error in approximating spatial derivatives and yield an upstream bias

*Received by the editors September 13, 1996; accepted for publication November 24, 1997; published electronically July 16, 1999. This research was supported in part by DE-FG03-95ER25263 and DE-FG05-95ER25266, by ONR N00014-94-1-1163, by funding from the Institute for Scientific Computation at Texas A&M University, and by funding from VISTA, a research cooperation between Statoil and the Norwegian Academy of Science and Letters.

http://www.siam.org/journals/sisc/20-6/30939.html

†Department of Mathematics, University of South Carolina, Columbia, SC 29208 (hwang@math.sc.edu, sharpley@math.sc.edu).

‡Department of Mathematics, University of Bergen, Johs. Brunsgt. 12, N-5007, Bergen, Norway (helge.dahle@mi.uib.no, magne.espedal@mi.uib.no).

§Institute for Scientific Computation, Texas A&M University, College Station, TX 77843-3404 (ewing@isc.tamu.edu)

in the resulting numerical schemes. Hence, they are susceptible to time truncation errors that introduce numerical diffusion and the restrictions on the size of the Courant number, and they tend to be ineffective for transient advection-dominated problems. They generally require small time steps for reasons of accuracy, because the time truncation error depends on high-order time derivatives of the solutions that are large when a sharp front passes by. Other Eulerian methods, such as the Petrov–Galerkin FEM [74, 9] and the total variation diminishing scheme [18], attempt to reduce the overall truncation error by using negative temporal numerical diffusion to cancel positive spatial numerical diffusion. Therefore, they also suffer from the Courant number restrictions. Also included in the class of Eulerian methods is the streamline diffusion finite element method (SDM) [24, 44, 10, 45, 46, 38, 37, 47, 49, 50, 51, 52, 77, 78]. Via a framework of space-time FEM, the SDM uses piecewise polynomial trial/test functions over a partition on a space-time domain (spatial domain $\times$ current time interval). By defining the test functions delicately, the SDM adds a numerical diffusion only in the direction of characteristics (streamline) to suppress the oscillation and does not introduce any crosswind diffusion. Therefore, this method possesses many physical and numerical advantages other Eulerian methods do not have. However, this method contains an undetermined parameter in the test functions that needs to be chosen very carefully to obtain accurate numerical results. If the parameter is chosen too small, the numerical solutions will exhibit oscillations. But if it is too large, the SDM will introduce excessive numerical diffusion and seriously smear the numerical solutions. Unfortunately, an optimal choice of the parameter is not clear and is heavily problem dependent. Moreover, the number of unknowns are doubled compared to many standard Eulerian or characteristic methods.

Because of the hyperbolic nature of advective transport, characteristic analysis is natural to aid in the solution of advection-diffusion equations and has led to many related approximation techniques, including the method of characteristics of [36, 56, 6, 43]; the characteristic Galerkin method of [22, 66]; the Eulerian–Lagrangian method of [55]; the transport-diffusion method of [57]; the modified method of characteristics of [23, 28]; the operator-splitting method of [25, 75, 19]; and the Lagrangian–Galerkin method of [54]. Characteristic methods effectively solve the advective component by a characteristic tracking algorithm and treat the diffusive term separately. These methods have significantly reduced the time truncation errors in the Eulerian methods, have generated accurate numerical solutions even if large time steps are used, and have eased the Courant number restrictions of Eulerian methods. Problems with many characteristic methods arise in the areas of rigorously treating boundary fluxes when characteristics intersect inflow or outflow boundaries and of maintaining mass conservation.

The Eulerian–Lagrangian localized adjoint method (ELLAM) was first introduced by Celia et al. [14], Russell [60], and Herrera et al. [42] for the solution of one-dimensional (constant-coefficient) advection-diffusion equations. The ELLAM formalism provides a general characteristic solution procedure for advection-dominated problems, and it presents a consistent framework for treating general boundary conditions and maintaining mass conservation. Subsequently, Russell and Trujillo [61], Wang [67], and Wang, Ewing, and Russell [70] derived different ELLAM schemes for one-dimensional linear variable-coefficient advection-diffusion equations with general inflow and outflow boundary conditions based on different (forward or backward) techniques for the tracking of characteristics of the velocity field. Celia and Ferrand [12] and Healy and Russell [40] extended ELLAM to a finite-volume setting for one-

dimensional advection-diffusion equations. Ewing [27] and Dahle, Ewing, and Russell [20] addressed the ELLAM techniques for one-dimensional nonlinear advection-diffusion equations. Ewing and Wang [29, 30, 31] also developed ELLAM schemes for the solution of one-dimensional advection-reaction equations with an initial condition and inflow boundary condition. In addition, Celia and Zisman [15] and Ewing and Wang [32, 33] generalized ELLAM schemes for one-dimensional advection-diffusion-reaction transport equations. Others have applied ELLAM schemes to solve the systems of one-dimensional reactive transport problems from bioremediation and other applications [70, 64].

While the computational advantages of ELLAM approximations have been demonstrated for one-dimensional advection-dominated problems by the extensive research mentioned above, practical implementation of ELLAM schemes in multiple spatial dimensions requires careful algorithm development in which some research has been carried out in this direction. Russell and Trujillo [61] addressed various issues in multidimensional ELLAM schemes. Wang [67] developed an ELLAM simulator to solve two-dimensional linear advection-diffusion equations with general inflow and outflow boundary conditions by combining forward- and backward-tracking algorithms. Theoretically optimal-order error estimates for the derived scheme were also proved, and various numerical experiments were performed. Some of these results were reported in [30, 31, 72]. By using an explicit mapping of the finite elements at the current time level to the spatial grids at the previous time, Binning [7] and Binning and Celia [8] reported on a finite-volume ELLAM formulation for unsaturated transport in two dimensions. Relations and differences between the two approaches are discussed in some detail in section 4 of this paper. Healy and Russell developed a finite-volume ELLAM scheme for two-dimensional linear advection-diffusion equations [41]. Celia [11] also explored the development of an ELLAM scheme for three-dimensional advection-diffusion equations.

A different but related method is the "characteristic-mixed finite-element" method [1, 76, 2], which uses piecewise-constant space-time test functions. As with the standard mixed method, a coupled system results for both the concentration and the diffusive flux. The theoretically proven error estimate is $(O(\Delta x)^{3/2})$ for grid size Δx, which is suboptimal by a factor $O((\Delta x)^{1/2})$. For ELLAM schemes with piecewise linear trial/test functions for one-dimensional advection-diffusion equations, advection-diffusion-reaction equations, and first-order advection-reaction equations, optimal-order error estimates of $O((\Delta x)^2)$ have been proven by Ewing and Wang [29, 30, 34], Wang, Ewing, and Russell [70], and Wang and Ewing [69].

Based on the approach presented in [67], an ELLAM scheme is developed in this paper for the numerical solution of two-dimensional linear advection-diffusion equations with general inflow and outflow boundary conditions. We have organized this paper as follows. We begin in section 2 by presenting a space-time variational formulation of the model equations. In section 3 we derive a corresponding ELLAM scheme for this formulation with implementational issues discussed in section 4. Section 5 provides a brief description of some well-studied and widely used methods, including the Galerkin finite element method, the quadratic Petrov–Galerkin method, the cubic Petrov–Galerkin method, and the streamline diffusion finite element method. In section 6 we carry out numerical experiments and compare the performance of the ELLAM scheme with the numerical methods described in section 5. In section 7 we summarize our observations and results.

2. Variational formulation. A general linear, variable-coefficient advection-diffusion partial differential equation in two dimensions can be written as follows:

$$(R(\mathbf{x},t)u)_t + \nabla\cdot(\mathbf{V}(\mathbf{x},t)u - \mathbf{D}(\mathbf{x},t)\nabla u) = f(\mathbf{x},t), \quad (\mathbf{x},t) = (x,y,t) \in \Sigma = \Omega \times J, \tag{2.1}$$

where $u_t = \frac{\partial u}{\partial t}$, $\nabla u = (\frac{\partial u}{\partial x}, \frac{\partial u}{\partial y})^T$, Ω is a spatial domain, and $J = [0,T]$ is a time interval. The nomenclature is such that $R(\mathbf{x},t)$ is a retardation coefficient, $\mathbf{V}(\mathbf{x},t) = (V_1(\mathbf{x},t), V_2(\mathbf{x},t))$ is a fluid velocity field, $\mathbf{D}(\mathbf{x},t) = (D_{ij}(\mathbf{x},t))_{i,j=1}^2$ is a diffusion-dispersion tensor, $f(\mathbf{x},t)$ is a given forcing function, and $u(\mathbf{x},t)$ is the solute concentration of a dissolved substance. Mathematically, R has positive lower and upper bounds, $\mathbf{D}(\mathbf{x},t)$ is a symmetric and positive definite matrix with uniform lower and upper bounds that are independent of $(\mathbf{x},t)$.

Let the space-time boundary $\Gamma = \partial\Omega \times J$ be decomposed as the union of an inflow boundary $\Gamma^{(I)}$, an outflow boundary $\Gamma^{(O)}$, and a noflow boundary $\Gamma^{(N)}$ (i.e., $\Gamma = \Gamma^{(I)} \cup \Gamma^{(O)} \cup \Gamma^{(N)}$). In general, an inflow boundary during one time period might become an outflow or a noflow boundary in the next time period or vice versa. At $\Gamma^{(I)}$ or $\Gamma^{(O)}$, one of Dirichlet, Neumann, or Robin (flux) boundary conditions may be imposed by setting, respectively,

$$\begin{aligned} u(\mathbf{x},t) &= g_1^{(i)}(\mathbf{x},t), (\mathbf{x},t) \in \Gamma^{(i)}, \\ -\mathbf{D}\nabla u(\mathbf{x},t)\cdot\mathbf{n} &= g_2^{(i)}(\mathbf{x},t), (\mathbf{x},t) \in \Gamma^{(i)}, \\ (\mathbf{V}u - \mathbf{D}\nabla u)(\mathbf{x},t)\cdot\mathbf{n} &= g_3^{(i)}(\mathbf{x},t), (\mathbf{x},t) \in \Gamma^{(i)}, \end{aligned} \tag{2.2}$$

where $\mathbf{n} = \mathbf{n}(\mathbf{x})$ is the outward unit normal, $i = I$ or O represents the inflow or outflow boundary, respectively. A noflow boundary condition is specified at $\Gamma^{(N)}$ by

$$(\mathbf{V}u - \mathbf{D}\nabla u)(\mathbf{x},t)\cdot\mathbf{n} = 0, \quad (\mathbf{x},t) \in \Gamma^{(N)}. \tag{2.3}$$

In addition to the boundary conditions, an initial condition $u(\mathbf{x},0) = u_0(\mathbf{x})$ is needed to close (2.1).

The ELLAM formalism uses a time-marching algorithm. Let N_t be a positive integer. We define a partition of time interval $J = [0,T]$ by

$$0 = t_0 < t_1 < t_2 < \cdots < t_n < \cdots < t_{N_t-1} < t_{N_t} = T.$$

With space-time test functions w that vanish outside $\Sigma_n \equiv \Omega \times J_n$ with $J_n \equiv (t_{n-1}, t_n]$ and are discontinuous in time at time t_{n-1}, one can write a space-time variational formulation for (2.1) as follows:

$$\begin{aligned} &\int_\Omega (R\ u\ w)(\mathbf{x},t_n)\ d\mathbf{x} + \int_{\Sigma_n} \nabla w\cdot(\mathbf{D}\nabla u)\ d\mathbf{x}dt \\ &\quad + \int_{\Gamma_n} (\mathbf{V}u - \mathbf{D}\nabla u)\cdot\mathbf{n}\ w\ dS - \int_{\Sigma_n} u\ (R\ w_t + \mathbf{V}\cdot\nabla w)\ d\mathbf{x}dt \\ &= \int_\Omega\ (R\ u\ w)(\mathbf{x},t_{n-1}^+)\ d\mathbf{x} + \int_{\Sigma_n} f\ w\ d\mathbf{x}dt, \end{aligned} \tag{2.4}$$

where $\Gamma_n = \partial\Omega \times J_n$ and $w(\mathbf{x},t_{n-1}^+) = \lim_{t\to t_{n-1}^+} w(\mathbf{x},t)$.

In the ELLAM framework, one should define the test functions w to satisfy the equation $Rw_t + \mathbf{V}\cdot\nabla w \equiv 0$ so the last term on the left-hand side of (2.4) vanishes.

However, in general one cannot track characteristics exactly for a variable-velocity field. Nevertheless, this adjoint term should be small if one can track the characteristics reasonably well, and the test functions are constant along the approximate characteristics. In fact, we have proved an optimal-order convergence rate for the derived ELLAM scheme even if a one-step Euler algorithm is used in the characteristic tracking and this adjoint term is dropped [67, 34]. To conserve mass, the test functions should sum to one [14]. The scheme developed in this paper satisfies this condition. In this case dropping the last term on the left-hand side does not affect mass conservation since that term vanishes if $w \equiv 1$ [61, 67].

We are now in a position to rewrite (2.4). Given a point $(\bar{\mathbf{x}}, \bar{t})$ with $\bar{t} \in [t_{n-1}, t_n]$, we consider the initial-value problem for the ordinary differential equation

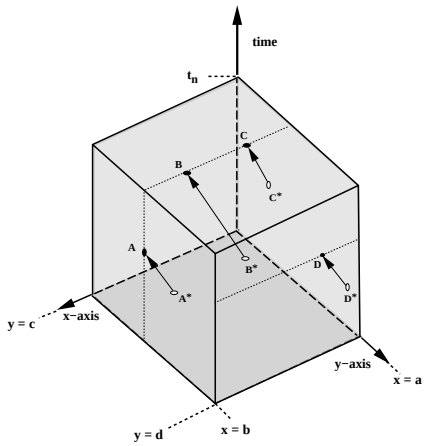
$$\begin{aligned} \frac{d\mathbf{x}}{dt} &= \mathbf{V}^R(\mathbf{x}, t) \equiv \frac{\mathbf{V}(\mathbf{x}, t)}{R(\mathbf{x}, t)}, \\ \mathbf{x}(\bar{t}) &= \bar{\mathbf{x}}, \end{aligned} \tag{2.5}$$

which tracks the characteristics from $(\bar{\mathbf{x}}, \bar{t})$. We denote the solution of this equation at time $\theta \in J_n$ by $\mathbf{X}(\theta; \bar{\mathbf{x}}, \bar{t})$ [40]. This notation can refer to tracking either forward or backward in time. In particular, we define

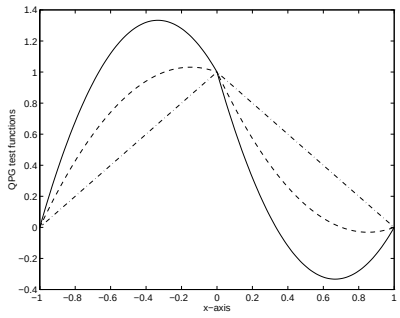
$$\begin{aligned} \mathbf{x}^* &= \mathbf{X}(t_{n-1}; \mathbf{x}, t_n), \\ \tilde{\mathbf{x}} &= \mathbf{X}(t_n; \mathbf{x}, t_{n-1}). \end{aligned} \tag{2.6}$$

Thus, $(\mathbf{x}, t_n)$ backtracks to $(\mathbf{x}^*, t_{n-1})$ and $(\mathbf{x}, t_{n-1})$ tracks forward to $(\tilde{\mathbf{x}}, t_n)$. In the numerical scheme, an exact tracking is preferred whenever possible. However, it is impractical in most applications. In practice, one can use either a one-point Euler quadrature, a multiple micro-time step tracking within a global time step, or a Runge–Kutta quadrature in the tracking of characteristics. Note that in many applications, (2.1) is usually coupled with an associated potential or pressure equation whose solution is often obtained via the mixed finite element method. In this case, a Raviart–Thomas space is often used for the velocity field, which is calculated at each cell interface. Within each cell, $V_1(\mathbf{x}, t)$ (or $V_2(\mathbf{x}, t)$) is piecewise linear (or constant) in the x direction and piecewise constant (or linear) in the y direction. Under the assumption that the velocity field is steady, a semianalytical technique has been developed [58, 62, 40] to track the characteristics on a cell-by-cell basis. Recently, Lu [53] extended this semianalytical approach to nonsteady velocity fields where velocity is assumed to vary linearly in time within each time interval.

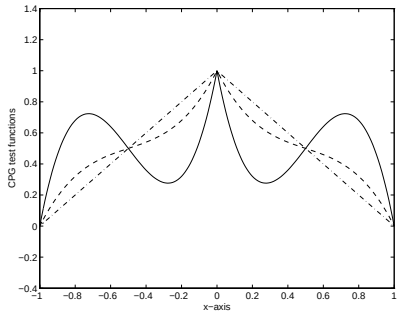
To accurately measure the time period taken for a particle to move along a characteristic from the previous time level (or from the inflow boundary) to the current time level (or the outflow boundary), we introduce space-time location-dependent time steps. We use Figure 1(a) to illustrate how these are defined. In this figure, we use letters A–D to denote points (i) at the future time step t_n (points B and C) or (ii) on the outflow boundary (points A and D), while A^*–D^* denote the corresponding feet of their characteristics. In our example, we have set the rear planes $x = a$ and $y = c$ as the inflow boundaries with the frontmost planes $x = b$ and $y = d$ as the outflow boundaries. Time is represented in the vertical direction. For any $\mathbf{x} \in \Omega$ at time t_n, we define a time step $\Delta t^{(I)}(\mathbf{x}) = \Delta t \equiv t_n - t_{n-1}$ if the characteristic $\mathbf{X}(\theta; \mathbf{x}, t_n)$ does not backtrack to the space-time boundary Γ_n during the time period J_n (this case is illustrated by point B at time t_n in Figure 1(a)), and $\Delta t^{(I)}(\mathbf{x}) = t_n - t^*(\mathbf{x})$ otherwise (see, for example, point C). In the latter case where the foot of the characteristic (point C^*) lies on an inflow boundary, $t^*(\mathbf{x}) \in J_n$ is the time when $\mathbf{X}(\theta; \mathbf{x}, t_n)$



(a) Characteristic tracking from the interior domain and inflow boundary.



(b) Test functions: -. for GAL, - - and — for QPG with $V = 1$, $D = 0.3$ and $V = 1$, $D = 10^{-4}$.



(c) Test functions: -. for GAL, - - and — for CPG with $Cu = 0.5$ and $Cu = 1$.

FIG. 1. *Illustration of characteristic tracking and test functions.*

intersects the boundary Γ_n (i.e., $\mathbf{X}(t^*(\mathbf{x});\mathbf{x},t_n)\in\Gamma_n$). Similarly, for any point on the outflow boundary $(\mathbf{x},t)\in\Gamma_n^{(O)}$ (e.g., points A or D), we define $\Delta t^{(O)}(\mathbf{x},t)=t-t_{n-1}$ if the characteristic $\mathbf{X}(\theta;\mathbf{x},t)$ does not intersect Γ_n during the time period $[t_{n-1},t]$; otherwise we set $\Delta t^{(O)}(\mathbf{x},t)=t-t^*(\mathbf{x},t)$. The first case is illustrated by point A on the space-time boundary $x=b$, while the second case is demonstrated by point D on the space-time boundary $y=d$. Here we denote by $t^*(\mathbf{x},t)\in[t_{n-1},t]$ the time when $\mathbf{X}(\theta;\mathbf{x},t)$ intersects Γ_n.

By enforcing the backward Euler quadrature at the current time t_n and at the outflow space-time boundary $\Gamma_n^{(O)}$, we approximate the space-time volume integral of the source term (the second term on the right-hand side) in (2.4) by an integral at time t_n and one at $\Gamma_n^{(O)}$ by following the characteristics. Here $\Gamma_n^{(i)}=\Gamma^{(i)}\cap J_n$ $(i=I,O,N)$ represents the space-time inflow, outflow, and noflow boundaries during the time interval J_n. To avoid confusion in the following derivation, we replace the dummy variables $\mathbf{x}\in\Omega$ and $t\in J_n$ in this term by $\mathbf{y}\in\Omega$ and $\theta\in J_n$. Thus, $\int_{\Sigma_n} f\ w(\mathbf{x},t)\ d\mathbf{x}dt=\int_{\Sigma_n} f\ w(\mathbf{y},\theta)\ d\mathbf{y}d\theta$. Let $\Sigma_n^{(O)}\subset\Sigma_n$ be the set of points in the space-time strip Σ_n that will flow out of Σ_n during the time interval J_n. We decompose Σ_n to be the union of $\Sigma_n^{(O)}$ and $\Sigma_n-\Sigma_n^{(O)}$. For any $(\mathbf{y},\theta)\in\Sigma_n-\Sigma_n^{(O)}$, there exists a point $\mathbf{x}\in\Omega$ such that $\mathbf{x}=\mathbf{X}(t_n;\mathbf{y},\theta)$. Thus, we can invert this relation to obtain $\mathbf{y}=\mathbf{X}(\theta;\mathbf{x},t_n)$. Similarly, for any $(\mathbf{y},\theta)\in\Sigma_n^{(O)}$, there exists a pair $(\mathbf{x},t)\in\Gamma_n^{(O)}$ such that $\mathbf{y}=\mathbf{X}(\theta;\mathbf{x},t)$. By splitting the space-time volume integral on Σ_n as one on $\Sigma_n-\Sigma_n^{(O)}$ and one on $\Sigma_n^{(O)}$ and applying the backward Euler quadrature at time t_n for the first and at boundary $\Gamma_n^{(O)}$ for the second, we obtain the following equation:

$$
\begin{aligned}
&\int_{\Sigma_n} f(\mathbf{y},\theta)w(\mathbf{y},\theta)\ d\mathbf{y}d\theta\\
&\quad=\int_{\Sigma_n-\Sigma_n^{(O)}} f(\mathbf{y},\theta)w(\mathbf{y},\theta)\ d\mathbf{y}d\theta+\int_{\Sigma_n^{(O)}} f(\mathbf{y},\theta)w(\mathbf{y},\theta)\ d\mathbf{y}d\theta\\
&\quad=\int_{\Sigma_n-\Sigma_n^{(O)}} f(\mathbf{X}(\theta;\mathbf{x},t_n),\theta)\ w(\mathbf{X}(\theta;\mathbf{x},t_n),\theta)\ d\mathbf{X}d\theta\\
&\qquad+\int_{\Sigma_n^{(O)}} f(\mathbf{X}(\theta;\mathbf{x},t),\theta)\ w(\mathbf{X}(\theta;\mathbf{x},t),\theta)\ d\mathbf{X}d\theta\\
&\quad=\int_{\Omega}\Delta t^{(I)}(\mathbf{x})\ f_n\ w_n(\mathbf{x})\ d\mathbf{x}+\int_{\Gamma_n^{(O)}}\Delta t^{(O)}(\mathbf{x},t)\ f\ w\ \mathbf{V}\cdot\mathbf{n}\ dS+E_f(w).
\end{aligned}
\tag{2.7}
$$

Here $f_n(\mathbf{x})=f(\mathbf{x},t_n)$, E_f is the truncation error from the application of the backward Euler quadrature. In the derivation of (2.7), we have used the fact that the test function w is constant along the characteristics.

Similarly, we can rewrite the diffusion-dispersion term as

$$
\begin{aligned}
&\int_{\Sigma_n}\nabla w\cdot(\mathbf{D}\nabla u)(\mathbf{y},\theta)\ d\mathbf{y}d\theta\\
&\quad=\int_{\Omega}\Delta t^{(I)}(\mathbf{x})\ \nabla w_n\cdot(\mathbf{D}_n\nabla u_n)(\mathbf{x})\ d\mathbf{x}\\
&\qquad+\int_{\Gamma_n^{(O)}}\Delta t^{(O)}(\mathbf{x},t)\ \nabla w\cdot(\mathbf{D}\nabla u)\ \mathbf{V}\cdot\mathbf{n}\ dS+E_{\mathbf{D}}(u,w),
\end{aligned}
\tag{2.8}
$$

where $E_{\mathbf{D}}(u,w)$ is the truncation error term.

Substituting (2.7) and (2.8) for the second terms on both the left- and right-hand sides of (2.4), we obtain the following variational formulation:

$$\begin{aligned}
&\int_\Omega R_n u_n\ w_n\ d\mathbf{x} + \int_\Omega \Delta t^{(I)}(\mathbf{x})\ \nabla w_n \cdot (\mathbf{D}_n \nabla u_n)\ d\mathbf{x} \\
&\qquad + \int_{\Gamma_n^{(O)}} \Delta t^{(O)}(\mathbf{x},t)\ \nabla w \cdot (\mathbf{D}\nabla u)\ \mathbf{V}\cdot\mathbf{n}\ dS \\
&\qquad + \int_{\Gamma_n} (\mathbf{V}u - \mathbf{D}\nabla u)\cdot\mathbf{n}\ w\ dS - \int_{\Sigma_n} u\ (R\ w_t + \mathbf{V}\cdot\nabla w)\ d\mathbf{x}dt \\
&= \int_\Omega R_{n-1}u_{n-1}\ w_{n-1}^+\ d\mathbf{x} + \int_\Omega \Delta t^{(I)}(\mathbf{x})\ f_n\ w_n\ d\mathbf{x} \\
&\qquad + \int_{\Gamma_n^{(O)}} \Delta t^{(O)}(\mathbf{x},t)\ f\ w\ \mathbf{V}\cdot\mathbf{n}\ dS + E(u,w),
\end{aligned} \tag{2.9}$$

where $E(u,w) = -E_{\mathbf{D}}(u,w) + E_f(w)$.

3. An ELLAM scheme. While the numerical scheme can be derived for a general domain Ω with a quasi-uniform triangular or rectangular partition, we assume the domain $\Omega = (a,b)\times(c,d)$ for simplicity to be a rectangular domain with a uniform rectangular partition:

$$\begin{aligned}
x_i &= a + i\Delta x, \quad i = 0,1,\ldots,N_x, \quad \Delta x = \frac{b-a}{N_x}, \\
y_j &= c + j\Delta y, \quad j = 0,1,\ldots,N_y, \quad \Delta y = \frac{d-c}{N_y},
\end{aligned}$$

where N_x and N_y are two positive integers. We define the test functions w_{ij} to be piecewise-linear "hat" functions at time t_n ($w_{ij}(\mathbf{x}_{kl}, t_n) = \delta_{ik}\delta_{jl}$, where $\mathbf{x}_{kl} = (x_k, y_l)$, $\delta_{ik} = 1$ if $i = k$ and 0 otherwise) and to be constant along the characteristics. At time t_n, we also use piecewise-linear trial functions $U(\mathbf{x}, t_n)$.

3.1. Interior nodes and noflow boundary. In this subsection we develop the scheme at the nodes inside Ω or on the noflow boundary $\Gamma_n^{(N)}$ that are related to neither the inflow boundary $\Gamma_n^{(I)}$ nor the outflow boundary $\Gamma_n^{(O)}$. It is assumed that the type of boundary (inflow, outflow, or noflow) will be kept unchanged during the time interval J_n. Let

$$\begin{aligned}
\Gamma_n(q) &= \left\{(\mathbf{x},t)\in\Gamma_n \,\middle|\, x = q\right\} \equiv \left\{(\mathbf{x},t)\,\middle|\, x = q, y\in[c,d], t\in J_n\right\},\ q = a, b, \\
\Gamma_n(q) &= \left\{(\mathbf{x},t)\in\Gamma_n \,\middle|\, y = q\right\} \equiv \left\{(\mathbf{x},t)\,\middle|\, x\in[a,b], y = q, t\in J_n\right\},\ q = c, d.
\end{aligned} \tag{3.1}$$

We define the Courant numbers

$$\begin{aligned}
Cu_x^{(q)} &= \max_{(\mathbf{x},t)\in\Gamma_n(q)} \frac{|V_1(\mathbf{x},t)|\Delta t}{\Delta x} \quad \text{for } q = a,\ b, \\
Cu_y^{(q)} &= \max_{(\mathbf{x},t)\in\Gamma_n(q)} \frac{|V_2(\mathbf{x},t)|\Delta t}{\Delta y} \quad \text{for } q = c,\ d.
\end{aligned} \tag{3.2}$$

If $\Gamma_n(q)$ ($q = a, b$) is an inflow or outflow boundary (which implies that $Cu_x^{(q)} > 0$), we define $IC_x^{(q)}$ to be $IC_x^{(q)} = [Cu_x^{(q)}] + 1$, where $[Cu_x^{(q)}]$ is the integer part of $Cu_x^{(q)}$.

If $\Gamma_n(q)$ is a noflow boundary (which indicates that $Cu_x^{(q)} = 0$), we define $IC_x^{(q)} = 0$. $IC_y^{(q)}$ is defined similarly. In addition, we define

$$\begin{aligned} \hat{\Omega} &= [a + IC_x^{(a)}\Delta x, b - IC_x^{(b)}\Delta x] \times [c + IC_y^{(c)}\Delta y, d - IC_y^{(d)}\Delta y], \\ \Omega_{ij} &= [x_{i-1}, x_{i+1}] \times [y_{j-1}, y_{j+1}]. \end{aligned} \tag{3.3}$$

Furthermore, let Σ_{nij}^* be the prism obtained by backtracking Ω_{ij} along the characteristics from t_n to t_{n-1} and $\tilde{\Sigma}_{nij}$ be the prism obtained by tracing Ω_{ij} forward along the characteristics from t_{n-1} to t_n.

When $\Omega_{ij} \subset \hat{\Omega}$, Σ_{nij}^* or $\tilde{\Sigma}_{nij}$ does not intersect $\Gamma_n^{(I)}$ or $\Gamma_n^{(O)}$ during the time period J_n. The third and fourth terms on the left-hand side and the third term on the right-hand side of (2.9) vanish. Dropping the last terms on both sides of (2.9), replacing the exact solution u and the general test function w by the piecewise-linear trial function U and test function w_{ij}, we obtain the following equation:

$$\begin{aligned} &\int_\Omega R_n U_n \; w_{ijn}(\mathbf{x}) \; d\mathbf{x} + \int_\Omega \Delta t \; \nabla w_{ijn} \cdot (\mathbf{D}_n \nabla U_n)(\mathbf{x}) \; d\mathbf{x} \\ &\quad = \int_\Omega \; R_{n-1} U_{n-1} \; w_{ij,n-1}^+(\mathbf{x}) \; d\mathbf{x} + \int_\Omega \Delta t \; f_n \; w_{ijn}(\mathbf{x}) \; d\mathbf{x} \end{aligned} \tag{3.4}$$

with $w_{ijn}(\mathbf{x}) = w_{ij}(\mathbf{x}, t_n)$. Note that in (3.4), the integrals at time t_n are actually defined on Ω_{ij} (with the obvious modification near the boundary $\partial\Omega$), since Ω_{ij} is the support of w_{ijn}. The first term on the right-hand side is actually defined on the backtracked image (at time t_{n-1}) of Ω_{ij} at time t_n, which can be of a very complicated shape and not aligned with any elements in Ω at time t_{n-1} due to the effect of the velocity field, even though Ω_{ij} is rectangular. Consequently, the evaluation of this term is tricky and, in fact, crucial to the accuracy and mass conservation property of the scheme. This will be discussed in detail in section 4. At this point, one can easily see that the scheme has a 9-banded, symmetric, and positive-definite coefficient matrix.

3.2. Inflow boundary conditions. In contrast to many characteristic methods that treat boundary conditions in an ad hoc manner, the ELLAM scheme naturally incorporates boundary conditions into its formulation. Thus, one can approximate boundary conditions accurately. In fact, if Σ_{nij}^* intersects the inflow boundary $\Gamma_n^{(I)}$, the test function w_{ij} assumes nonzero values on portions of Γ_n. Thus, the fourth term on the left-hand side of (2.9) does not vanish. For an inflow flux boundary condition, the scheme becomes

$$\begin{aligned} &\int_\Omega R_n U_n \; w_{ijn}(\mathbf{x}) \; d\mathbf{x} + \int_\Omega \Delta t^{(I)}(\mathbf{x}) \; \nabla w_{ijn} \cdot (\mathbf{D}_n \nabla U_n)(\mathbf{x}) \; d\mathbf{x} \\ &\quad = \int_\Omega R_{n-1} U_{n-1} \; w_{ij,n-1}^+(\mathbf{x}) \; d\mathbf{x} + \int_\Omega \Delta t^{(I)}(\mathbf{x}) \; f_n \; w_n(\mathbf{x}) \; d\mathbf{x} \\ &\qquad - \int_{\Gamma_n^{(I)}} \; g_3^{(I)} \; w_{ij}(\mathbf{x}, t) \; dS. \end{aligned} \tag{3.5}$$

Keep in mind that the first term on the right-hand side of (3.5) is now defined on the image (at time t_{n-1}) of the portion of Ω_{ij} that is not taken to the boundary Γ_n. The part of the integral that is missing from this term is picked up by the last term on the right-hand side of (3.5), which is defined on the image of the portion of

Ω_{ij} which is taken to the boundary Γ_n. Notice that the factor $\Delta t^{(I)}(\mathbf{x})$ at time t_n now depends on $\mathbf{x}$, since $\mathbf{X}(\theta;\mathbf{x},t_n)$ can encounter the boundary Γ_n. $\Delta t^{(I)}(\mathbf{x})$ reflects the time period over which the diffusion-dispersion and source act. For an inflow flux boundary condition the derived scheme still has a 9-banded, symmetric, and positive-definite coefficient matrix.

Repeating the above derivation for an inflow Dirichlet boundary condition yields the following equation:

$$\begin{aligned}
&\int_\Omega R_n U_n\ w_{ijn}(\mathbf{x})\ d\mathbf{x} + \int_\Omega \Delta t^{(I)}(\mathbf{x})\ \nabla w_{ijn}\cdot(\mathbf{D}_n\nabla U_n)(\mathbf{x})\ d\mathbf{x}\\
&\qquad - \int_{\Gamma_n^{(I)}} (\mathbf{D}\nabla U)\cdot\mathbf{n}\ w_{ij}(\mathbf{x},t)\ dS \qquad (3.6)\\
&= \int_\Omega R_{n-1}U_{n-1}\ w^+_{ij,n-1}(\mathbf{x})\ d\mathbf{x} + \int_\Omega \Delta t^{(I)}(\mathbf{x})\ f_n\ w_{ijn}(\mathbf{x})\ d\mathbf{x}\\
&\qquad - \int_{\Gamma_n^{(I)}} \mathbf{V}\cdot\mathbf{n}\ g_1^{(I)}\ w_{ij}(\mathbf{x},t)\ dS.
\end{aligned}$$

While all other terms in (3.6) are similar to those in (3.5), the third term on the left-hand side couples the unknown boundary diffusive flux with unknown interior function values. If one simply represents ∇U as a discrete gradient dependent on imposed boundary values of U, one might introduce strong temporal truncation errors. To overcome this difficulty, we approximate $\nabla U(\mathbf{x},t)$ at the inflow boundary $\Gamma_n^{(I)}$ implicitly by $\nabla U(\mathbf{X}(t_n;\mathbf{x},t),t_n)$ at time t_n. This removes the difficulty of evaluating an unknown diffusive boundary flux. The error introduced is small since it is along the characteristics and, in fact, does not affect the convergence rate of the scheme [70]. Note that this term introduces nonsymmetry to the coefficient matrix near the inflow boundary.

As with the standard finite element methods, the Dirichlet boundary condition is essential and is imposed directly on the solution u with no degrees of freedom on the inflow boundary $\Gamma_n^{(I)}$. However, the test functions should sum to one to conserve mass [14]. Thus, on each element $\Omega_{ij} = [x_{i-1}, x_{i+1}]\times[y_{j-1}, y_{j+1}]$ that has at least one vertex on the inflow boundary $\Gamma_n^{(I)}$, the test functions are chosen such that they sum to one on this element. For example, suppose $x = x_0 \equiv a$ is an inflow boundary. Then the interior nodes $\mathbf{x}_{1,j} = (x_1, y_j)$ with $x_1 = a+\Delta x$ and $1 \le j \le N_y - 1$ are adjacent to the inflow boundary $x = a$. In this case, the corresponding test functions must satisfy $w_{1j}(x,y) = w_{1j}(x_1,y)$, i.e., they are constant in x direction over the interval $[a, x_1]$.

A derivation similar to that of (3.5) yields a scheme for (2.1) with an inflow Neumann boundary condition. This differs from (3.5) in that $g_3^{(I)}$ is replaced by $g_2^{(I)}$ and an extra term $\int_{\Gamma_n^{(I)}} U\ w_{ij}\ \mathbf{V}\cdot\mathbf{n}\ dS$ appears on the left-hand side of the equations. Because $\mathbf{V}\cdot\mathbf{n}|_{\Gamma_n^{(I)}} < 0$, this term has a different sign from the first term on the left-hand side of (3.5).

If $\Gamma_n^{(I)}$ can be decomposed as $\Gamma_n^{(I)} = \Gamma_{n,1}^{(I)} \cup \Gamma_{n,2}^{(I)} \cup \Gamma_{n,3}^{(I)}$, where inflow Dirichlet, Neumann, and Robin boundary conditions are imposed on $\Gamma_{n,1}^{(I)}$, $\Gamma_{n,2}^{(I)}$, and $\Gamma_{n,3}^{(I)}$, respectively, one can write out the scheme accordingly.

3.3. Outflow boundary conditions. The situation at the outflow boundary $\Gamma_n^{(O)}$ is different from that at an inflow boundary $\Gamma_n^{(I)}$. The number of spatial degrees of freedom crossing the outflow boundary $\Gamma_n^{(O)}$ is essentially the Courant number in

the normal direction. To preserve the information, one should discretize in time at the outflow boundary $\Gamma_n^{(O)}$ with about the same number of degrees of freedom. More precisely, we define

$$Cu^{(O)} = \max_{(\mathbf{x},t)\in\Gamma_n^{(O)}} \left\{ \frac{|V_1(\mathbf{x},t)|\Delta t}{\Delta x}, \frac{|V_2(\mathbf{x},t)|\Delta t}{\Delta y} \right\} \tag{3.7}$$

and $IC^{(O)} = [Cu^{(O)}] + 1$. Then we define a uniform local refinement in time at the outflow boundary $\Gamma_n^{(O)}$

$$t_{n,i} = t_n - \frac{i\Delta t}{IC} \quad \text{for } \ i = 0, 1, \ldots, IC.$$

Of course, if one is not interested in accurate simulation near the outflow boundary $\Gamma_n^{(O)}$, one need not refine in time at $\Gamma_n^{(O)}$. This corresponds to the choice of $IC = 1$. In any case, we define the test functions w_{ij} to be the piecewise-linear hat functions at the nodes at the outflow boundary $\Gamma_n^{(O)}$ and to be constant along the characteristics. We define the trial functions $U(\mathbf{x},t)$ for $(\mathbf{x},t) \in \Gamma_n^{(O)}$ to be the piecewise-linear functions at $\Gamma_n^{(O)}$. Incorporating the outflow Neumann boundary condition into (2.9) yields a scheme for (2.1) with the stated boundary condition as follows:

$$\begin{aligned}
&\int_\Omega R_n U_n \; w_{ijn}(\mathbf{x}) \; d\mathbf{x} + \int_\Omega \Delta t^{(I)}(\mathbf{x}) \; \nabla w_{ijn} \cdot (\mathbf{D}_n \nabla U_n)(\mathbf{x}) \; d\mathbf{x} \\
&\quad + \int_{\Gamma_n^{(O)}} \Delta t^{(O)}(\mathbf{x},t) \; \nabla w_{ij} \cdot (\mathbf{D}\nabla U) \; \mathbf{V}\cdot\mathbf{n}(\mathbf{x},t) \; dS \\
&\quad + \int_{\Gamma_n^{(O)}} U \; w_{ij} \; \mathbf{V}\cdot\mathbf{n}(\mathbf{x},t) \; dS \\
&= \int_\Omega R_{n-1} U_{n-1} \; w^+_{ij,n-1}(\mathbf{x}) \; d\mathbf{x} + \int_\Omega \Delta t^{(I)}(\mathbf{x}) \; f_n \; w_{ijn}(\mathbf{x}) \; d\mathbf{x} \\
&\quad + \int_{\Gamma_n^{(O)}} \Delta t^{(O)}(\mathbf{x},t) \; f \; w_{ij} \; \mathbf{V}\cdot\mathbf{n}(\mathbf{x},t) \; dS - \int_{\Gamma_n^{(O)}} g_2^{(O)} \; w_{ij}(\mathbf{x},t) \; dS.
\end{aligned} \tag{3.8}$$

Because U, not ∇U, is defined as unknowns at the outflow boundary $\Gamma_n^{(O)}$, it is difficult to approximate $\nabla U \cdot \mathbf{n}|_{\Gamma_n^{(O)}}$ numerically. To circumvent this difficulty, we utilize the boundary condition (2.2) to express $\nabla U \cdot \mathbf{n}|_{\Gamma_n^{(O)}}$ in terms of $U|_{\Gamma_n^{(O)}}$ and the tangential component of $\nabla U|_{\Gamma_n^{(O)}}$, which can be computed by differentiating U on $\Gamma_n^{(O)}$. To demonstrate these ideas, we assume that $\Gamma_n(b)$ (i.e., the "eastern" boundary $x = b$) is an outflow boundary. The outflow Neumann boundary condition in (2.2) now reads

$$-\mathbf{D}\nabla u \cdot \mathbf{n} \equiv -D_{11}u_x - D_{12}u_y = g_2^{(O)},$$

from which one can express $\nabla u \cdot \mathbf{n}|_{\Gamma_n^{(b)}} = u_x$ in terms of the tangential component of ∇u (u_y in this case) and u as follows:

$$u_x = -\frac{D_{12}u_y + g_2^{(O)}}{D_{11}}.$$

This yields

$$\mathbf{D}\nabla u = \left(-g_2^{(O)}, \frac{|\mathbf{D}| u_y - D_{21} g_2^{(O)}}{D_{11}} \right)^T,$$

where $|\mathbf{D}| = D_{11}D_{22} - D_{12}D_{21}$ is the determinant of $\mathbf{D}$.

Using the facts that the test functions w_{ij} satisfy the equation $Rw_t + \mathbf{V} \cdot \nabla w = 0$ and that $V_1|_{\Gamma_n^{(b)}} > 0$, since $\Gamma_n^{(b)}$ is an outflow boundary, we can denote $\nabla w_{ij} \cdot \mathbf{n}|_{\Gamma_n^{(b)}}$ in the third term on the left-hand side of (3.8) by $w_x = -(Rw_t + V_2 w_y)/V_1$. Then we can rewrite the third term on the left-hand side of (3.8) as

$$\begin{aligned} &\int_{\Gamma_n^{(b)}} \Delta t^{(O)}(\mathbf{x}, t)\ \nabla w_{ij} \cdot (\mathbf{D}\nabla U)\ \mathbf{V} \cdot \mathbf{n}(\mathbf{x}, t)\ dS \\ &\quad = \int_{\Gamma_n^{(b)}} \Delta t^{(O)}(\mathbf{x}, t)\ \frac{V_1 |\mathbf{D}|}{D_{11}}\ w_{ijy} U_y(\mathbf{x}, t)\ dS \\ &\qquad + \int_{\Gamma_n^{(b)}} g_2^{(O)} \left[(Rw_{ijt} + V_2 w_{ijy}) - \frac{V_1 D_{21}}{D_{11}} w_{ijy} \right] (\mathbf{x}, t) dS. \end{aligned} \tag{3.9}$$

Substituting this equation for the third term on the left-hand side of (3.8), we obtain a numerical scheme for (2.1) with an outflow Neumann boundary condition. The derived scheme has a symmetric and positive-definite coefficient matrix.

Since the numerical solution U is known at time t_{n-1} from the computation at time t_{n-1}, there are no degrees of freedom on the boundary $\Gamma_n^{(O)}$ at time t_{n-1}. To conserve mass, the test functions on $\Gamma_n^{(O)}$ that intersect Ω at time t_{n-1} are chosen such that they sum to one; this was discussed following (3.6).

Incorporating the flux boundary condition into (2.9), one can derive a scheme similar to (3.8), except that the last term on its left-hand side disappears and $g_2^{(O)}$ is replaced by $g_3^{(O)}$. Again, we need to express $\nabla u \cdot \mathbf{n}|_{\Gamma_n^{(O)}}$ and $\nabla w_{ij} \cdot \mathbf{n}|_{\Gamma_n^{(O)}}$ by their tangential derivatives and functional values. If we still assume that $\Gamma_n^{(b)}$ is an outflow boundary, the outflow flux boundary condition in (2.2) now becomes

$$D_{11} u_x + D_{12} u_y = V_1\ u - g_3^{(O)},$$

which yields

$$u_x = -\frac{D_{12}}{D_{11}} u_y + \frac{V_1\ u - g_3^{(O)}}{D_{11}}.$$

Combining these two equations gives

$$\mathbf{D}\ \nabla u = \left(V_1 u - g_3^{(O)}, \frac{|\mathbf{D}|}{D_{11}} u_y + \frac{D_{21}}{D_{11}} \left(V_1 u - g_3^{(O)} \right) \right)^T.$$

Similar derivation to (3.9) results in the following equation:

$$\begin{aligned} &\int_{\Gamma_n^{(b)}} \Delta t^{(O)}(\mathbf{x}, t)\ \nabla w_{ij} \cdot (\mathbf{D}\nabla U)\ \mathbf{V} \cdot \mathbf{n}(\mathbf{x}, t)\ dS \\ &\quad = \int_{\Gamma_n^{(b)}} \Delta t^{(O)}(\mathbf{x}, t) \left[\frac{V_1 |\mathbf{D}|}{D_{11}} w_{ijy} U_y \right. \end{aligned}$$

$$+\frac{V_1^2 D_{21}}{D_{11}} w_{ijy} U - V_1 (R w_{ijt} + V_2 w_{ijy}) U \Big] (\mathbf{x}, t) dS \tag{3.10}$$
$$+ \int_{\Gamma_n^{(b)}} g_3^{(O)} \left[(R w_{ijt} + V_2 w_{ijy}) - \frac{V_1 D_{21} w_{ijy}}{D_{11}} \right] (\mathbf{x}, t) dS.$$

Substituting (3.10) for the third term on the left-hand side of (3.8), dropping the last term on its left-hand side, and replacing $g_2^{(O)}$ by $g_3^{(O)}$, we obtain a numerical scheme for (2.1) with an outflow flux boundary condition.

For (2.1) with an outflow Dirichlet boundary condition, the equations at the outflow boundary define the unknowns to be the normal derivatives of the solutions and are decoupled from the equations at the interior domain given by (3.4). They are omitted here since they are needed only for mass conservation.

If $\Gamma_n^{(O)}$ can be decomposed as $\Gamma_n^{(I)} = \Gamma_{n,1}^{(O)} \cup \Gamma_{n,2}^{(O)} \cup \Gamma_{n,3}^{(O)}$, where outflow Dirichlet, Neumann, and Robin boundary conditions are imposed on $\Gamma_{n,1}^{(O)}$, $\Gamma_{n,2}^{(O)}$, and $\Gamma_{n,3}^{(O)}$, respectively, one can write out the scheme accordingly.

4. Implementation. In this section we address some practical implementational issues, which arise due to the use of characteristic tracking.

4.1. Evaluation of integrals and tracking algorithms. Some integrals in the numerical scheme derived in section 3 are standard in FEM and can be evaluated fairly easily, while others can be difficult. In this subsection, we discuss the evaluation of the integrals in (3.4), and we discuss the treatment of boundary terms in (3.5)–(3.10) in the next subsection.

Note that the trial function $U(\mathbf{x}, t_n)$ and test functions $w_{ij}(\mathbf{x}, t_n)$ are defined as standard tensor products of piecewise-linear functions at time t_n; the integrals in (3.4) are standard in finite element methods, except for the first term on the right-hand side. In this term, the value of $U(\mathbf{x}, t_{n-1})$ is known from the solution at time t_{n-1}. However, keep in mind that the test functions $w_{ij,n-1}^{+} = \lim_{t \to t_{n-1}^{+}} w_{ij}(\mathbf{x}, t) = w_{ij}(\tilde{\mathbf{x}}, t_n)$, where $\tilde{\mathbf{x}} = \mathbf{X}(t_n; \mathbf{x}, t_{n-1})$ is the point at the head corresponding to $\mathbf{x}$ at the foot. The evaluation of this term becomes much more challenging in multiple dimensions due to the multidimensional deformation of each finite element Ω_{ij} on which the test functions are defined as the geometry is backtracked from time t_n to time t_{n-1}.

In modified method of characteristics and many other characteristic schemes, this term has traditionally been rewritten as an integral at time t_n with the standard value of $w_{ij}(\mathbf{x}, t_n)$ but backtracking to evaluate $U(\mathbf{x}^*, t_{n-1})$ where $\mathbf{x}^* = \mathbf{X}(t_{n-1}; \mathbf{x}, t_n)$ is the point at the foot corresponding to $\mathbf{x}$ at the head [19, 23, 25]. As a matter of fact, it has been shown that in characteristic methods the backward-tracking algorithm is critical in the evaluation of this term, which is in turn critical to the accuracy of the scheme [4]. Because of this, the backward-tracking algorithm has been used in many ELLAM works [7, 14, 19, 29, 30, 31, 33, 60, 68, 71]. However, for multidimensional problems the evaluation of this term with a backtracking algorithm requires significant effort, due to the need to define the geometry at time t_{n-1}, which requires mapping of points along the boundary of the element and subsequent interpolation and mapping onto the fixed spatial grid at the previous time level t_{n-1}. Binning [7] and Binning and Celia [8] used such a mapping in two dimensions in a procedure that was computationally very intensive, especially when part or all of the element being mapped intersects a space-time boundary Γ_n. This approach is considered impractical in two and three dimensions [7, 11]. For one-dimensional problems, the evaluation of this term is relatively simple since the boundaries of the spatial elements are points

rather than lines or surfaces. In this case, these problems were overcome in the works cited above.

The most practical approach for evaluating this term is to use a forward-tracking algorithm, which was proposed by Russell and Trujillo [61] and was implemented by Healy and Russell for a one-dimensional problem [40] and by Ewing and Wang [30, 31] and Wang [67] for a two-dimensional problem. This would enforce the integration quadrature at t_{n-1} with respect to a fixed spatial grid on which R_{n-1} and U_{n-1} are defined, the difficult evaluation is the test function $w^+_{ij,n-1}$. Rather than backtracking the geometry and estimating the test functions by mapping the deformed geometry onto the fixed grid, discrete quadrature points chosen on the fixed grid at t_{n-1} in a regular fashion (say, standard Gaussian points) can be forward-tracked to time t_n, where evaluation of w_{ij} is straight-forward. Algorithmically, this is implemented by evaluating R and U at a quadrature point $\mathbf{x}_p$ at time t_{n-1}, then tracking the point $\mathbf{x}_p$ from t_{n-1} to $\tilde{\mathbf{x}}_p = \mathbf{X}(t_n; \mathbf{x}_p, t_{n-1})$ at t_n and determining which test functions are nonzero at $\tilde{\mathbf{x}}_p$ at t_n, so that the amount of mass associated with $\mathbf{x}_p$ can be added to the corresponding position in the right-hand side vector in the global discrete linear algebraic system. Notice that this forward-tracking has no effect on the solution grid or the data structure of the discrete system. Therefore, the forward-tracking algorithm used here does not suffer from the complication of distorted grids, which complicates many forward tracking algorithms, and is a major attraction of the backtracking in characteristic methods.

4.2. Inflow boundaries. If $\Omega_{ij} \not\subset \hat{\Omega}$, either Σ^*_{nij} intersects the inflow boundary $\Gamma^{(I)}_n$ or $\tilde{\Sigma}_{nij}$ intersects the outflow boundary $\Gamma^{(O)}_n$. First, consider the former case given by (3.5) or (3.6).

The first term on the left-hand side of (3.5) is standard in finite element methods. The second terms on both sides are standard, except that the time step $\Delta t^{(I)}(\mathbf{x})$ defined below (2.6) depends on $\mathbf{x}$. In the numerical implementation, we calculate these integrals with quadrature points at time t_n. Hence, we evaluate $\Delta t^{(I)}(\mathbf{x})$ by backtracking at these points. For each quadrature point $\mathbf{x}_p \in \Omega_{ij}$ at time t_n, we need to track the characteristic $\mathbf{X}(\theta; \mathbf{x}_p, t_n)$ for $\theta \in J_n$ to determine if it reaches the boundary Γ_n or not. If so, we calculate the time $t^*(\mathbf{x}_p)$ when the characteristic reaches the boundary Γ_n and assign $\Delta t^{(I)}(\mathbf{x}_p) = t_n - t^*(\mathbf{x}_p)$; otherwise, $\Delta t^{(I)}(\mathbf{x}_p) = \Delta t$. Notice that the backtracking algorithm is used only to calculate $\Delta t^{(I)}(\mathbf{x})$, which appears in the diffusion-dispersion term, and does not affect mass conservation. The first term on the right-hand side of (3.5) can still be evaluated by a forward-tracking algorithm as in section 4.1.

Notice that in the last term on the right-hand side of (3.5), $g^{(I)}_3(\mathbf{x}, t)$ is defined at the space-time boundary $\Gamma^{(I)}_n$, but the test function $w_{ij}(\mathbf{x}, t) = w_{ij}(\tilde{\mathbf{x}}, t_n)$, where $\tilde{\mathbf{x}} = \mathbf{X}(t_n; \mathbf{x}, t)$ is the point at the head at time t_n, corresponds to the point $\mathbf{x}$ at the foot at time t. Therefore, we use a forward-tracking algorithm to calculate this term. This would enforce the integration quadrature at the space-time boundary $\Gamma^{(I)}_n$ with respect to a fixed spatial grid on which $g^{(I)}_3(\mathbf{x}, t)$ is defined and track forward the discrete quadrature points chosen on the fixed grid at the space-time boundary $\Gamma^{(I)}_n$ in a regular fashion to time t_n, where one evaluates w_{ij}.

Except for the last term on its left-hand side, the terms in (3.6) are similar to those in (3.5). The evaluation of this term is the same as that for the last term on the right-hand side of (3.5), except that one needs to use forward tracking to evaluate both ∇U and the test function w_{ij}.

4.3. Outflow boundaries. Consider the case when $\Omega_{ij} \not\subset \hat{\Omega}$ and $\tilde{\Sigma}_{nij}$ intersects the outflow boundary $\Gamma_n^{(O)}$. The numerical scheme is given by (3.8)–(3.10). We discuss only the evaluation of the integrals in (3.8) and (3.9) since the evaluation of the integrals in (3.10) is similar.

The first term on the left-hand side of (3.8) is standard. The second terms on both sides of (3.8) can be calculated as in section 4. Keep in mind that the integrals are local, even though they are expressed as the integrals on Ω at time t_n. Hence, $\Delta t^{(I)}(\mathbf{x}) = \Delta t$, except at the corner of $\Omega = (a,b) \times (c,d)$ where $\Gamma_n^{(O)}$ and $\Gamma_n^{(I)}$ intersect. The fact that inflow and outflow boundaries can intersect in multiple spatial dimensions makes the implementation more complicated than that for one-dimensional problems where inflow and outflow boundaries do not meet (as long as the one-dimensional velocity field keeps a definite sign). As a result, in evaluating the second terms on both sides of (3.8), we need to use a backward-tracking algorithm to calculate $\Delta t^{(I)}(\mathbf{x})$ near the corner of Ω where the inflow boundary $\Gamma_n^{(I)}$ and outflow boundary $\Gamma_n^{(O)}$ meet.

In (3.8), the four integrals defined on $\Gamma_n^{(O)}$ (with the first $\Gamma_n^{(O)}$ integral given by (3.9)) are standard since both the trial function U and the test functions w_{ij} are defined on $\Gamma_n^{(O)}$. We would enforce the integration quadrature on $\Gamma_n^{(O)}$. Recall that the factor $\Delta t^{(O)}(\mathbf{x},t)$ in some of these terms is defined by (below (2.6)) $\Delta t^{(O)}(\mathbf{x},t) = t - t_{n-1}$, except when the characteristic $\mathbf{X}(\theta;\mathbf{x},t)$ meets $\Gamma_n^{(I)}$. In this case $\Delta t^{(O)}(\mathbf{x},t) = t - t^*(\mathbf{x},t)$, where $t^*(\mathbf{x},t) \in J_n$ is the time when $\mathbf{X}(\theta;\mathbf{x},t)$ intersects $\Gamma_n^{(I)}$. In the numerical implementation, we simply let $\Delta t^{(O)}(\mathbf{x},t) = t - t_{n-1}$, except near the corner where the inflow boundary $\Gamma_n^{(I)}$ and the outflow boundary $\Gamma_n^{(O)}$ intersect. At the corner region, we use a backward-tracking algorithm to locate $t^*(\mathbf{x},t)$ and let $\Delta t^{(O)}(\mathbf{x},t) = t - t^*(\mathbf{x},t)$. As mentioned in section 4, the use of backtracking in the calculation of $\Delta t^{(I)}(\mathbf{x})$ and $\Delta t^{(O)}(\mathbf{x},t)$ does not effect mass conservation.

The first term on the right-hand side of (3.8) can be evaluated by a forward-tracking algorithm as in sections 4.1–4.2. However, notice that at each quadrature point $\mathbf{x}_p \in \Omega_{ij}$ at time t_{n-1}, the characteristic $\mathbf{X}(\theta;\mathbf{x}_p,t_{n-1})$ may intersect $\Gamma_n^{(O)}$. In the current context, we need to use a forward tracking to determine if $\mathbf{X}(\theta;\mathbf{x}_p,t_{n-1})$ will or will not intersect $\Gamma_n^{(O)}$. In the latter case we evaluate $w_{ij}(\tilde{\mathbf{x}}_p,t_n)$ as in sections 4.1–4.2. In the former case, we need to locate the head of the characteristic at the space-time boundary $\Gamma_n^{(O)}$ and calculate the values of w_{ij} at $\Gamma_n^{(O)}$ on which they are defined.

5. Description of some other numerical methods. In this section we briefly describe the Galerkin finite element method (GAL); the quadratic Petrov–Galerkin method [16, 5, 13]; the cubic Petrov–Galerkin method [74, 7]; and the streamline diffusion finite element method [44, 10, 45, 46, 38, 37, 47, 48, 49, 50, 51, 77, 78]. For simplicity of representation, we assume that in (2.1), $R(\mathbf{x},t) \equiv 1$. The GAL, QPG, and CPG schemes can be unified as follows:

$$\begin{aligned}
&\int_\Omega U_n\ w_{ij}\ d\mathbf{x} + \lambda\Delta t\left\{\int_\Omega \nabla w_{ij}\cdot(\mathbf{D}_n\nabla U_n)\ d\mathbf{x} - \int_\Omega \mathbf{V}_n U_n\cdot\nabla w_{ij}\ d\mathbf{x}\right\}\\
&\quad = \int_\Omega U_{n-1}\ w_{ij}\ d\mathbf{x} - (1-\lambda)\Delta t\left\{\int_\Omega \nabla w_{ij}\cdot(\mathbf{D}_{n-1}\nabla U_{n-1})\ d\mathbf{x}\right.\\
&\qquad\qquad \left. - \int_\Omega \mathbf{V}_n U_n\cdot\nabla w_{ij}\ d\mathbf{x}\right\} \qquad (5.1)
\end{aligned}$$

$$+\Delta t\left\{\lambda\int_\Omega f_n\ w_{ij}\ d\mathbf{x}+(1-\lambda)\int_\Omega f_{n-1}\ w_{ij}\ d\mathbf{x}\right\}+\text{ boundary terms.}$$

Here $\lambda\in[0,1]$ is the weighting parameter between the time levels t_{n-1} and t_n. In particular, $\lambda=1$ and 0.5 yield the backward-Euler (BE-) and the Crank–Nicholson (CN-) schemes, respectively. The trial function space consists of the standard continuous and piecewise-bilinear polynomials. The test functions are also in the tensor product form $w_{ij}(x,y)=w_i(x)w_j(y)$. In the GAL scheme, $w_i(x)$ and $w_j(y)$ are the standard one-dimensional hat functions. In the QPG, $w_i(x)$ and $w_j(y)$ are constructed by adding an asymmetric perturbation to the original piecewise-linear hat functions

$$w_i(x)=\begin{cases}\dfrac{x-x_{i-1}}{\Delta x}+\nu\dfrac{(x-x_{i-1})(x_i-x)}{(\Delta x)^2}, & x\in[x_{i-1},x_i],\\ \dfrac{x_{i+1}-x}{\Delta x}-\nu\dfrac{(x-x_i)(x_{i+1}-x)}{(\Delta x)^2}, & x\in[x_i,x_{i+1}],\\ 0 & \text{otherwise.}\end{cases}\tag{5.2}$$

Here $\nu=3[\coth(\frac{V\Delta x}{2D})-\frac{2D}{V\Delta x}]$ for constant V and D. For variable V and D, one replaces $\frac{V}{D}$ by its mean on each element. A typical one-dimensional QPG test function is sketched in Figure 1(b). As defined above, the two-dimensional QPG test function $w_{ij}(x,y)$ is just a tensor product of the two one-dimensional QPG test functions $w_i(x)$ and $w_j(y)$. The CPG method was derived for the Crank–Nicholson time discretization. In the CPG, $w_i(x)$ and $w_j(y)$ are defined as the original piecewise linear hat functions with a symmetric cubic perturbation added to each nonzero piece

$$w_i(x)=\begin{cases}\dfrac{x-x_{i-1}}{\Delta x}+\gamma\dfrac{(x-x_{i-1})(x_i-x)(x_{i-1}+x_i-2x)}{(\Delta x)^3}, & x\in[x_{i-1},x_i],\\ \dfrac{x_{i+1}-x}{\Delta x}-\gamma\dfrac{(x-x_i)(x_{i+1}-x)(x_i+x_{i+1}-2x)}{(\Delta x)^3}, & x\in[x_i,x_{i+1}],\\ 0 & \text{otherwise.}\end{cases}\tag{5.3}$$

Here $\gamma=5Cu^2$ with $Cu=\frac{V\Delta t}{\Delta x}$ being the Courant number. For variable V one replaces V by its arithmetic mean on each element. A typical one-dimensional CPG test function is plotted in Figure 1(c).

The SDM is a type of discontinuous Galerkin FEM and applies to a nonconservative analogue of (2.1). For the nonconservative advection-diffusion equation,

$$\begin{aligned}u_t+\mathbf{V}(\mathbf{x},t)\cdot\nabla u-\nabla(\mathbf{D}(\mathbf{x},t)\nabla u) &= f(\mathbf{x},t), && (\mathbf{x},t)\in\Sigma,\\ u(\mathbf{x},0) &= u_0(\mathbf{x}), && \mathbf{x}\in\Omega,\\ u(\mathbf{x},t) &= 0, && (\mathbf{x},t)\in\Gamma,\end{aligned}\tag{2.1$'$}$$

the trilinear SDM reads as follows: find a piecewise-trilinear (linear in time) function $U(\mathbf{x},t)$ on the space-time slab $\Sigma_n\equiv\Omega\times J_n$, which is discontinuous in time at t_{n-1} and t_n and satisfies the homogeneous Dirichlet boundary condition, such that

$$\begin{aligned}&\int_{\Sigma_n}\Big[U_t+\mathbf{V}\cdot\nabla U\Big]\Big[W+\delta(W_t+\mathbf{V}\cdot\nabla W)\Big]\ d\mathbf{x}dt+\int_{\Sigma_n}\nabla W\cdot(\mathbf{D}\nabla U)\ d\mathbf{x}dt\\ &\quad-\delta\int_{\Sigma_n}\nabla\cdot(\mathbf{D}\nabla U)(W_t+\mathbf{V}\cdot\nabla W)\ d\mathbf{x}dt+\int_\Omega U^+_{n-1}W^+_{n-1}\ d\mathbf{x}\\ &=\int_{\Sigma_n}f\Big[W+\delta(W_t+\mathbf{V}\cdot\nabla W)\Big]\ d\mathbf{x}dt+\int_\Omega U^-_{n-1}\ W^+_{n-1}\ d\mathbf{x}\end{aligned}\tag{5.4}$$

for any test function W with the same form as U. Here $W_{n-1}^{+} = \lim_{t \to t_{n-1}^{+}} w(\mathbf{x}, t)$ and $W_{n-1}^{-} = \lim_{t \to t_{n-1}^{-}} w(\mathbf{x}, t)$, $U_0^{-} = u_0(\mathbf{x})$, and δ is typically chosen to be of $O(h)$ with h being the diameter of the space-time partition on the slab Σ_n. The third term on the left-hand side is carried out elementwise, since it is not well defined for piecewise-trilinear functions.

The choice of δ has significant effects on the accuracy of the numerical solutions. If δ is chosen too small, the numerical solutions will exhibit oscillations. If δ is too big, the SDM will seriously damp the numerical solutions. Unfortunately, an optimal choice of δ is not clear and is heavily problem dependent. Extensive research has been conducted on the SDM, including proper choices of δ [44, 10, 45, 46, 38, 37, 47, 48, 49, 50, 51, 77, 78]. Since the theme of this paper is not on the development of the SDM, in our numerical experiments we use a generally accepted choice of δ which may not be best possible for a given problem. According to [49, 50, 77], we set

$$\delta = \frac{Kh}{\sqrt{1 + |\mathbf{V}|^2}} \tag{5.5}$$

if the mesh Peclet number $|\mathbf{V}|h > |\mathbf{D}|$, and $\delta = 0$ otherwise. K is typically to be 1 or 0.5. In our numerical experiments, we will use these values along with several others to indicate the general behavior. Moreover, the SDM generally increases the dimension of the problem by one (although the measure in this dimension is small). For problem (2.1), which is two dimensions in space, (5.4) are defined on three-dimensional space-time domain Σ_n. Numerically, one has to partition the three-dimensional "thick slices" into tetrahedra or prisms. Usually this will double the number of unknowns in GAL, QPG, CPG, and ELLAM schemes.

While the SDM can capture a jump discontinuity of the exact solution in a thin region, the numerical solution may develop over- and under-shoots about the exact solution within this layer. A modified SDM with improved shock-capturing properties was proposed [45, 49, 50] which consists of adding a "shock-capturing" term to the diffusion by introducing a crosswind control that is close to the steep fronts or "shocks." This modified SDM scheme performs much better in terms of catching the steep fronts or the jump discontinuities of the exact solutions; however, it leads to a nonlinear scheme even though the underlying governing partial differential equation is linear and involves another undetermined parameter. Thus, we will not use this scheme in our comparison and just remind the reader that in particular cases the SDM may perform better than those shown in the examples here if the appropriate modifications and optimization schemes are used.

6. Computational results. In this section we present one- and two-dimensional numerical experiments to investigate the performance of the ELLAM scheme developed in this paper and to compare it with the numerical methods described in section 5. The numerical experiments contain both examples (with analytical solutions) that are either smooth or have steep fronts.

6.1. The one-dimensional transport of a diffused square wave. To observe the performance of all the methods in section 5 and ELLAM scheme for problems with analytical solutions that have a steep front, this example considers the transport of a one-dimensional diffused square wave. The initial condition $u_0(x)$ is given by

$$u_0(x) = \begin{cases} 1 & \text{if } x \in [x_l, x_r] \subset (a, b), \\ 0 & \text{otherwise.} \end{cases} \tag{6.1}$$

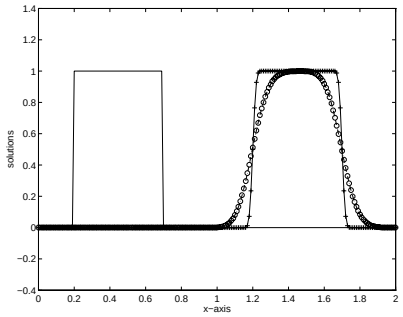
We assume that the one-dimensional transport equation has constant coefficients so that we can find the analytical solution in a closed form. Homogeneous inflow and outflow Dirichlet boundary conditions are specified at $x = a$ and $x = b$. As long as the diffused square wave does not intersect the outflow boundary during the time interval $[0, T]$, the analytical solution $u(x, t)$ can be expressed as

$$
\begin{aligned}
u(x,t) &= \frac{1}{\sqrt{4\pi Dt}} \int_{-\infty}^{\infty} u_0(x - Vt - s) \exp\left(\frac{-s^2}{4Dt}\right) ds \\
&= \frac{1}{2}\left[erf\left(\frac{x - Vt - x_l}{\sqrt{4Dt}}\right) - erf\left(\frac{x - Vt - x_r}{\sqrt{4Dt}}\right)\right],
\end{aligned} \tag{6.2}
$$

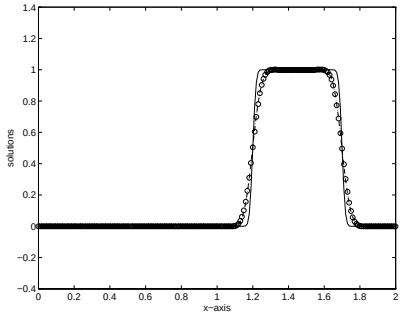
where $erf(x) = \frac{2}{\sqrt{\pi}} \int_0^x \exp(-s^2) ds$ is the error function.

In the numerical experiments the data are chosen as follows: The space domain is $(a, b) = (0, 2)$, the time interval $[0, T] = [0, 1]$, $R = 1$, $V = 1$, $D = 10^{-4}$, and $f = 0$. In (6.1) and (6.2), $x_l = 0.2$ and $x_r = 0.7$, so that the diffused square wave essentially vanishes at the outflow boundary $x = b$ during the time period of $[0, T]$. The grid size $\Delta x = \frac{1}{100}$ is chosen so that the analytical solution can be represented properly. The backward Euler–Galerkin linear finite element (BE-GAL), quadratic Petrov–Galerkin finite element (BE-QPG), and cubic Petrov–Galerkin finite element (BE-CPG) solutions are plotted against the analytical solution in Figures 2(a)–(c) for $\Delta t = \frac{1}{200}$, $\frac{1}{800}$, and $\frac{1}{2000}$, respectively. The ELLAM solution is also plotted in Figure 2(a) for $\Delta t = \frac{1}{10}$, which gives a Courant number 10 and a Peclet number 100. The Crank–Nicholson–Galerkin linear finite element (CN-GAL), quadratic Petrov–Galerkin finite element (CN-QPG), and cubic Petrov–Galerkin finite element (CN-CPG) solutions are plotted in Figures 3(a)–(c) for $\Delta t = \frac{1}{100}$, $\frac{1}{200}$, and $\frac{1}{1000}$, respectively. To view the numerical solutions clearly, we do not plot the analytical solution in these figures. One can compare the CN-GAL, CN-QPG, and CN-CPG numerical solutions with the analytical ones in Figures 2(a)–(c). The SDM solutions are plotted in Figures 4(a)–(b) for $\Delta t = \frac{1}{20}$ and $\Delta t = \frac{1}{100}$. The SDM solution is also plotted in Figure 4(c) for $\Delta x = \Delta t = \frac{1}{50}$ to further observe the effect of the choice of the parameter δ.

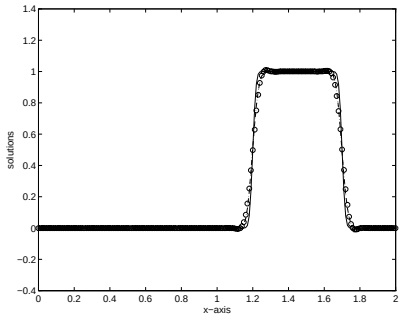
It is observed that the BE-GAL, BE-QPG, and BE-CPG schemes generate almost identical numerical solutions. With a time step of $\Delta t = \frac{1}{200}$, the backward Euler schemes generate over-damped numerical solutions without any observable overshoot or undershoot. As the time step Δt is reduced to $\frac{1}{800}$ and $\frac{1}{2000}$, the numerical diffusion is reduced considerably and the numerical solutions are quite close to the analytical solution. With a time step of $\Delta t = \frac{1}{100}$, the CN-GAL and CN-QPG solutions display overshoot and undershoot. The maximum and minimum values of CN-GAL and CN-QPG solutions are 1.212, 1.153, -0.219, and -0.153, respectively. The CN-CPG solution also has many wiggles but with a much smaller magnitude. (Its maximum and minimum values are 1.035 and -0.031.) As the time step Δt is reduced to $\frac{1}{200}$, the undershoot and overshoot of the CN-GAL solution are reduced by about 40%. (The maximum and minimum values are 1.132 and -0.134.) The undershoot and overshoot of CN-QPG solution are reduced by 70% (the maximum and minimum values are 1.047 and -0.047) and are comparable to those of CN-CPG solution (whose maximum and minimum values are 1.033 and -0.033). As the time step Δt decreases to $\frac{1}{1000}$, the undershoot and overshoot of CN-GAL solution are further reduced, but those of CN-QPG and CN-CPG solutions do not change much. In essence, when the time step is relatively large (the Courant number is up to one), the CN-CPG scheme yields better solutions than the CN-GAL and CN-QPG schemes.



(a) — for init. and anal.; + for ELLAM ($\Delta t = \frac{1}{10}$); - -, o, and ... for GAL, QPG, and CPG ($\Delta t = \frac{1}{200}$).



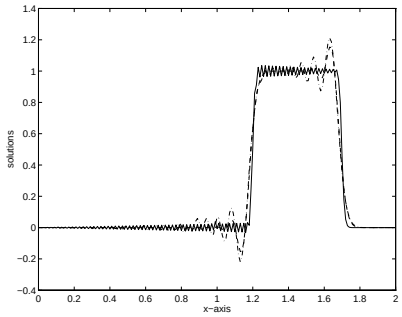
(b) — for anal.; - -, o, and ... for GAL, QPG, and CPG with $\Delta t = \frac{1}{800}$.



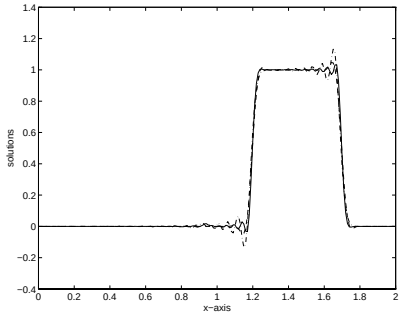
(c) — for anal.; - -, o, and ... for GAL, QPG, and CPG with $\Delta t = \frac{1}{2000}$.

FIG. 2. *ELLAM and backward Euler (BE-) GAL, QPG, and CPG methods,* $\Delta x = \frac{1}{100}$.

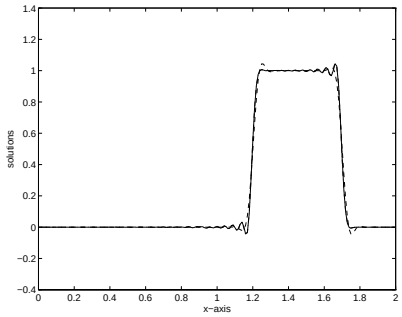
With a time step $\Delta t = \frac{1}{20}$, the SDM solution starts to approximate the analytical solution. As the K in δ decreases from 1 to 0.001, the smearing in the numerical solutions is reduced considerably and the overshoot/undershoot is also reduced slightly (from 1.0952 and -0.0577 to 1.0714 and -0.0721). Thus, the optimal value of K seems to be 0.001 (the smallest of the three K values). As the time step Δt decreases to $\frac{1}{100}$, the SDM solutions become more accurate and have much less damping. But in this



(a) -., - -, and — for GAL, QPG, and CPG with $\Delta t = \frac{1}{100}$.



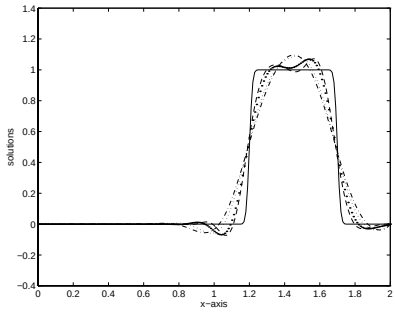
(b) -., - -, and — for GAL, QPG, and CPG with $\Delta t = \frac{1}{200}$.



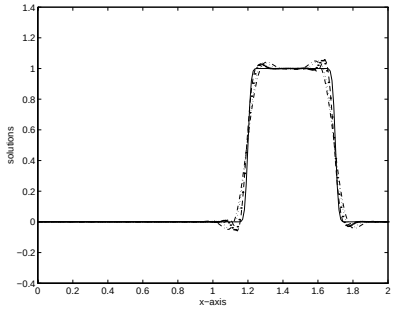
(c) -., - -, and — for GAL, QPG, and CPG with $\Delta t = \frac{1}{1000}$.

FIG. 3. *Crank–Nicholson (CN-) GAL, QPG, and CPG methods,* $\Delta x = \frac{1}{100}$.

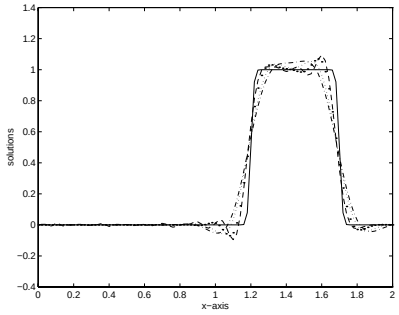
case some wiggles appear near the locations where the analytical solution has steep fronts. Reducing K in δ from 1 to 0.001 gives a smaller L^2 error in the SDM solution, but increases the overshoot/undershoot slightly (from 1.0493 and -0.0493 to 1.0592 and -0.0592). In Figure 4(c), the magnitude of the overshoot and undershoot in SDM solutions is almost doubled (from 1.0520 and -0.0548 to 1.0932 and -0.0952) when K is reduced from 1 to 0.001. Unfortunately, there is no universal rule on the choice of the K. As we mentioned above, the modified SDM with a shock capturing property



(a) — for anal., -., ..., and · for SDM with $K = 1, 0, 5$, and 0.001, $\Delta x = \frac{1}{100}$, $\Delta t = \frac{1}{20}$.



(b) — for anal., -., ..., and · for SDM with $K = 1, 0, 5$, and 0.001, $\Delta x = \frac{1}{100}$, $\Delta t = \frac{1}{100}$.



(c) — for anal., -., ..., and · for SDM with $K = 1, 0, 5$, and 0.001, $\Delta x = \frac{1}{50}$, $\Delta t = \frac{1}{50}$.

FIG. 4. *Streamline diffusion finite element method.*

should generate better numerical solutions than those shown here. However, one has to solve a nonlinear system even though the underlying partial differential equation is a linear one and must face choosing an additional undetermined parameter. In contrast, with a fairly large time step $\Delta t = \frac{1}{10}$, the ELLAM scheme yields a very accurate numerical solution that is better than any one of the GAL, QPG, CPG, or

SDM solutions with even much finer time steps. Moreover, the ELLAM scheme uses only half the number of unknowns as in the SDM and does not require optimization of indefinite constants.

6.2. A two-dimensional rotating Gaussian pulse. This example considers the transport of a two-dimensional rotating Gaussian pulse. The spatial domain is $\Omega = (-0.5, 0.5) \times (-0.5, 0.5)$, the rotating field is imposed as $V_1(x, y) = -4y$, and $V_2(x, y) = 4x$. The time interval is $[0, T] = [0, \pi/2]$, which is the time period required for one complete rotation. The initial condition $u_0(x, y)$ is given by

$$u_0(x, y) = \exp\left(-\frac{(x - x_c)^2 + (y - y_c)^2}{2\sigma^2}\right), \tag{6.3}$$

where x_c, y_c, and σ are the centered and standard deviations, respectively. The corresponding analytical solution for (2.1) with $R = 1$, a constant diffusion coefficient D, and $f = 0$ is given by

$$u(x, y, t) = \frac{2\sigma^2}{2\sigma^2 + 4Dt} \exp\left(-\frac{(\bar{x} - x_c)^2 + (\bar{y} - y_c)^2}{2\sigma^2 + 4Dt}\right), \tag{6.4}$$

where $\bar{x} = x\cos(4t) + y\sin(4t)$ and $\bar{y} = -x\sin(4t) + y\cos(4t)$.

In the numerical experiments, the data are chosen as follows: $D = 10^{-4}$, $x_c = -0.25$, $y_c = 0$, $\sigma = 0.0447$, which gives $2\sigma^2 = 0.0040$. A uniform spatial grid of $\Delta x = \Delta y = \frac{1}{64}$ is used (in all the plots and most of the experiments in Table 1), unless it is specified otherwise, in which case a uniform spatial grid of $\Delta x = \Delta y = \frac{1}{96}$ is used. It is easy to see that the $u_0(x, y)$ defined by (6.3) is centered at (x_c, y_c) with a minimum value 0 and a maximum value 1. Its surface and contour plots are presented in Figures 5(a) and (b). Figures 5(c) and (d) are the surface and contour plots for the analytical solution, which has a minimum value 0 and a maximum value 0.8642 (due to the effect of diffusion).

This problem provides an example for a homogeneous two-dimensional advection-diffusion equation with a variable velocity field and a known analytical solution. Moreover, this problem changes from the advection dominance in most of the domain to the diffusion dominance in the region that is close to the origin. These types of problems often arise in many important applications and are more difficult to simulate compared with purely advection-dominated problems. This example has been used widely to test for numerical artifacts of different schemes, such as numerical stability and numerical diffusion, spurious oscillations, and phase errors.

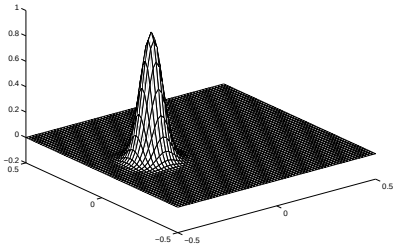
In our experiments, we have systematically varied the time steps to examine the performance of each method, using a uniform spatial grid of $\Delta x = \Delta y = \frac{1}{64}$ in most experiments. This is because the temporal errors dominate the numerical solutions with all the methods other than the ELLAM schemes. In this case the maximum grid Peclet number reaches 442. The grid Peclet number at the center of the Gaussian pulse is about 156. We also perform some experiments with a finer spatial grid of $\Delta x = \Delta y = \frac{1}{96}$. In this case the maximum grid Peclet number reaches 295, and the grid Peclet number at the center of the Gaussian pulse is about 104. All comparative methods tested yield strongly nonsymmetric systems, while the ELLAM scheme inherently symmetrizes its discrete algebraic system. We use a preconditioned conjugate gradient square algorithm (PCGS) to solve these systems, even though this places ELLAM at a disadvantage. In Table 1 we present the minimum and maximum values of the numerical solutions with each method and the per time step CPU

TABLE 1
Comparison of CPU time for different methods.

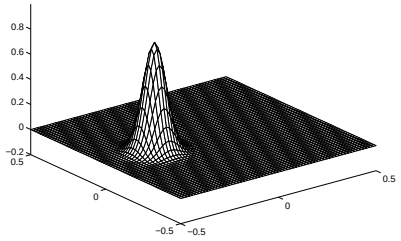
	Δt	$\Delta x = \Delta y$	Max.	Min.	CPU (sec.) /time step	Overall CPU	Fig. #
Analytical	N/A	$\frac{1}{64}$	0.8642	0	N/A	N/A	5(c)–(d)
ELLAM-Er	$\frac{\pi}{40}$	$\frac{1}{64}$	0.8302	0	17	5 min. 39 sec.	6(a)–(b)
ELLAM-RK	$\frac{\pi}{30}$	$\frac{1}{64}$	0.8630	0	10.2	2 min. 33 sec.	6(c)–(d)
BE-GAL	$\frac{\pi}{200}$	$\frac{1}{64}$	0.2546	0	15.4	25 min. 40 sec.	-
	$\frac{\pi}{800}$	$\frac{1}{64}$	0.4517	0	13.85	1 hr. $32\frac{1}{3}$ min.	7(a)–(b)
	$\frac{\pi}{800}$	$\frac{1}{96}$	0.4523	0	31.75	3 hr. $31\frac{2}{3}$ min.	-
	$\frac{\pi}{2000}$	$\frac{1}{64}$	0.6006	0	13.65	3 hr. $47\frac{1}{2}$ min.	-
	$\frac{\pi}{2000}$	$\frac{1}{96}$	0.6015	0	30.85	8 hr. $34\frac{1}{6}$ min.	-
	$\frac{\pi}{4000}$	$\frac{1}{64}$	0.6964	0	13.6	7 hr. $33\frac{1}{3}$ min.	-
	$\frac{\pi}{6000}$	$\frac{1}{64}$	0.7401	0	13.55	11 hr. $17\frac{1}{2}$ min.	8(a)–(b)
BE-QPG	$\frac{\pi}{200}$	$\frac{1}{64}$	0.2003	-0.0002	17.75	29 min. 35 sec.	-
	$\frac{\pi}{800}$	$\frac{1}{64}$	0.3486	-0.0003	17.9	1 hr. $59\frac{1}{3}$ min.	7(c)–(d)
	$\frac{\pi}{800}$	$\frac{1}{96}$	0.3842	0	39.5	4 hr. $23\frac{1}{3}$ min.	-
	$\frac{\pi}{2000}$	$\frac{1}{64}$	0.4563	-0.0008	18.1	5 hr. $1\frac{2}{3}$ min.	-
	$\frac{\pi}{2000}$	$\frac{1}{96}$	0.5035	-0.0003	40.1	11 hr. $8\frac{1}{3}$ min.	-
	$\frac{\pi}{4000}$	$\frac{1}{64}$	0.5250	-0.0015	18.25	10 hr. $8\frac{1}{3}$ min.	-
	$\frac{\pi}{6000}$	$\frac{1}{64}$	0.5554	-0.0017	18.45	15 hr. $22\frac{1}{2}$ min.	8(c)–(d)
CN-GAL	$\frac{\pi}{200}$	$\frac{1}{64}$	0.7861	-0.1564	15.05	25 min. 5 sec.	9(a)–(b)
	$\frac{\pi}{400}$	$\frac{1}{64}$	0.8438	-0.0159	13.95	$46\frac{1}{2}$ min.	10(a)–(b)
CN-QPG	$\frac{\pi}{200}$	$\frac{1}{64}$	0.6197	-0.0978	17.95	29 min. 55 sec.	9(c)–(d)
	$\frac{\pi}{400}$	$\frac{1}{64}$	0.6412	-0.0081	18	1 hr.	10(c)–(d)
CN-CPG	$\frac{\pi}{200}$	$\frac{1}{64}$	N/A	N/A	N/A	N/A	-
	$\frac{\pi}{400}$	$\frac{1}{64}$	0.8555	-0.0002	20	1 hr. $6\frac{2}{3}$ min.	11(a)–(b)
SDM							
K=0.5	$\frac{\pi}{200}$	$\frac{1}{64}$	0.7089	-0.0147	96.9	2 hr. 41.5 min.	11(c)–(d)
K=0.01	$\frac{\pi}{200}$	$\frac{1}{64}$	0.8250	-0.0021	96.9	2 hr. 41.5 min.	12(a)–(b)
K=0.001	$\frac{\pi}{200}$	$\frac{1}{64}$	0.8281	-0.0019	96.9	2 hr. 41.5 min.	12(c)–(d)

and the overall CPU each method consumed, which was measured on a SGI Indigo Workstation. We realize, of course, that some code optimization may be possible, but feel that these timings are representative of each scheme's efficiency on these model problems. The surface and contour plots for selected runs of each method in Table 1 are presented in Figures 5–12.

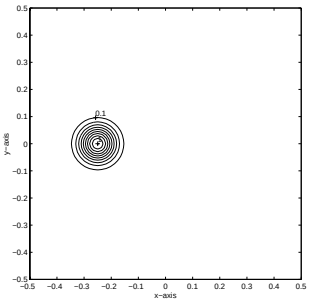
6.2.1. The ELLAM simulation. The ELLAM-Er solution is obtained by using a time step of $\Delta t = \frac{\pi}{40}$ in solving the ELLAM scheme and using an Euler quadrature with a micro-time step of $\Delta t_f = \frac{\Delta t}{80}$ in tracking the characteristics. The maximum



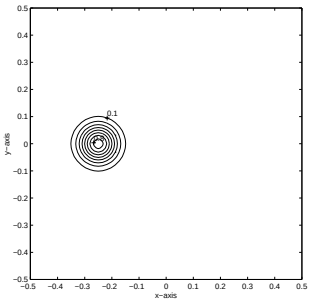
(a) initial condition, min = 0, max = 1



(c) analytical, min = 0, max = 0.8642



(b) initial condition

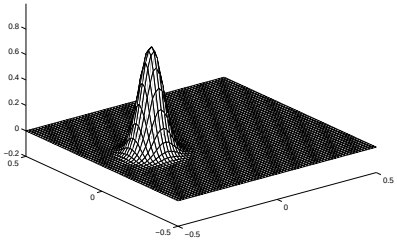


(d) analytical solution

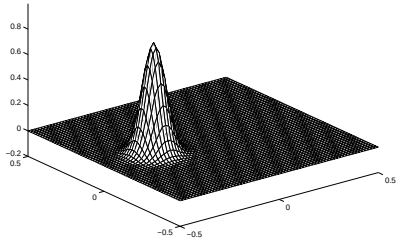
FIG. 5. *The initial condition at* $t = 0$, *and the analytical solution at* $T = \frac{\pi}{2}$, $\Delta x = \Delta y = \frac{1}{64}$.

Courant number reaches 14, while the Courant number at the center of the Gaussian pulse is 5. The ELLAM-Er solution has a minimum value 0 and a maximum value 0.8302, whose surface and contour plots are given in Figures 6(a)–(b). As one can see from these plots and Table 1, a very accurate numerical solution is obtained in about five and a half minutes. The ELLAM-RK solution is obtained by using a time step of $\Delta t = \frac{\pi}{30}$ in solving the ELLAM scheme and using a second-order Runge–Kutta quadrature with a microtime step of $\Delta t_f = \frac{\Delta t}{4}$ in tracking the characteristics. In this case, the maximum Courant number reaches 19, while the Courant number at the center of the Gaussian pulse is about 6.7. The ELLAM-RK solution has a minimum value 0 and a maximum value 0.8630, whose plots are in Figures 6(c)–(d). The use of a more accurate second-order Runge–Kutta tracking algorithm enables us to significantly reduce the number of microtime steps (from 80 in an Euler tracking to 4 in the Runge–Kutta tracking) in tracking characteristics, and so the CPU time per global time step (from 17 seconds in ELLAM-Er to 10.2 seconds in ELLAM-RK). Moreover, the number of global time steps is reduced from 20 in the ELLAM-Er simulation to 15 in the ELLAM-RK simulation. Thus, the ELLAM-RK simulation further reduces the overall CPU time to 2 minutes and 33 seconds.

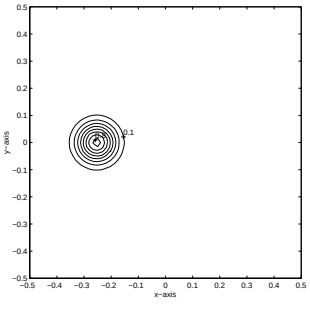
6.2.2. The BE-GAL and BE-QPG simulation. Due to its unconditional stability and simplicity in implementations, the fully implicit backward Euler temporal discretization still dominates most production codes in engineering applications. Thus, we carry out extensive experiments to investigate the performance of this discretization. With a time step of $\Delta t = \frac{\pi}{200}$, which gives a maximum Courant number of 2.84 and a Courant number of 1 at the center of the Gaussian pulse, the BE-GAL and



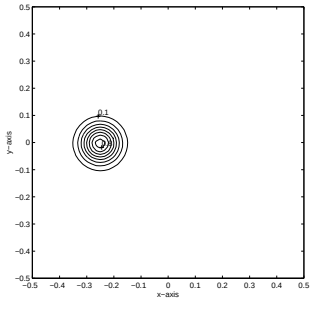
(a) ELLAM-Er, min = 0.0000, max = 0.8630



(c) ELLAM-RK, min = 0.0000, max = 0.8302



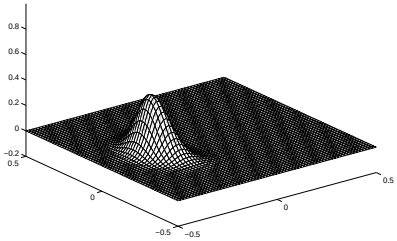
(b) ELLAM-Er, $\Delta t = \frac{\pi}{40}$



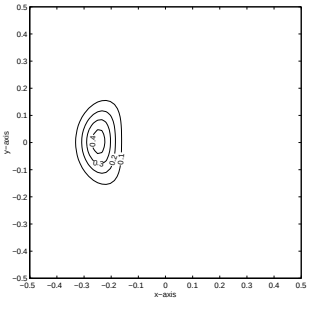
(d)ELLAM-RK, $\Delta t = \frac{\pi}{30}$

FIG. 6. *The ELLAM-Euler and ELLAM-RK solutions at* $T = \frac{\pi}{2}$, $\Delta x = \Delta y = \frac{1}{64}$.

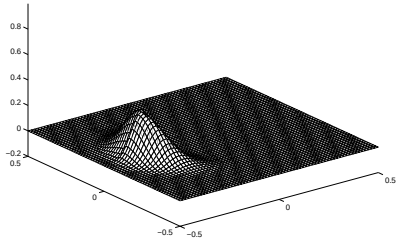
BE-QPG solutions have minimum values of 0 and -0.0002 and maximum values of 0.2546 and 0.2003, respectively. Recall that the exact solution has a maximum value of 0.8642; the BE-GAL and BE-QPG solutions are excessively overdamped. Moreover, the BE-GAL and BE-QPG schemes require more iterations in the PCGS solver than the ELLAM does, because they yield strongly nonsymmetric coefficient matrices. The BE-GAL and BE-QPG solutions with a much finer time step of $\Delta t = \frac{\pi}{800}$ are presented in Figure 7(a)–(d). The minimum values are 0 for the BE-GAL solution and -0.0003 for the BE-QPG solution, while the maximum values are 0.4517 for the BE-GAL solution and 0.3486 for BE-QPG solution. These solutions are still very diffusive and are considerably deformed, especially for the BE-QPG solution. The more severe deformation in the BE-QPG solution is due to the effect of grid orientation incurred by the upwinding in the QPG method (see [26]). With the same time step of $\Delta t = \frac{\pi}{800}$, we also reduce the spatial grid from $\Delta x = \Delta y = \frac{1}{64}$ to $\Delta x = \Delta y = \frac{1}{96}$ to observe the improvement of the numerical solutions. The BE-QPG solution has a slight improvement, while the BE-GAL has essentially no improvement. However, the CPU time has been significantly increased. With a comparable overall CPU time we could use a much finer time step of $\Delta t = \frac{\pi}{2000}$ and still use the coarse spatial grid of $\Delta x = \Delta y = \frac{1}{64}$. In this case the numerical solutions have more visible improvement. This shows that even with a time step of $\Delta t = \frac{\pi}{800}$ and a spatial grid of $\Delta x = \Delta y = \frac{1}{64}$, the temporal dominance still dominates the numerical solutions in the backward temporal discretization. Note that for $\Delta x = \Delta y = \frac{1}{64}$ and $\Delta t = \frac{\pi}{800}$, the corresponding maximum Courant number and the Courant number at the center of the Gaussian pulse are 0.7 and 0.25, respectively.



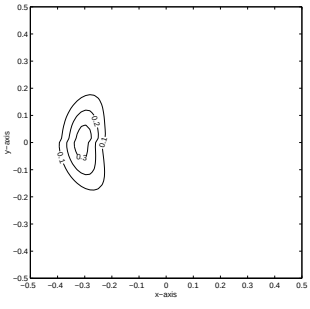
(a) BE-GAL, min = 0.0000, max = 0.4517



(b) BE-GAL



(c) BE-QPG, min = −0.0003, max = 0.3486

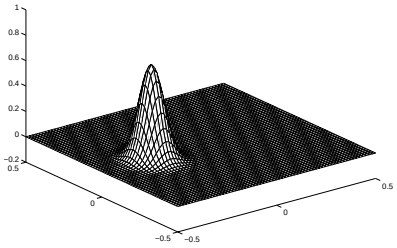


(d) BE-QPG

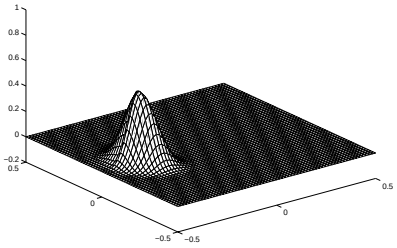
FIG. 7. *The BE-GAL and BE-QPG solutions at* $T = \frac{\pi}{2}, \Delta x = \Delta y = \frac{1}{64}$, *and* $\Delta t = \frac{\pi}{800}$.

To obtain BE-GAL and BE-QPG solutions with reasonable accuracy, we proceed further. With a time step of $\Delta t = \frac{\pi}{2000}$, we reduce the spatial grid from $\Delta x = \Delta y = \frac{1}{64}$ to $\Delta x = \Delta y = \frac{1}{96}$. Again one sees slight improvement in the BE-QPG solution and no improvement in the BE-GAL solution, but a significant increase in the overall CPU time. Using less overall CPU time, we could use the original coarse spatial grid of $\Delta x = \Delta y = \frac{1}{64}$, but a finer time step of $\Delta t = \frac{\pi}{4000}$. Note that for $\Delta x = \Delta y = \frac{1}{64}$ and $\Delta t = \frac{\pi}{2000}$, the corresponding maximum Courant number and the Courant number at the center of the Gaussian pulse are 0.28 and 0.1, respectively. This shows that the temporal error still dominates the BE-GAL and BE-QPG solutions. Our last numerical experiments with the backward Euler temporal discretization used a spatial grid of $\Delta x = \Delta y = \frac{1}{64}$ and a time step of $\Delta t = \frac{\pi}{6000}$. The minimum values are 0 for BE-GAL solution and −0.0017 for BE-QPG solution, while the maximum values are 0.7401 for BE-GAL solution and 0.5554 for BE-QPG solution. Their surface and contour plots are presented in Figure 8(a)–(d). Note that in this case the maximum Courant number and the Courant number at the center of the Gaussian pulse are 0.09 and 0.03, respectively. However, the BE-GAL solution is still not comparable with the two ELLAM solutions. The BE-QPG solution is even much worse. In fact, Figure 8(c) and (d) show that the BE-QPG solution has severe deformation. However, the overall CPU time is more than 11 hours for the BE-GAL solution and more than 15 hours for the BE-QPG solution. This is in contrast to the 4 minutes and 39 seconds for the ELLAM-Er solution and 2 minutes 33 seconds for the ELLAM-RK solution.

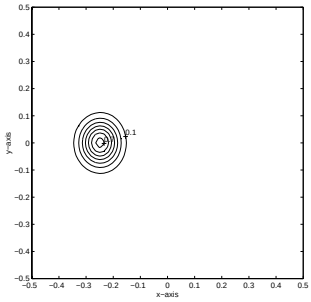
Therefore, even though the backward Euler temporal discretization is unconditionally stable and simple to implement, extremely small time steps have to be used



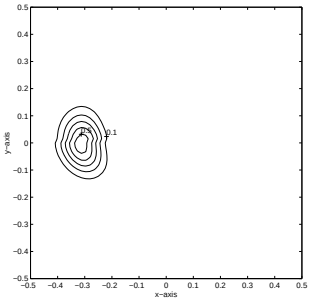
(a) BE-GAL, min = 0.0000, max = 0.7401



(c) BE-QPG, min = −0.0017, max = 0.5554



(b) BE-GAL



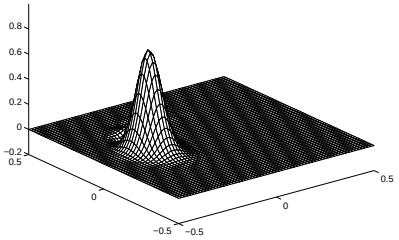
(d) BE-QPG

FIG. 8. *The BE-GAL and BE-QPG solutions at* $T = \frac{\pi}{2}$, $\Delta x = \Delta y = \frac{1}{64}$, *and* $\Delta t = \frac{\pi}{6000}$.

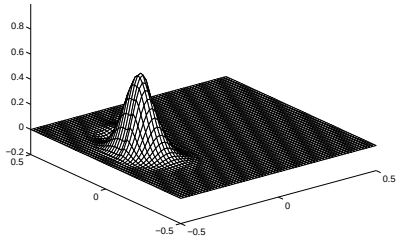
in these schemes, not for the purpose of stability, but for the purpose of comparative accuracy (see [26]). Consequently, this significantly reduces the efficiency of the simulation.

6.2.3. The CN-GAL, CN-QPG, and CN-CPG simulation. The CN-GAL and CN-QPG solutions are presented in Figure 9(a)–(d) for a spatial grid of $\Delta x = \Delta y = \frac{1}{64}$ and a time step of $\Delta t = \frac{\pi}{200}$. The CN-GAL solution has minimum and maximum values −0.1564 and 0.7861. Severe undershoot, deformation, and phase errors are observed in the CN-GAL solution in Figure 9(a) and (b). The CN-QPG solution has a minimum value −0.0978 and a maximum value 0.6197, respectively. Hence, the CN-QPG solution has about 40% less undershoot than the CN-GAL solution, but it also has serious damping, phase error, and deformation. The CN-CPG solution is not available (unbounded) for the time step and spatial grid. Note that the maximum Courant number is 2.84 and the Courant number is 1 at the center of the Gaussian pulse in the current circumstances. Also, note that the Crank–Nicholson temporal discretization yields more accurate numerical solutions than the backward Euler temporal discretization due to its higher-order temporal accuracy. The overall CPU time in this case is about 25 minutes for the CN-GAL solution and about 30 minutes for the CN-QPG solution. The BE-GAL and BE-QPG solutions do not have undershoot, but the CN-GAL and CN-QPG solutions do exhibit serious problems in this regard and indicate a considerable disadvantage for the Crank–Nicholson discretization.

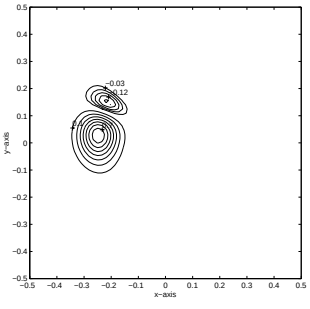
We further reduce the time step to $\Delta t = \frac{\pi}{400}$ in the numerical simulation. In this case, the CN-CPG solution is also available. We present the results in Table 1 and



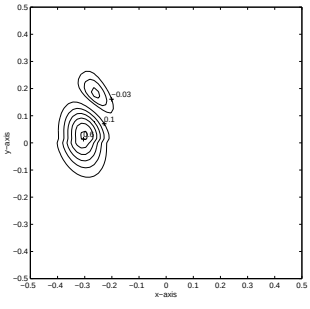
(a) CN-GAL, min = −0.1564, max = 0.7861



(c) CN-QPG, min = −0.0978, max = 0.6197



(b) CN-GAL

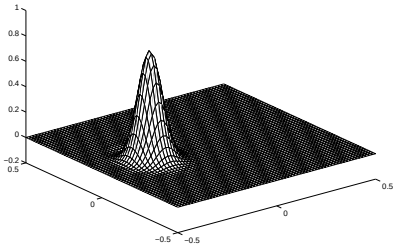


(d) CN-QPG

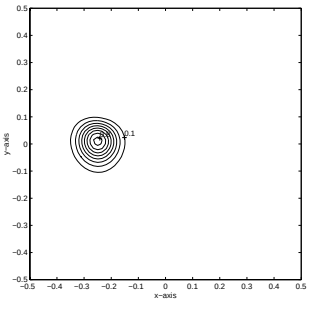
FIG. 9. *The CN-GAL and CN-QPG solutions at* $T = \frac{\pi}{2}$, $\Delta x = \Delta y = \frac{1}{64}$, *and* $\Delta t = \frac{\pi}{200}$.

Figures 10(a)–(d) and 11(a) and (b). The maximum values are 0.8438 for the CN-GAL solution, 0.6412 for the CN-QPG solution, and 0.8555 for the CN-CPG solution, while the minimum values are −0.0159 for the CN-GAL solution, −0.0081 for the CN-QPG solution, and −0.0002 for the CN-CPG solution. As one can see, the numerical solutions have been improved considerably and the undershoot in the solutions has been rapidly reduced. However, these solutions still have deformation, especially in the case of the CN-QPG solution.

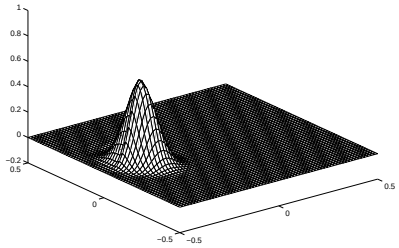
6.2.4. The SDM simulation. The surface and contour plots of SDM solutions are plotted in Figures 11(c) and (d) and 12(a)–(d) for a time step of $\Delta t = \frac{\pi}{200}$ and $\Delta x = \Delta y = \frac{1}{64}$. The undetermined parameter K in (5.5) equals 0.5, 0.1, and 0.001, respectively. As K decreases from 0.5 to 0.1 and then to 0.001, the maximum and minimum values of the corresponding SDM solutions change from 0.7089 and −0.0147 to 0.8250 and −0.0021 and then to 0.8281 and −0.0019. Namely, the SDM solutions have eliminated almost all the damping and undershoot and become more accurate. The numerical solutions will no longer improve as one further reduces the value of K. The SDM solutions have no phase error or deformation but do require the most CPU time per time step since they have double the number of unknowns as those for the other methods. This in turn requires more iterations in solving the linear system. Furthermore, on each (space-time) cell, the SDM has eight basis functions, which are the tensor product of three univariate functions, while all other methods have four basis functions on each (space) cell, which are the tensor product of two univariate functions.



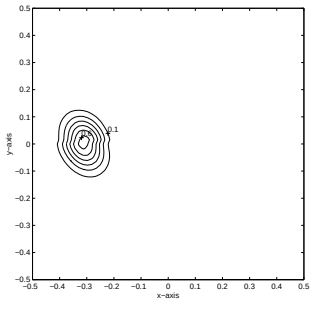
(a) CN-GAL, min = −0.0159, max = 0.8438



(b) CN-GAL



(c) CN-QPG, min = −0.0081, max = 0.6412



(d) CN-QPG

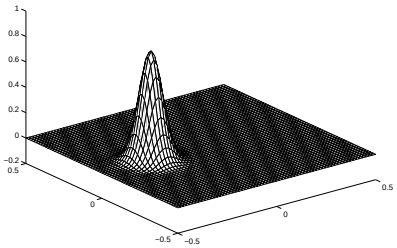
FIG. 10. *The CN-GAL and CN-QPG solutions at* $T = \frac{\pi}{2}$, $\Delta x = \Delta y = \frac{1}{64}$, *and* $\Delta t = \frac{\pi}{400}$.

In summary, one sees from Table 1 and Figures 5–12 that the ELLAM scheme is the most (CPU) cost effective per time step and is much more cost effective over all, since the ELLAM scheme outperforms the other methods tested with much fewer time steps.

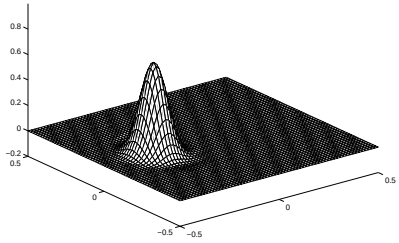
We refer readers to [29, 30, 34, 67, 69, 70] for the asymptotic convergence rates of the ELLAM schemes in space and time, where numerical experiments were performed and theoretical convergence error estimates were derived for the asymptotic convergence rates of the ELLAM schemes for first-order linear hyperbolic equations and advection-diffusion equations in one space dimension. A theoretical error estimate for an ELLAM scheme for a two-dimensional advection-diffusion equation with constant coefficients was also outlined in Wang [67].

7. Summary. In this paper we have developed an Eulerian–Lagrangian localized adjoint method for two-dimensional linear advection-diffusion equations with general inflow and outflow boundary conditions based on the approach presented in Wang [67]. The derived numerical scheme conserves mass and treats any combinations of inflow and outflow Dirichlet, Neumann, and Robin boundary conditions in a systematic manner.

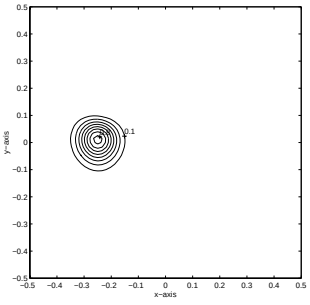
Traditional forward-tracked characteristic methods or particle methods advance the grids following the characteristics, which typically result in severely distorting the evolving grids, even though the initial grids were uniform. This greatly complicates the solution procedures. Many characteristic methods including certain ELLAM schemes have been developed using a backtracking algorithm to avoid these problems [1, 2,



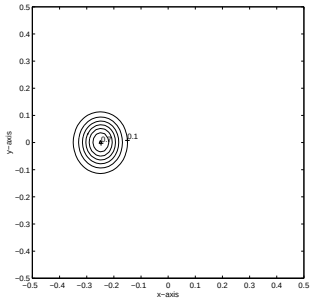
(a) CN-CPG, min $= -0.0002$, max $= 0.8555$



(c) SDM, min $= -0.0147$, max $= 0.7089$



(b) CN-GPG, $\Delta t = \frac{\pi}{400}$



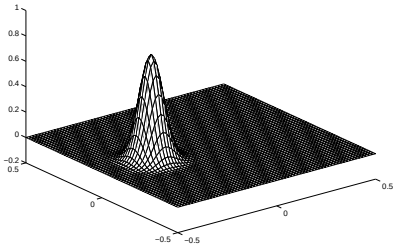
(d) SDM, $K = 0.5$, $\Delta t = \frac{\pi}{200}$

FIG. 11. *The CN-CPG and SDM solutions at* $T = \frac{\pi}{2}$, $\Delta x = \Delta y = \frac{1}{64}$.

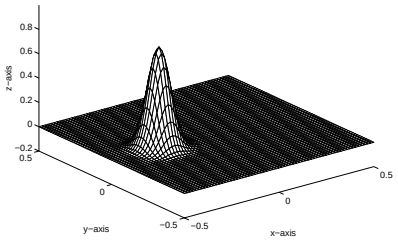
7, 8, 14, 19, 23, 25, 26, 29, 30, 31, 33, 34, 60, 71]. However, for multidimensional problems, backtracked characteristic methods require significant effort due to the need to define the geometry at time t_{n-1}, which requires the tracking of points along the boundary of the element and subsequent interpolation and mapping onto the fixed spatial grid at the previous time level t_{n-1}. This approach is computationally very intensive, especially when part or all of the element being mapped intersects a space-time boundary [8, 11].

The ELLAM scheme in this paper uses a forward-tracking approach [20, 40, 61, 67] to track quadrature points at time t_{n-1} in evaluating the storage terms and inflow boundary terms on the right-hand side of (3.4), (3.5), (3.6), and (3.8). Thus, this forward-tracking scheme has no effect on the underlying grid or the data structure of the discrete system. Furthermore, the scheme uses a backtracking of characteristics to evaluate the $\mathbf{x}$-dependent time step $\Delta t^{(I)}(\mathbf{x})$ in the diffusion-dispersion term.

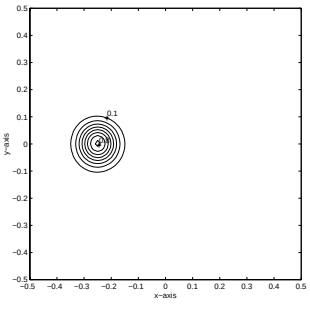
In this paper we have performed one- and two-dimensional numerical experiments to observe the performance of the ELLAM scheme and to compare it with many intensely investigated and well-received methods such as the standard Galerkin finite element method, quadratic Petrov–Galerkin finite element method, and cubic Petrov–Galerkin method, which use either a backward Euler or a Crank–Nicholson temporal discretization, as well as the streamline diffusion finite element method. The numerical experiments show that the ELLAM scheme has generated very accurate numerical solutions (compared with the other methods considered), even though a much larger time step is used in the ELLAM scheme. Consequently, the ELLAM scheme has a significantly enhanced efficiency. In the context of one-dimensional first-order linear



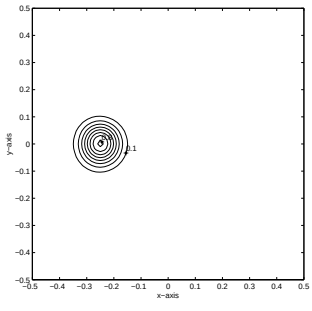
(a)$K = 0.01$, min $= -0.0021$, max $= 0.8250$



(c) $K = 0.001$, min $= -0.0019$, max $= 0.8281$



(b) $K = 0.01$



(d) $K = 0.001$

FIG. 12. *TThe SDM solutions at* $T = \frac{\pi}{2}$, $\Delta x = \Delta y = \frac{1}{64}$, *and* $\Delta t = \frac{\pi}{200}$.

hyperbolic equations, a more extensive comparison of the ELLAM schemes with many other well-regarded methods, including the continuous and discontinuous Galerkin finite element method [35, 49, 48, 59], the monotonic upstream-centered scheme for conservation laws [17, 65], and the essentially nonoscillatory scheme [21, 39, 63], as well as the methods described in section 5, can be found in the work of Wang, Al-Lawatia, and Telyakovskiy [73].

Finally, we point out that the Eulerian methods are relatively easy to formulate and to implement, in general. In contrast, due to the use of the Lagrangian coordinates, characteristic methods (including the ELLAM scheme) typically require more implementational work, especially for multidimensional problems.

Acknowledgments. We wish to acknowledge the many helpful discussions and suggestions regarding this work which were provided by Professors Michael A. Celia and Thomas F. Russell.

REFERENCES

[1] T. ARBOGAST, A. CHILAKAPATI, AND M. F. WHEELER, *A characteristic-mixed method for contaminant transport and miscible displacement*, in Computational Methods in Water Resources IX, Vol. I, Numerical Methods in Water Resources, T. Russell et al., eds., Computational Mechanics Publications and Elsevier Applied Science, London, New York, 1992, pp. 77–84.

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

[3] R. Bank, J. Bürger, W. Fichtner, and R. Smith, *Some upwinding techniques for finite element approximations of convection diffusion equations*, Numer. Math., 58 (1990), pp. 185–202.

[4] A. M. Baptista, *Solution of Advection-Dominated Transport by Eulerian-Lagrangian Methods Using the Backwards Method of Characteristics*, Ph.D. thesis, Massachusetts Institute of Technology, Cambridge, MA, 1987.

[5] J. W. Barrett and K. W. Morton, *Approximate symmetrization and Petrov-Galerkin methods for diffusion-convection problems*, Comput. Methods Appl. Mech. Engrg., 45 (1984), pp. 97–122.

[6] J. P. Benque and J. Ronat, *Quelques difficulties des modeles numeriques en hydraulique*, Comput. Methods Appl. Sci. Engrg., (1982), pp. 471–494.

[7] P. J. 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.

[8] P. J. Binning and 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 Resources Res., 32 (1996), pp. 103–114.

[9] E. T. Bouloutas and M. A. Celia, *An improved cubic Petrov-Galerkin method for simulation of transient advection-diffusion processes in rectangularly decomposable domains*, Comput. Methods Appl. Mech. Engrg., 91 (1991), pp. 289–308.

[10] A. Brooks and T. J. R. Hughes, *Streamline upwind Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations*, Comput. Methods Appl. Mech. Engrg., 32 (1982), pp. 199–259.

[11] M. A. Celia, *Eulerian-Lagrangian localized adjoint methods for contaminant transport simulations*, in Computational Methods in Water Resources X, Vol. I, Peters et al., eds., Water Science and Technology Library, Vol. 12, Kluwer Academic Publishers, Dordrecht, Netherlands, 1994, pp. 207–216.

[12] M. A. Celia and L. A. Ferrand, *A comparison of ELLAM formulations for simulation of reactive transport in groundwater*, Advances in Hydro-Science and Engineering, 1(B), Wang, ed., University of Mississippi Press, Jackson, MS, 1993, pp. 1829–1836.

[13] M. A. Celia, I. Herrera, E. T. Bouloutas, and J. S. Kindred, *A new numerical approach for the advective-diffusive transport equation*, Numer. Methods Partial Differential Equations, 5 (1989), pp. 203–226.

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

[15] M. A. Celia and S. Zisman, *An Eulerian-Lagrangian localized adjoint method for reactive transport in groundwater*, in Conference on Computational Methods in Water Resources VIII, Vol. I., G. Gambolati et al., eds., Springer-Verlag, Berlin, 1990, pp. 383–392.

[16] I. Christie, D. F. Griffiths, A. R. Mitchell, and O. C. Zienkiewicz, *Finite element methods for second order differential equations with significant first derivatives*, Internat. J. Numer. Engrg., 10 (1976), pp. 1389–1396.

[17] P. Colella, *A direct Eulerian MUSCL scheme for gas dynamics*, SIAM J. Sci. Statist. Comput., 6 (1985), pp. 104–117.

[18] R. A. Cox and T. Nishikawa, *A new total variation diminishing scheme for the solution of advective-dominant solute transport*, Water Resources Res., 27 (1991), pp. 2645–2654.

[19] H. K. Dahle, M. S. Espedal, R. E. Ewing, and O. Sævareid, *Characteristic adaptive subdomain methods for reservoir flow problems*, Numer. Methods Partial Differential Equations, (1990), pp. 279–309.

[20] H. K. Dahle, R. E. Ewing, and T. F. Russell, *Eulerian-Lagrangian localized adjoint methods for a nonlinear convection-diffusion equation*, Comput. Methods Appl. Mech. Engrg., 122 (1995), pp. 223–250.

[21] C. N. Dawson, *Godunov-mixed methods for advective flow problems in one space dimension*, SIAM J. Numer. Anal., 28 (1991), pp. 1282–1309.

[22] L. Demkowitz and J. T. Oden, *An adaptive characteristic Petrov-Galerkin finite element method for convection-dominated linear and nonlinear parabolic problems in two space variables*, Comput. Methods Appl. Mech. Engrg., 55 (1986), pp. 63–87.

[23] J. Douglas, Jr., and T. F. Russell, *Numerical methods for convection-dominated diffusion problems based on combining the method of characteristics with finite element or finite difference procedures*, SIAM J. Numer. Anal., 19 (1982), pp. 871–885.

[24] K. Eriksson and C. Johnson, *Adaptive streamline diffusion finite element methods for stationary convection-diffusion problems*, J. Math. Comp., 60 (1993), pp. 167–188.

[25] M. S. Espedal and R. E. Ewing, *Characteristic Petrov-Galerkin subdomain methods for two-phase immiscible flow*, Comput. Methods Appl. Mech. Engrg., 64 (1987), pp. 113–135.
[26] R. E. Ewing, ed., *The Mathematics of Reservoir Simulation*, Frontiers in Appl. Math. 1, SIAM, Philadelphia, PA, 1984.
[27] R. E. Ewing, *Operator splitting and Eulerian-Lagrangian localized adjoint methods for multi-phase flow*, in The Mathematics of Finite Elements and Applications VII, Academic Press, London, 1991, pp. 215–232.
[28] R. E. Ewing, T. F. Russell, and M. F. Wheeler, *Simulation of Miscible Displacement Using Mixed Methods and a Modified Method of Characteristics*, SPE 12241, Society of Petroleum Engineers of AIME, Dallas, TX, 1983, pp. 71–81.
[29] R. E. Ewing and H. Wang, *Eulerian-Lagrangian Localized Adjoint Methods for Linear Advection Equations*, Comput. Mech. '91, Springer International, Berlin, 1991, pp. 245-250.
[30] R. E. Ewing and H. Wang, *Eulerian-Lagrangian localized adjoint methods for linear advection or advection-reaction equations and their convergence analysis*, Comput. Mech., 12 (1993), pp. 97–121.
[31] R. E. Ewing and H. Wang, *An Eulerian-Lagrangian localized adjoint method for variable-coefficient advection-reaction problems*, in Advances in Hydro-Science and Engineering, 1(B), Wang, ed., University of Mississippi Press, Jackson, MS, 1993, pp. 2010–2015.
[32] R. E. Ewing and H. Wang, *An Eulerian-Lagrangian localized adjoint method with exponential-along-characteristic test functions for variable-coefficient advective-diffusive-reactive equations*, in Proceedings of KAIST Mathematical Workshop, Analysis and Geometry, U. Choi, D. Kwak, J. Yim, eds., Taejon, Korea, 8 (1993), pp. 77–91.
[33] R. E. Ewing and H. Wang, *Eulerian-Lagrangian localized adjoint methods for variable-coefficient advective-diffusive-reactive equations in groundwater contaminant transport*, in Advances in Optimization and Numerical Analysis, Mathematics and Its Applications, Vol. 275, Gomez and Hennart, eds., Kluwer Academic Publishers, Dordrecht, Netherlands, 1994, pp. 185–205.
[34] R. E. Ewing and H. Wang, *An optimal-order estimate for Eulerian–Lagrangian localized adjoint methods for variable-coefficient advection-reaction problems*, SIAM J. Numer. Anal., 33 (1996), pp. 318–348.
[35] R. S. Falk and G. R. Richter, *Local error estimates for a finite element method for hyperbolic and convection-diffusion equations*, SIAM J. Numer. Anal., 29 (1992), pp. 730–754.
[36] A. O. Garder, D. W. Peaceman, and A. L. Pozzi, *Numerical calculations of multidimensional miscible displacement by the method of characteristics*, Soc. Pet. Engrg. J., 4 (1964), pp. 26–36.
[37] P. Hansbo, *The characteristic streamline diffusion method for the time-independent incompressible Navier-Stokes equations*, Comput. Methods Appl. Mech. Engrg., 99 (1992), pp. 171–186.
[38] P. Hansbo and A. Szepessy, *A velocity-pressure streamline diffusion finite element method for the incompressible Navier-Stokes equations*, Comput. Methods Appl. Mech. Engrg., 84 (1990), pp. 107–129.
[39] A. Harten, B. Engquist, S. Osher, and S. Chakravarthy, *Uniformly high order accurate essentially non-oscillatory schemes*, III, J. Comput. Phys., 71 (1987), pp. 231–241.
[40] R. W. Healy and T. F. Russell, *A finite-volume Eulerian-Lagrangian localized adjoint method for solution of the advection-dispersion equation*, Water Resources Res., 29 (1993), pp. 2399–2413.
[41] R. W. Healy and T. F. Russell, *Solution of the advection-dispersion equation in two dimensions by a finite-volume Eulerian-Lagrangian localized adjoint method*, Adv. Water Resources, 21 (1998), pp. 11–26.
[42] I. Herrera, R. E. Ewing, M. A. Celia, and T. F. Russell, *Eulerian-Lagrangian localized adjoint methods: The theoretical framework*, Numer. Methods Partial Differential Equations, 9 (1993), pp. 431–458.
[43] J. M. Hervouet, *Applications of the method of characteristics in their weak formulation to solving two-dimensional advection-equations on mesh grids*, in Computational Techniques for Fluid Flow, Recent Advances in Numerical Methods in Fluids 5, C. Taylor, J. Johnson, W. Smith, eds., Pineridge Press, Swansea, 1986, pp. 149–185.
[44] T. J. R. Hughes and A. N. Brooks, *A Multidimensional Upwinding Scheme with No Cross-wind Diffusion*, Finite Element Methods for Convection Dominated Flows 34, ASME, New York, 1979.
[45] T. J. R. Hughes and M. Mallet, *A new finite element formulation for computational fluid dynamics:* III. *The general streamline operator for multidimensional advective-diffusive systems*, Comput. Methods Appl. Mech. Engrg., 58 (1986), pp. 305-328.

[46] T. J. R. Hughes, *Multiscale phenomena: Green functions, the Dirichlet-to-Neumann formulation, subgrid scale models, bubbles, and the origins of stabilized methods*, Comput. Methods Appl. Mech. Engrg., 127 (1995), pp. 387-401.

[47] C. Johnson and U. Nävert, *An analysis of some finite element methods for advection-diffusion equations*, in Analytical and Numerical Approaches to Asymptotic Problems in Analysis, O. Alexsson, L. Frank, and A. van der Sluis, eds., North–Holland, Amsterdam, 1981.

[48] C. Johnson and J. Pitkäranta, *An analysis of discontinuous Galerkin methods for a scalar hyperbolic equation*, J. Math. Comp., 46 (1986), pp. 1–26.

[49] C. Johnson, *Numerical Solutions of Partial Differential Equations by the Finite Element Method*, Cambridge University Press, Cambridge, UK, 1987.

[50] C. Johnson, A. Szepessy, and P. Hansbo, *On the convergence of shock-capturing streamline diffusion finite element methods for hyperbolic conservation laws*, J. Math. Comp., 54 (1990), pp. 107–129.

[51] C. Johnson and A. Szepessy, *Adaptive finite element methods for conservation laws based on a posteriori error estimates*, Comm. Pure Appl. Math., 98 (1995), pp. 199–234.

[52] G. Lube and D. Weiss, *Stabilized finite element methods for singularly perturbed parabolic problems*, Appl. Numer. Math., 17 (1995), pp. 431–459.

[53] N. Lu, *A semianalytical method of path line computation for transient finite-difference groundwater flow models*, Water Resources Res., 30 (1994), pp. 2449–2459.

[54] K. W. Morton, A. Priestley, and E. Süli, *Stability of the Lagrangian-Galerkin method with nonexact integration*, RAIRO M^2AN, 22 (1988), pp. 123–151.

[55] S. P. Neuman, *An Eulerian-Lagrangian numerical scheme for the dispersion-convection equation using conjugate space-time grids*, J. Comp. Phys., 41 (1981), pp. 270–294.

[56] G. F. Pinder and H. H. Cooper, *A numerical technique for calculating the transient position of the saltwater front*, Water Resources Res., (1970), pp. 875–882.

[57] O. Pironneau, *On the transport-diffusion algorithm and its application to the Navier-Stokes equations*, Numer. Math., 38 (1982), pp. 309–332.

[58] D. W. Pollock, *Semianalytical computation of path lines for finite-difference models*, Ground Water, 26 (1988), pp. 743–750.

[59] G. R. Richter, *An optimal-order error estimate for the discontinuous Galerkin method*, Math. Comp., 50 (1988), pp. 75–88.

[60] T. F. Russell, *Eulerian-Lagrangian localized adjoint methods for advection-dominated problems*, in Proceedings of the 13th Dundee Conference on Numerical Analysis, D. Griffiths and G. Watson, eds., Pitman Res. Notes Math. Ser. 228, Longman Scientific & Technical, Harlow, UK, 1990, pp. 206–228.

[61] T. F. Russell and R. V. Trujillo, *Eulerian-Lagrangian localized adjoint methods with variable coefficients in multiple dimensions*, in Computational Methods in Surface Hydrology, G. Gambolati, ed., Springer-Verlag, Berlin, 1990, pp. 357–363.

[62] A. L. Schafer-Perini and J. L. Wilson, *Efficient and accurate front tracking for two-dimensional groundwater flow models*, Water Resources Res., 27 (1991), pp. 1471–148.

[63] C. Shu and S. Osher, *Efficient implementation of essentially non-oscillatory shock capturing schemes*, J. Comput. Phys., 77 (1988), pp. 439–471.

[64] J. E. Våg, H. Wang, and H. K. Dahle, *Eulerian-Lagrangian localized adjoint methods for systems of nonlinear advective-diffusive-reactive equations*, Adv. Water Resources, 19 (1996), pp. 297–315.

[65] B. van Leer, *On the relation between the upwind-differencing schemes of Godunov, Engquist-Osher, and Roe*, SIAM J. Sci. Statist. Comp., 5 (1984), pp. 1–20.

[66] E. Varoglu and W. D. L. Finn, *Finite elements incorporating characteristics for one-dimensional diffusion-convection equation*, J. Comp. Phys., 34 (1980), pp. 371–389.

[67] H. Wang, *Eulerian-Lagrangian Localized Adjoint Methods: Analyses, Numerical Implementations and Their Applications*, Ph.D. thesis, Department of Mathematics, University of Wyoming, Laramie, WY, 1992.

[68] H. Wang, R. E. Ewing, and T. F. Russell, *ELLAM for variable-coefficient convection-diffusion problems arising in groundwater applications*, in Computational Methods in Water Resources IX, Vol. I, Numerical Methods in Water Resources, Computational Mechanics Publications and Elsevier Applied Science, London, New York, 1992, pp. 25–31.

[69] H. Wang and R. E. Ewing, *Optimal-order convergence rates for ELLAM for reactive transport and contamination in groundwater*, Numer. Methods Partial Differential Equations, 11 (1995), pp. 1–31.

[70] H. Wang, R. E. Ewing, and T. F. Russell, *Eulerian-Lagrangian localized methods for convection-diffusion equations and their convergence analysis*, IMA J. Numer. Anal., 15

(1995), pp. 405–459.

[71] H. Wang, R. E. Ewing, and M. A. Celia, *Eulerian-Lagrangian localized adjoint method for reactive transport with biodegradation*, Numer. Methods Partial Differential Equations, 11 (1995), pp. 229–254.

[72] H. Wang, R. C. Sharpley, and S. Man, *An ELLAM scheme for advection-diffusion equations in multi-dimensions*, in Computational Methods in Water Resources XI, A. Aldama et al., eds., Computational Mechanics Publications, Southampton, Boston, 1996, pp. 99–106.

[73] H. Wang, M. A. Al-Lawatia, and A. S. Telyakovskiy, *Runge-Kutta characteristic methods for first-order linear hyperbolic equations*, Numer. Methods Partial Differential Equations, 13 (1997), pp. 617–661.

[74] J. J. Westerink and D. Shea, *Consider higher degree Petrov-Galerkin methods for the solution of the transient convection-diffusion equation*, Internat. J. Numer. Methods Engrg., 28 (1989), pp. 1077-1101.

[75] M. F. Wheeler and C. N. Dawson, *An operator-splitting method for advection-diffusion-reaction problems*, MAFELAP Proceedings 6, J. Whiteman, ed., Academic Press, New York, 1988, pp. 463–482.

[76] D. Yang, *A characteristic-mixed method with dynamic finite element space for convection-dominated diffusion problems*, J. Comput. Appl. Math., 43 (1992), pp. 343–353.

[77] G. Zhou, *An Adaptive Streamline Diffusion Finite Element Method for Hyperbolic Systems in Gas Dynamics*, Ph.D. thesis, Department of Mathematics, University of Heidelberg, Heidelberg, Germany, 1992.

[78] G. Zhou, *A local L^2-error analysis of the streamline diffusion method for nonstationary convection-diffusion systems*, Mathematical Modeling and Numerical Analysis, RAIRO, 25 (1995), pp. 577–603.