Journal of Computational Physics **152,** 120–163 (1999)
Article ID jcph.1999.6239, available online at http://www.idealibrary.com on IDEAL®

A Family of Eulerian–Lagrangian Localized Adjoint Methods for Multi-dimensional Advection-Reaction Equations

Hong Wang,* Richard E. Ewing,† Guan Qin,‡ Stephen L. Lyons,‡
Mohamed Al-Lawatia,§ and Shushuang Man*

**Department of Mathematics, University of South Carolina, Columbia, South Carolina 29208; †Institute for Scientific Computation, Texas A&M University, College Station, Texas 77843-3404; ‡Upstream Strategic Research Center, Mobil Technology Company, 13777 Midway Road, Dallas, Texas 75244-4390; §Department of Mathematics and Statistics, Sultan Qaboos University, P.O. Box 36, Al-Khod Postal Code 123, Muscat, Sultanate of Oman*
E-mail: ewing@isc.tamu.edu

Received March 30, 1998; revised January 11, 1999

We develop a family of Eulerian–Lagrangian localized adjoint methods for the solution of the initial-boundary value problems for first-order advection-reaction equations on general multi-dimensional domains. Different tracking algorithms, including the Euler and Runge–Kutta algorithms, are used. The derived schemes, which are fully mass conservative, naturally incorporate inflow boundary conditions into their formulations and do not need any artificial outflow boundary conditions. Moreover, they have regularly structured, well-conditioned, symmetric, and positive-definite coefficient matrices, which can be efficiently solved by the conjugate gradient method in an optimal order number of iterations without any preconditioning needed. Numerical results are presented to compare the performance of the ELLAM schemes with many well studied and widely used methods, including the upwind finite difference method, the Galerkin and the Petrov–Galerkin finite element methods with backward-Euler or Crank–Nicolson temporal discretization, the streamline diffusion finite element methods, the monotonic upstream-centered scheme for conservation laws (MUSCL), and the Minmod scheme. © 1999 Academic Press

Key Words: advection-reaction equations; characteristic methods; comparison of numerical methods; Eulerian–Lagrangian methods; linear hyperbolic problems; numerical solutions of advection-reaction equations.

---

1. INTRODUCTION

Many difficult problems arise in the numerical simulation of fluid flow processes within porous media in petroleum reservoir simulation and in subsurface contaminant transport

0021-9991/99 $30.00

and remediation. The mathematical models used to describe these complex flow processes are coupled systems of time-dependent nonlinear partial differential equations (PDEs) and constraining equations. These problems are basically advection or reaction dominated. Because of the nonlinearity and couplings of these governing PDEs, the moving steep fronts present in the solutions of these PDEs, the effect of the singularities at wells, the heterogeneities of media that yield rough coefficients in these PDEs, and the enormous size of field-scale applications, these systems present severe difficulties.

A mathematical model for describing compressible, multicomponent fluid flow processes within porous media in petroleum reservoir simulation and in subsurface modeling and remediation can be represented by a system of PDEs [1, 2, 5, 24, 39, 51]

$$\begin{aligned} &\frac{\partial}{\partial t}(\phi\rho) + \nabla\cdot(\rho\mathbf{v}) = \rho q, && \mathbf{x}\in\Omega, t\in[0,T], \\ &\mathbf{v} = -\frac{\mathbf{K}}{\mu}(\nabla p - \rho g\nabla d), && \mathbf{x}\in\Omega, t\in[0,T], \end{aligned} \tag{1.1}$$

$$\frac{\partial}{\partial t}(\kappa_i\phi\rho c_i) + \nabla\cdot(\rho\mathbf{v}c_i) + R_i c_i = \rho\bar{c}_i q, \qquad \mathbf{x}\in\Omega, t\in[0,T], i = 1,\ldots,N-1. \tag{1.2}$$

Here $\Omega \subset \mathbb{R}^d$ is a bounded domain that has a Lipschitz continuous boundary $\partial\Omega$, $\mathbf{x} := (x_1, x_2, \ldots, x_d)$, $\nabla := (\partial/\partial x_1, \partial/\partial x_2, \ldots, \partial/\partial x_d)$. p and $\mathbf{v} := (V_1, V_2, \ldots, V_d)$ are the pressure and Darcy velocity of the fluid mixture, ϕ and $\mathbf{K}(\mathbf{x})$ are the porosity and permeability of the medium, ρ is the mass density of the fluid mixture, q is the volumetric flow rate accounting for the effect of the source and sink terms (e.g., injection and production wells), $d(\mathbf{x})$ is the reservoir depth, $\mu(c_1, \ldots, c_N)$ is the viscosity of the fluid mixture, and κ_i is the retardation coefficient that has pronounced effects in such enhanced oil reservoir technologies as polymer and surfactant flooding and in subsurface contaminant transport and remediation [1, 5, 24, 39]. $c_i(\mathbf{x}, t)$ $(i = 1, 2, \ldots, N)$, with $\sum_{i=1}^N c_i = 1$, stand for the mass fractions of the ith component in the fluid mixture, such as methane, ethane, propane, and n-butane in reservoir simulation [1, 2, 24, 51], or brine and trace-species radionuclides in subsurface contaminant transport of hazardous nuclear waste [57, 58], or organic contaminants and nutrients in bioremediation [7, 46, 47]. The $R_i(c_1, \ldots, c_N)$ $(i = 1, 2, \ldots, N-1)$ represent the first-order reaction coefficients that could have significant effect in subsurface contaminant transport and remediation, and the $\bar{c}_i(\mathbf{x}, t)$ $(i = 1, 2, \ldots, N)$ are either the specified concentrations of the injected fluids at injection wells or the resident concentrations at the production wells. In Eq. (1.2) we chose to neglect the effect of diffusion-dispersion, because it is often very small. We refer readers to [24] for models including the diffusion-dispersion term.

Different boundary conditions may be imposed on the system (1.1)–(1.2) depending on specific applications. For instance, in petroleum reservoir simulation, the boundary $\partial\Omega$ is often a noflow boundary characterized by

$$\mathbf{v}\cdot\mathbf{n} = 0, \qquad \mathbf{x}\in\partial\Omega, \tag{1.3}$$

where $\mathbf{n}(\mathbf{x})$ is the unit outward normal to the boundary $\partial\Omega$. This reflects the fact that the boundary of the reservoir is impermeable. In this case, no boundary condition should be specified for Eq. (1.2).

In a numerical simulation of subsurface contaminant transport and remediation, part of the boundary $\partial\Omega^{(I)}$ is an inflow boundary characterized by

$$\mathbf{v} \cdot \mathbf{n} < 0, \qquad \mathbf{x} \in \partial\Omega^{(I)}. \tag{1.4}$$

In this case an inflow Dirichlet boundary condition is imposed on the boundary $\partial\Omega^{(I)}$ for Eq. (1.2)

$$c(\mathbf{x}, t) = g(\mathbf{x}, t), \qquad \mathbf{x} \in \partial\Omega^{(I)},\ t \in [0, T]. \tag{1.5}$$

Nevertheless, no boundary condition should be specified for Eq. (1.2) on the outflow boundary $\partial\Omega^{(O)}$ characterized by

$$\mathbf{v} \cdot \mathbf{n} > 0, \qquad \mathbf{x} \in \partial\Omega^{(O)}, \tag{1.6}$$

since the concentration $c(\mathbf{x}, t)$ on $\partial\Omega^{(O)}$ is determined completely by its value inside the domain Ω.

In addition, the following initial conditions for the pressure and the concentration

$$\begin{aligned} p(\mathbf{x}, 0) &= p_0(\mathbf{x}), \qquad \mathbf{x} \in \Omega, \\ c(\mathbf{x}, 0) &= c_0(\mathbf{x}), \qquad \mathbf{x} \in \Omega \end{aligned} \tag{1.7}$$

are imposed for the system (1.1)–(1.2).

Remark 1.1. In subsurface contaminant transport and remediation and in petroleum reservoir simulation [5, 7, 24, 39, 46, 51, 57, 58], the fluid flows are often incompressible, that is, characterized by

$$\rho = \rho_r \tag{1.8}$$

with ρ_r being the reference density, or weakly compressible, that is, described by the equation of state

$$\rho = \rho_r \exp(c_p(p - p_r)) \tag{1.9}$$

and its various simplified versions. In Eq. (1.9), c_p is the compressibility of the fluid that is usually very small. ρ_r is the density at the reference pressure p_r.

Remark 1.2. In multiphase flows such as the immiscible displacement of hydrocarbons by water in secondary recovery and the compositional models that describe the transport of N chemical components in the gas, oil, and water phases, an analogue of Eqs. (1.1) holds for each phase. After some rearrangements, the PDEs for the different phases can be rewritten as a nonlinear parabolic PDE for the pressure, and one (for two-phase flows) or two (for three-phase flows) nonlinear hyperbolic PDEs for the saturations of the phases [10, 13, 24, 51]. Meanwhile, an analogue of Eq. (1.2) still holds for each of the N chemical components in multiphase/multicomponent fluid flows. The nonlinear hyperbolic PDEs also arise in many other important applications, such as famous Euler equations in the mathematical modeling of aerodynamics, and often present severe numerical and analytical difficulties [34, 48, 66].

Remark 1.3. While advection-reaction PDEs and nonlinear hyperbolic conservation laws present serious difficulties that are common for various applications, they also exhibit salient features/difficulties for different applications. In aerodynamics, the concerned fluids are mainly (e.g., ideal) highly compressible gases. The width of the steep fronts in the solutions is often so small that these fronts are usually treated as "shock discontinuities." The interaction of these shock discontinuities could be extremely complicated. In the porous medium fluid flow processes, the governing PDEs (1.1)–(1.2) were obtained via a volume averaging mechanism and should model the behavior of the flows and the steep fronts of the solutions on a macroscopic scale. Due to the enormous size of field-scale applications, quite large grid-spacings must be used in a field-scale simulation. Hence, the fronts of the solutions are still very steep on any reasonable grids but probably do not form a real shock discontinuity, especially for the unknown concentrations governed by the transport PDEs (1.2). Moreover, the heterogeneities of the reservoir media and the effect of singular sources and sinks (representing injection and production wells, in practice) often introduce severe new difficulties. In addition, in petroleum reservoir simulation, the pressure could be fairly high (say, 5000–6000 psia), especially near the wells. This in turn causes the deformation of the porous media and introduces an additional phase, whose behavior (e.g., adsorption, desorption, etc.) needs to be simulated in the simulation. Among others, the following formula has been used to model the deformation of porous media [2]

$$\phi = \phi_r(\mathbf{x}) \exp(c_\phi (p - p_r)), \tag{1.10}$$

where c_ϕ is the compressibility of the porous medium, and $\phi_r(\mathbf{x})$ is the porosity of the medium at the reference pressure p_r. All these issues introduce essential difficulties that are probably encountered less frequently in many other applications.

Remark 1.4. Note that the principal variables of physical interest in the system (1.1)–(1.2) are the concentrations $c_i(\mathbf{x}, t)$ $(i = 1, 2, \ldots, N)$ of the N components. In petroleum reservoir simulation, they show fluid composition changes due to fluid flows and phase behavior changes. Equivalently, they show how much oil is recovered. In subsurface contaminant transport and remediation, changes in c_i illustrate the transport of various species or components in groundwater or the effect of remediation, which one wants to determine. Hence, this paper focuses on improving the numerical approximation techniques to the transport equations (1.2). After being decoupled from the pressure PDE (1.1) and other transport PDEs (1.2) for $j = 1, 2, \ldots, N - 1$ with $j \neq i$, the transport PDE (1.2) for the component i is virtually linear with respect to the concentration c_i with a possible nonlinearity arising from the first-order reaction coefficient $R_i(c_1, \ldots, c_N)$. Therefore, in this paper we develop a numerical method for linear advection-reaction PDEs with a particular interest on its application to porous medium flows.

2. NUMERICAL METHODS FOR ADVECTION-DOMINATED PDEs

The numerical treatment of advection-reaction PDEs often presents severe numerical and analytical difficulties. Standard finite difference or finite element methods tend to generate numerical solutions with severe non-physical undershoot and overshoot. In industrial applications, upstream weighting techniques are commonly used to stabilize the numerical approximations in large-scale simulators. However, these methods produce excessive numerical dispersion and potentially spurious effects related to the orientation of the grid. Two general classes of improved approximations can be identified from the literature: the Eulerian methods that use the standard temporal discretization and the characteristic methods whose main distinguishing feature is the use of characteristics to carry out the discretization in time.

2.1. *Eulerian Methods*

Most Eulerian methods are based on upstream weighting techniques. The optimal test function methods [4, 11, 14] attempt to minimize spatial errors and yield an upstream bias in the resulting numerical schemes. Hence, they are susceptible to time truncation errors that introduce numerical dispersion and the restrictions on the size of the time steps. They tend to be ineffective for transient advection-dominated problems. Some other Eulerian methods [8, 17, 76] attempt to reduce the local truncation errors by using nonzero spatial errors to cancel temporal errors. The streamline diffusion finite element methods (SDMs) [9, 40, 41, 44] add a numerical diffusion only in the direction of streamlines with no crosswind diffusion introduced and usually generate fairly accurate numerical solutions. However, these methods contain 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 methods could still generate numerical solutions with oscillations. But if it is chosen too large, the methods will introduce excessive numerical dispersion and smear the numerical solutions. Unfortunately, an optimal choice of the parameter is not clear and is heavily problem-dependent.

High resolution methods, such as the Godunov methods, the total variation diminishing methods (TVD), and the essentially non-oscillatory (ENO) methods [15, 18, 22, 36, 64, 68, 70], are well suited for the solution of nonlinear hyperbolic conservation laws and resolve shock discontinuities in the solutions without excessive smearing or spurious oscillations. Moreover, they conserve mass; this property is of essential importance in virtually all applications. In aerodynamics where the fluids are highly compressible and often exhibit shock discontinuities, high resolution methods have been successfully applied and have generated very satisfactory results. Because these methods are mostly explicit, the size of the time steps in these methods is subject to the CFL constraint. Few references could be found in the literature on the application of high resolution methods to porous medium flows in the presence of strongly heterogeneous porous media and injection and production wells.

2.2. *Characteristic Methods*

Because of the hyperbolic nature of advective transport, characteristic methods have been investigated extensively and have been successfully applied to solve advection-reaction PDEs [16, 21, 34, 35, 48, 50, 52–55, 65, 71].

It is worth noticing that the Courant number actually indicates the number of cells information propagates on the numerical grids per time step. Because Eulerian methods use the standard temporal discretization in the time direction, they cannot accurately simulate all of the wave interactions that take place if the information propagates more than one cell per time step (i.e., if the CFL condition is violated), either for the reason of stability (for explicit methods) or for the reason of accuracy (for implicit methods). On the other hand, by using the characteristic tracking, characteristic methods follow the movement of information or particles as well as their interactions. In fact, in the Lagrangian coordinates, the advection-reaction PDEs (1.2) can be rewritten as non-homogeneous first-order ordinary differential equations. In other words, following the characteristics one would only see the effect of the reaction terms and the right-hand side source terms but not the effect of the moving steep fronts. Hence, the solutions are much smoother along the characteristics than they are in the time direction. This explains why characteristic methods allow large time steps to be used in a numerical simulation while still maintaining its stability and accuracy.

While characteristic methods have improved accuracy and efficiency as well as other advantages, they usually require extra implementational effort and raise many implementational and analytical issues that need to be addressed. Traditional forward or particle tracking methods advance the grids following the characteristics. They greatly reduce temporal errors and, thus, generate fairly accurate solutions even if large time steps are used. However, they often severely distort the evolving grids and greatly complicate the solution procedures. The modified method of characteristics (MMOC) [21] follows the flow direction by tracking the characteristics backward from a fixed grid at the current time step and hence, avoids the grid distortion problems present in forward tracking methods. The MMOC symmetrizes and stabilizes the governing PDEs which greatly reduces temporal errors; therefore it allows for large time steps in a simulation without the loss of accuracy and eliminates the excessive numerical dispersion and grid orientation effects present in many Eulerian methods [24, 62]. However, the major drawbacks of many previous characteristic methods are that they fail to conserve mass and have difficulties in treating general boundary conditions.

2.3. *The Eulerian–Lagrangian Localized Adjoint Method (ELLAM)*

The Eulerian–Lagrangian localized adjoint method (ELLAM) [12, 38] was originally proposed by Celia, Russell, Herrera, and Ewing in solving one-dimensional (constant-coefficient) advection-diffusion PDEs. The ELLAM methodology provides a general characteristic solution procedure for advection-diffusion PDEs and a consistent framework for treating general boundary conditions and maintaining mass conservation. Thus, it overcomes the two principal shortcomings of the previous characteristic methods while maintaining their numerical advantages. Subsequently, Healy and Russell [37] extended the ELLAM concept and developed a finite-volume ELLAM scheme for one-dimensional advection-diffusion PDEs. Dahle, Russell, and Ewing [20, 25] developed an ELLAM scheme for the one-dimensional Buckley–Leverett equation arising from immiscible fluid flow processes in porous media. Binning and Celia [6] and some of the authors [72, 75] developed ELLAM schemes for two-dimensional advection-diffusion PDEs.

In this paper we develop a family of ELLAM schemes for first-order linear advection-reaction PDEs on general d-dimensional spatial domains. Recall that for second-order advection-diffusion PDEs, boundary conditions are specified at both inflow and outflow boundaries. However, for first-order advection-reaction PDEs, boundary conditions can be specified only at inflow boundaries. Consequently, many Eulerian and characteristic finite difference methods often require an artificial outflow boundary condition to be added. In contrast, the ELLAM schemes developed in this paper treat outflow boundaries in a systematic manner and conserve mass without any artificial outflow boundary conditions needed. Second, the ELLAM schemes for second-order advection-diffusion PDEs yield coefficient matrices that have mesh-size-dependent condition numbers and are not necessarily symmetric for certain types of boundary conditions. In contrast, the ELLAM schemes developed in this paper yield well-conditioned, regularly structured, symmetric and positive-definite coefficient matrices for first-order linear advection-reaction PDEs. Consequently, the discrete algebraic systems can be efficiently solved by the conjugate gradient method in an optimal order number of operations (i.e., the same order of operations as explicit methods) without any preconditioning needed. Third, ELLAM schemes were previously developed for second-order advection-diffusion PDEs on two-dimensional rectangular domains [6, 72, 75] with rectangular spatial partitions and piecewise-bilinear trial and test functions. In this paper, ELLAM schemes are developed for first-order linear advection-reaction PDEs on general d-dimensional spatial domains with d-dimensional simplex or rectangular spatial partitions and piecewise polynomial trial and test functions of degree less than or equal to κ. Finally, as in the case of advection-diffusion PDEs, the ELLAM schemes significantly reduce temporal truncation errors and generate accurate numerical solutions for first-order advection-reaction PDEs, even if large time steps are used.

The rest of the paper is organized as follows: In Section 3, we derive a reference weak formulation for linear advection-reaction PDEs. In Section 4, we develop a family of ELLAM schemes. In Section 5, we discuss implementational issues. In Section 6, we briefly recall some widely used numerical methods for advection-reaction equations. In Section 7, we perform numerical experiments to compare the performance of the ELLAM schemes with many well studied and widely used methods, including the upwinding finite difference method, various Galerkin and Petrov–Galerkin finite element methods, the streamline diffusion finite element methods, the monotonic upstream-centered scheme for conservation laws (MUSCL) [15, 70], and the Minmod scheme [36, 64]. Section 8 contains summary and discussions.

3. VARIATIONAL FORMULATION

3.1. *Model Problem*

In this paper, we develop a family of ELLAM schemes for linear advection-reaction PDEs. Hence, we assume that all the coefficients in the transport PDEs (1.2) are known and consider the following multi-dimensional linear advection-reaction PDE

$$Lc := \frac{\partial(\Phi c)}{\partial t} + \nabla \cdot (\mathbf{v}c(\mathbf{x}, t)) + R(\mathbf{x}, t)c = q(\mathbf{x}, t), \qquad \mathbf{x} \in \Omega,\ t \in (0, T], \tag{3.1}$$

where $\Omega \subset \mathbb{R}^d$ is a bounded domain with a Lipschitz continuous boundary $\partial\Omega$. We decompose the space-time boundary $\Gamma := \partial\Omega \times [0, T]$ as

$$\Gamma := \Gamma^{(I)} \cup \Gamma^{(N)} \cup \Gamma^{(O)}, \tag{3.2}$$

with

$$\begin{aligned} \Gamma^{(I)} &:= \{(\mathbf{x}, t) \mid \mathbf{x} \in \partial\Omega, t \in [0, T], \mathbf{v}(\mathbf{x}, t) \cdot \mathbf{n}(\mathbf{x}) < 0\}, \\ \Gamma^{(N)} &:= \{(\mathbf{x}, t) \mid \mathbf{x} \in \partial\Omega, t \in [0, T], \mathbf{v}(\mathbf{x}, t) \cdot \mathbf{n}(\mathbf{x}) = 0\}, \\ \Gamma^{(O)} &:= \{(\mathbf{x}, t) \mid \mathbf{x} \in \partial\Omega, t \in [0, T], \mathbf{v}(\mathbf{x}, t) \cdot \mathbf{n}(\mathbf{x}) > 0\} \end{aligned} \tag{3.3}$$

being the inflow, noflow, and outflow space-time boundaries, respectively. In general, $\Gamma^{(I)}$, $\Gamma^{(N)}$, and $\Gamma^{(O)}$ are time dependent and are not necessarily connected. Because Eq. (3.1) is a first-order hyperbolic equation, only an inflow boundary condition is specified on the inflow boundary $\Gamma^{(I)}$

$$c(\mathbf{x}, t) = g(\mathbf{x}, t), \qquad (\mathbf{x}, t) \in \Gamma^{(I)}, \tag{3.4}$$

and no boundary condition should be specified on the noflow boundary $\Gamma^{(N)}$ or the outflow boundary $\Gamma^{(O)}$. In addition, an initial condition

$$c(\mathbf{x}, 0) = c_0(\mathbf{x}) \tag{3.5}$$

is needed to close Eq. (3.1).

3.2. *Definition of Test Functions*

Let N_t be a positive integer. We define a quasi-uniform temporal partition on $[0, T]$ by

$$0 =: t_0 < t_1 < t_2 < \cdots < t_{N-1} < t_{N_t} := T. \tag{3.6}$$

Multiplying the governing equation (3.1) by the space-time test functions $w(\mathbf{x}, t)$ that are continuous and piecewise smooth, vanish outside the space-time strip $\Omega \times [t_{n-1}, t_n]$, and are discontinuous in time at time t_{n-1}, we obtain a space-time weak formulation

$$\begin{aligned}
&\int_\Omega \Phi(\mathbf{x}, t_n) c(\mathbf{x}, t_n) w(\mathbf{x}, t_n)\, d\mathbf{x} + \int_{t_{n-1}}^{t_n} \int_{\partial\Omega} \mathbf{v}(\mathbf{x}, t) \cdot \mathbf{n}(\mathbf{x}) c(\mathbf{x}, t) w(\mathbf{x}, t)\, dS \\
&\qquad - \int_{t_{n-1}}^{t_n} \int_\Omega c(\mathbf{x}, t)(\Phi w_t + \mathbf{v} \cdot \nabla w - Rw)(\mathbf{x}, t)\, d\mathbf{x}\, dt \\
&\quad = \int_\Omega \Phi(\mathbf{x}, t_{n-1}) c(\mathbf{x}, t_{n-1}) w(\mathbf{x}, t_{n-1}^+)\, d\mathbf{x} + \int_{t_{n-1}}^{t_n} \int_\Omega q(\mathbf{x}, t) w(\mathbf{x}, t)\, d\mathbf{x}\, dt,
\end{aligned} \tag{3.7}$$

where $w(\mathbf{x}, t_{n-1}^+) := \lim_{t \to t_{n-1}^+} w(\mathbf{x}, t)$, which takes into account the fact that $w(\mathbf{x}, t)$ is discontinuous in time at time t_{n-1}.

In the ELLAM framework [12, 38], the test functions w are chosen to satisfy the adjoint equation of the governing equation (3.1)

$$\Phi w_t + \mathbf{v} \cdot \nabla w - Rw = 0. \tag{3.8}$$

Let $\mathbf{y} = \mathbf{r}(\theta; \bar{\mathbf{x}}, \bar{t})$ be the characteristic passing through a given point $(\bar{\mathbf{x}}, \bar{t})$ with $\bar{t} \in [t_{n-1}, t_n]$ and let it be determined by the initial-value problem

$$\begin{gathered}
\frac{d\mathbf{y}}{d\theta} = \mathbf{v}^\Phi(\mathbf{y}, \theta) := \frac{\mathbf{v}(\mathbf{y}, \theta)}{\Phi(\mathbf{y}, \theta)}, \\
\mathbf{y}|_{\theta = \bar{t}} = \bar{\mathbf{x}}.
\end{gathered} \tag{3.9}$$

Equation (3.8) is rewritten

$$\begin{gathered}
-\frac{d}{d\theta} w(\mathbf{r}(\theta; \bar{\mathbf{x}}, \bar{t}), \theta) + R^\Phi(\mathbf{r}(\theta; \bar{\mathbf{x}}, \bar{t}), \theta) w(\mathbf{r}(\theta; \bar{\mathbf{x}}, \bar{t}), \theta) = 0, \\
w(\mathbf{r}(\theta; \bar{\mathbf{x}}, \bar{t}), \theta)|_{\theta = \bar{t}} = w(\bar{\mathbf{x}}, \bar{t}),
\end{gathered} \tag{3.10}$$

where $R^\Phi(\mathbf{x},t) := R(\mathbf{x},t)/\Phi(\mathbf{x},t)$. Solving Eq. (3.10) leads to the following expression for the test functions w

$$w(\mathbf{r}(\theta;\bar{\mathbf{x}},\bar{t}),\theta) = w(\bar{\mathbf{x}},\bar{t})e^{-\int_\theta^{\bar{t}} R^\Phi(\mathbf{r}(\gamma;\bar{\mathbf{x}},\bar{t}),\gamma)\,d\gamma}. \tag{3.11}$$

Remark 3.1. Equation (3.11) shows that the test functions w in the ELLAM formulation should vary exponentially along the characteristics defined by the ordinary differential equation (3.9). Moreover, once the value of $w(\bar{\mathbf{x}},\bar{t})$ is specified, the value of $w(\mathbf{r}(\theta;\bar{\mathbf{x}},\bar{t}),\theta)$ along the characteristic $\mathbf{y} = \mathbf{r}(\theta;\bar{\mathbf{x}},\bar{t})$ is completely determined. Therefore, to define the test functions w on the space-time strip $\Omega\times[t_{n-1},t_n]$, we only need to define w on $\bar{\Omega}$ at the time t_n and on the space-time outflow boundary $\Gamma_n^{(O)}$, where

$$\begin{aligned}\Gamma_n &:= \partial\Omega\times[t_{n-1},t_n],\\ \Gamma_n^{(I)} &:= \{(\mathbf{x},t)\in\Gamma_n \mid \mathbf{v}(\mathbf{x},t)\cdot\mathbf{n}(\mathbf{x}) < 0\},\\ \Gamma_n^{(N)} &:= \{(\mathbf{x},t)\in\Gamma_n \mid \mathbf{v}(\mathbf{x},t)\cdot\mathbf{n}(\mathbf{x}) = 0\},\\ \Gamma_n^{(O)} &:= \{(\mathbf{x},t)\in\Gamma_n \mid \mathbf{v}(\mathbf{x},t)\cdot\mathbf{n}(\mathbf{x}) > 0\}.\end{aligned} \tag{3.12}$$

3.3. *A Reference Equation*

We now evaluate the second term on the right-hand side of Eq. (3.7). To avoid confusion, we replace the dummy variables $\mathbf{x}$ and t in this term by $\mathbf{y}$ and θ and reserve $\mathbf{x}$ and t for the points in Ω at time t_n or on the space-time boundary Γ_n, representing either heads or feet of characteristics. Let $\Omega(\theta)\subset\Omega$ be the set of the points that will flow out of the domain Ω during the time period $[\theta,t_n]$. Hence, for any $\mathbf{y}\in\Omega\backslash\Omega(\theta)$, there exists an $\mathbf{x}\in\Omega$ such that $\mathbf{y} = \mathbf{r}(\theta;\mathbf{x},t_n)$. Similarly, for any $(\mathbf{y},\theta)\in\Omega(\theta)$, there exists a pair $(\mathbf{x},t)\in\Gamma_n^{(O)}$ such that $\mathbf{y} = \mathbf{r}(\theta;\mathbf{x},t)$. Therefore,

$$\begin{aligned}&\int_{t_{n-1}}^{t_n}\int_\Omega q(\mathbf{y},\theta)w(\mathbf{y},\theta)\,d\mathbf{y}\,d\theta\\ &\quad= \int_{t_{n-1}}^{t_n}\int_{\Omega\backslash\Omega(\theta)} q(\mathbf{r}(\theta;\mathbf{x},t_n),\theta)w(\mathbf{r}(\theta;\mathbf{x},t_n),\theta)\,d\mathbf{r}\,d\theta\\ &\quad\quad+ \int_{t_{n-1}}^{t_n}\int_{\Omega(\theta)} q(\mathbf{r}(\theta;\mathbf{x},t),\theta)w(\mathbf{r}(\theta;\mathbf{x},t),\theta)\,d\mathbf{r}\,d\theta.\end{aligned} \tag{3.13}$$

Applying the Euler quadrature at t_n to the first term on the right-hand side yields

$$\begin{aligned}&\int_{t_{n-1}}^{t_n}\int_{\Omega\backslash\Omega(\theta)} q(\mathbf{r}(\theta;\mathbf{x},t_n),\theta)w(\mathbf{r}(\theta;\mathbf{x},t_n),\theta)\,d\mathbf{r}\,d\theta\\ &\quad= \int_\Omega\int_{t^*(\mathbf{x})}^{t_n} q(\mathbf{r}(\theta;\mathbf{x},t_n),\theta)w(\mathbf{r}(\theta;\mathbf{x},t_n),\theta)|\mathbf{J}_1(\theta;\mathbf{x},t_n)|\,d\theta\,d\mathbf{x}\\ &\quad= \int_\Omega q(\mathbf{x},t_n)w(\mathbf{x},t_n)\left[\int_{t^*(\mathbf{x})}^{t_n} e^{-R^\Phi(\mathbf{x},t_n)(t_n-\theta)}\,d\theta\right]d\mathbf{x} + E_1(q,w)\\ &\quad= \int_\Omega \Psi^{(1)}(\mathbf{x},t_n)q(\mathbf{x},t_n)w(\mathbf{x},t_n)\,d\mathbf{x} + E_1(q,w),\end{aligned} \tag{3.14}$$

where

$$\mathbf{J}_1(\theta;\mathbf{x},t_n) := \left|\frac{\partial\mathbf{r}(\theta;\mathbf{x},t_n)}{\partial\mathbf{x}}\right| = 1 + \mathcal{O}(t_n-\theta) \tag{3.15}$$

is the Jacobian determinant of the transformation from $\mathbf{x}$ to $\mathbf{r}$. To accurately measure the effect of the reaction and source terms on a particle traveling from the previous time level or the inflow boundary to the current time level, for $\mathbf{x}\in\Omega$ at time t_n, we introduce a degenerating time step factor $\Delta t^{(I)}(\mathbf{x})$ by

$$\Delta t^{(I)}(\mathbf{x}) := \begin{cases} t_n - t_{n-1}, & \text{if } \mathbf{r}(\theta;\mathbf{x},t_n)\in\Omega, \forall\theta\in[t_{n-1},t_n],\\ t_n - t^*(\mathbf{x}), & \text{otherwise.}\end{cases} \tag{3.16}$$

In the latter case, $t^*(\mathbf{x})\in[t_{n-1},t_n]$ is the time when $\mathbf{r}(\theta;\mathbf{x},t_n)$ intersects the boundary $\partial\Omega$ (i.e., $\mathbf{r}(t^*(\mathbf{x});\mathbf{x},t_n)\in\partial\Omega$). In Eq. (3.14), the $\Psi^{(1)}(\mathbf{x},t_n)$ and $E_1(q,w)$ are given by

$$\Psi^{(1)}(x,t_n) := \begin{cases} \dfrac{1-e^{-R^\Phi(\mathbf{x},t_n)\Delta t^{(I)}(\mathbf{x})}}{R^\Phi(\mathbf{x},t_n)}, & \text{if } R^\Phi(\mathbf{x},t_n)\neq 0,\\ \Delta t^{(I)}(\mathbf{x}), & \text{otherwise,}\end{cases} \tag{3.17}$$

and

$$\begin{aligned} E_1(q,w) := \int_\Omega\int_{t^*(\mathbf{x})}^{t_n} &[q(\mathbf{r}(\theta;\mathbf{x},t_n),\theta)|\mathbf{J}_1(\theta;\mathbf{x},t_n)| - q(\mathbf{x},t_n)]\\ &\times w(\mathbf{x},t_n)e^{-R^\Phi(\mathbf{x},t_n)(t_n-\theta)}\,d\theta\,d\mathbf{x}. \end{aligned} \tag{3.18}$$

The second term on the right-hand side of Eq. (3.13) is treated similarly

$$\begin{aligned} &\int_{t_{n-1}}^{t_n}\int_{\Omega(\theta)} q(\mathbf{r}(\theta;\mathbf{x},t),\theta)w(\mathbf{r}(\theta;\mathbf{x},t),\theta)\,d\mathbf{r}\,d\theta\\ &\quad= \int_{\Gamma_n^{(O)}}\int_{t^*(\mathbf{x},t)}^{t} q(\mathbf{r}(\theta;\mathbf{x},t),\theta)w(\mathbf{r}(\theta;\mathbf{x},t),\theta)|\mathbf{J}_2(\theta;\mathbf{x},t)|\,d\theta\,dS\\ &\quad= \int_{\Gamma_n^{(O)}} \mathbf{v}(\mathbf{x},t)\cdot\mathbf{n}(\mathbf{x})q(\mathbf{x},t)w(\mathbf{x},t)\left[\int_{t^*(\mathbf{x},t)}^{t} e^{-R^\Phi(\mathbf{x},t)(t-\theta)}\,d\theta\right]d\theta\,dS + E_2(q,w)\\ &\quad= \int_{\Gamma_n^{(O)}} \mathbf{v}(\mathbf{x},t)\cdot\mathbf{n}(\mathbf{x})\Psi^{(2)}(\mathbf{x},t)q(\mathbf{x},t)w(\mathbf{x},t)\,dS + E_2(q,w), \end{aligned} \tag{3.19}$$

where $\mathbf{J}_2(\theta;\mathbf{x},t) = \mathbf{v}(\mathbf{x},t)\cdot\mathbf{n}(\mathbf{x})[1+\mathcal{O}(t-\theta)]$ is the Jacobian determinant of the transformation from $(\mathbf{x},t)\in\Gamma_n^{(O)}$ to $\mathbf{r}(\theta;\mathbf{x},t)\in\Omega(\theta)$ at time θ. Also, for $(\mathbf{x},t)\in\Gamma_n^{(O)}$, we introduce another degenerate time step factor $\Delta t^{(O)}(\mathbf{x},t)$ by

$$\Delta t^{(O)}(\mathbf{x},t) := \begin{cases} t - t_{n-1}, & \text{if } \mathbf{r}(\theta;\mathbf{x},t)\in\Omega, \forall\theta\in[t_{n-1},t],\\ t_n - t^*(\mathbf{x},t), & \text{otherwise,}\end{cases} \tag{3.20}$$

where we denote by $t^*(\mathbf{x},t)\in[t_{n-1},t]$ the time when $\mathbf{r}(\theta;\mathbf{x},t)$ intersects the boundary $\partial\Omega$.

$\Psi^{(2)}(\mathbf{x},t)$ and $E_2(q,w)$ are given by

$$\Psi^{(2)}(\mathbf{x},t) := \begin{cases} \dfrac{1-e^{-R^{\Phi}(\mathbf{x},t)}\Delta t^{(O)}(\mathbf{x},t)}{R^{\Phi}(\mathbf{x},t)}, & \text{if } R(\mathbf{x},t)\neq 0,\\ \Delta t^{(O)}(\mathbf{x},t), & \text{otherwise,} \end{cases} \tag{3.21}$$

and

$$E_2(q,w) := \int_{\Gamma_n^{(O)}}\int_{t^*(\mathbf{x},t)}^{t} [q(\mathbf{r}(\theta;\mathbf{x},t),\theta)|\mathbf{J}_2(\mathbf{r}(\theta;\mathbf{x},t))| - \mathbf{v}(\mathbf{x},t)\cdot\mathbf{n}(\mathbf{x})q(\mathbf{x},t)] \times w(\mathbf{x},t)e^{-R^{\Phi}(\mathbf{x},t)(t-\theta)}\,d\theta\,dS. \tag{3.22}$$

Substituting Eqs. (3.14) and (3.19) into Eq. (3.7) and incorporating the inflow boundary condition (3.4) into Eq. (3.7), we obtain the reference equation

$$\begin{aligned}\int_{\Omega}\Phi(\mathbf{x},t_n)c(\mathbf{x},t_n)w(\mathbf{x},t_n)\,d\mathbf{x} &+ \int_{\Gamma_n^{(O)}}\mathbf{v}(\mathbf{x},t)\cdot\mathbf{n}(\mathbf{x})c(\mathbf{x},t)w(\mathbf{x},t)\,dS\\ &= \int_{\Omega}\Phi(\mathbf{x},t_{n-1})c(\mathbf{x},t_{n-1})w(\mathbf{x},t_{n-1}^{+})\,d\mathbf{x} + \int_{\Omega}\Psi^{(1)}(\mathbf{x},t_n)q(\mathbf{x},t_n)w(\mathbf{x},t_n)\,d\mathbf{x}\\ &\quad + \int_{\Gamma_n^{(O)}}\Psi^{(2)}(\mathbf{x},t)\mathbf{v}(\mathbf{x},t)\cdot\mathbf{n}(\mathbf{x})q(\mathbf{x},t)w(\mathbf{x},t)\,dS\\ &\quad - \int_{\Gamma_n^{(I)}}\mathbf{v}(\mathbf{x},t)\cdot\mathbf{n}(\mathbf{x})g(\mathbf{x},t)w(\mathbf{x},t)\,dS + E(w),\end{aligned} \tag{3.23}$$

where

$$E(w) := \int_{t_{n-1}}^{t_n}\int_{\Omega} c(\mathbf{x},t)(\Phi w_t + \mathbf{v}\cdot\nabla w - Rw)(\mathbf{x},t)\,d\mathbf{x}\,dt + E_1(q,w) + E_2(q,w). \tag{3.24}$$

4. ELLAM SCHEMES

4.1. *Trial Spaces*

Recalling Remark 3.1, in order to define the test functions w in Eq. (3.7), we only need to specify them on $\bar{\Omega}$ at time t_n and on the space-time outflow boundary $\Gamma_n^{(O)}$. We first define a regular (d-dimensional tetrahedron or rectangular) finite element spatial partition $\mathcal{T}_h$ on $\bar{\Omega}$, with h being the diameter of the partition, as in standard finite element methods. We then extend the partition into a regular partition on $\bar{\Omega}$ at time t_n and on the space-time outflow boundary $\Gamma_n^{(O)}$ and denote the extended partition by $\mathcal{T}_{h,\Delta t}$.

Remark 4.1. If the spatial nodes on $\bar{\Omega}$ at time t_{n-1} are tracked forward, the number of spatial degrees of freedom crossing the space-time 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. Hence, the partition $\mathcal{T}_{h,\Delta t}$ should satisfy this condition. This condition can also be justified from another point of view. The ELLAM schemes use characteristic tracking in temporal discretization on Ω, so they are not subject to the CFL restriction on Ω. However, at the outflow boundary $\Gamma_n^{(O)}$ the discretization is in the time direction. Therefore, it should obey the CFL restriction for the reason of stability.

Let $S^\kappa(\mathcal{T}_{h,\Delta t})$ be the space of continuous and piecewise polynomials of degree less than or equal to κ, defined on $\overline{\Omega}$ at time t_n and the space-time outflow boundary $\Gamma_n^{(O)}$ with the partition $\mathcal{T}_{h,\Delta t}$. We use $\overline{\Lambda}$ to denote the closure of a set Λ and $\mathcal{N}(\Lambda)$ to denote the set of all the nodes in the partition $\mathcal{T}_{h,\Delta t}$ that are also in Λ. We decompose the set $\mathcal{N}(\overline{\Omega \cup \Gamma_n^{(O)}})$ of all the nodes in the partition $\mathcal{T}_{h,\Delta t}$ as

$$\mathcal{N}\left(\overline{\Omega \cup \Gamma_n^{(O)}}\right) = \mathcal{N}_n^{(I)} \cup \mathcal{N}_n^{(O)} \cup \mathcal{N}_n^{\Omega} \cup \mathcal{N}_n^{\Gamma} \cup \mathcal{N}_{n-1}^{(O)}, \tag{4.1}$$

with

$$\begin{aligned}
\mathcal{N}_n^{(I)} &:= \mathcal{N}\left(\overline{\partial\Omega^{(I)}(t_n)}\right),\\
\mathcal{N}_n^{(O)} &:= \mathcal{N}\left(\overline{\partial\Omega^{(O)}(t_n)}\right),\\
\mathcal{N}_n^{\Omega} &:= \mathcal{N}\left(\overline{\Omega} - \overline{\partial\Omega^{(I)}(t_n)} - \overline{\partial\Omega^{(O)}(t_n)}\right),\\
\mathcal{N}_n^{\Gamma} &:= \mathcal{N}\left(\overline{\Gamma_n^{(O)}} - \overline{\partial\Omega^{(O)}(t_n)} - \overline{\partial\Omega^{(O)}(t_{n-1})}\right),\\
\mathcal{N}_{n-1}^{(O)} &:= \mathcal{N}\left(\overline{\partial\Omega^{(O)}(t_{n-1})}\right),
\end{aligned} \tag{4.2}$$

where

$$\begin{aligned}
\partial\Omega^{(I)}(t) &:= \{\mathbf{x} \mid \mathbf{x} \in \partial\Omega, \mathbf{v}(\mathbf{x},t)\cdot\mathbf{n}(\mathbf{x}) < 0\},\\
\partial\Omega^{(N)}(t) &:= \{\mathbf{x} \mid \mathbf{x} \in \partial\Omega, \mathbf{v}(\mathbf{x},t)\cdot\mathbf{n}(\mathbf{x}) = 0\},\\
\partial\Omega^{(O)}(t) &:= \{\mathbf{x} \mid \mathbf{x} \in \partial\Omega, \mathbf{v}(\mathbf{x},t)\cdot\mathbf{n}(\mathbf{x}) > 0\}
\end{aligned} \tag{4.3}$$

are the spatial inflow, noflow, and outflow boundaries at time t, respectively.

Remark 4.2. For any node $(\mathbf{x}_i, t_n) \in \mathcal{N}_n^{(I)} \cup \mathcal{N}_n^{\Omega}$, the corresponding basis functions $w_i = w_i(\mathbf{x}, t_n)$. For any node $(\mathbf{x}_i, t_{n-1}) \in \mathcal{N}_{n-1}^{(O)}$ or $(\mathbf{x}_i, t_i) \in \mathcal{N}_n^{\Gamma}$, the corresponding basis functions $w_i = w_i(\mathbf{x}, t)$ with $\mathbf{x} \in \partial\Omega$. Because $\overline{\partial\Omega^{(O)}(t_n)}$ is the intersection of Ω at time t_n and $\Gamma_n^{(O)}$, for any node in $\mathcal{N}_n^{(O)}$ the corresponding basis functions $w_i = w_i(\mathbf{x}, t_n)$ for the part in $\overline{\Omega}$ at time t_n and $w_i = w_i(\mathbf{x}, t)$ for the part in $\Gamma_n^{(O)}$.

The trial functions in the ELLAM schemes are chosen from the space $S^\kappa(\mathcal{T}_{h,\Delta t})$. Since $\mathcal{N}_n^{(I)}$ is the set of the nodes at the inflow boundary $\partial\Omega^{(I)}(t_n)$, the inflow boundary condition (3.4) is imposed. Hence, on $\overline{\Omega}$ at time t_n, the trial functions $C(\mathbf{x}, t_n)$ are of the form

$$C(\mathbf{x}, t_n) := \sum_{\mathbf{x}_i \in \mathcal{N}_n^{\Omega} \cup \mathcal{N}_n^{(O)}} C(\mathbf{x}_i, t_n) w_i(\mathbf{x}, t_n) + \sum_{\mathbf{x}_i \in \mathcal{N}_n^{(I)}} g(\mathbf{x}_i, t_n) w_i(\mathbf{x}, t_n), \qquad \mathbf{x} \in \Omega. \tag{4.4}$$

Because $\mathcal{N}_{n-1}^{(O)}$ is the set of the nodes in $\partial\Omega^{(O)}(t_{n-1})$, where the $C(\mathbf{x}, t_{n-1})$ are known from the solutions at the previous time step, on the boundary $\Gamma_n^{(O)}$ the trial functions $C(\mathbf{x}, t)$ are

$$\begin{aligned}
C(\mathbf{x}, t) := \sum_{(\mathbf{x}_i, t_i) \in \mathcal{N}_n^{\Gamma} \cup \mathcal{N}_n^{(O)}} C(\mathbf{x}_i, t_i) w_i(\mathbf{x}, t) + \sum_{\mathbf{x}_i \in \mathcal{N}_{n-1}^{(O)}} C(\mathbf{x}_i, t_{n-1}) w_i(\mathbf{x}, t),&\\
(\mathbf{x}, t) \in \Gamma_n^{(O)}.&
\end{aligned} \tag{4.5}$$

4.2. *Test Spaces and ELLAM Schemes*

Because the second terms on the right-hand sides of Eqs. (4.4) and (4.5) are already known, the degrees of freedoms in (4.4) and (4.5) are at the nodes in $\mathcal{N}_n^{\Omega} \cup \mathcal{N}_n^{(O)} \cup \mathcal{N}_n^{\Gamma}$, whose cardinality $N_E = |\mathcal{N}_n^{\Omega}| + |\mathcal{N}_n^{(O)}| + |\mathcal{N}_n^{\Gamma}|$ (where $|\Lambda|$ is the cardinality of a set Λ). If one follows conventional finite element methods, in the numerical schemes the weak formulation (3.23) (with the truncation error term $E(w)$ neglected) should hold for all the basis test functions w_i associated with the nodes in $\mathcal{N}_n^{\Omega} \cup \mathcal{N}_n^{(O)} \cup \mathcal{N}_n^{\Gamma}$. This closes the discrete system. However, to conserve mass all test functions should sum exactly to one on $\overline{\Omega}$ at time t_n and at the outflow boundary $\Gamma_n^{(O)}$ [12], which is violated by the aforementioned test functions. Instead, all the basis functions $w_i \in S^{\kappa}(\mathcal{T}_{h,\Delta t})$, whose cardinality is $N_A = |\mathcal{N}(\overline{\Omega \cup \Gamma_n^{(O)}})|$, satisfy this condition. But the number of test functions is $N_A - N_E = |\mathcal{N}_n^{(I)}| + |\mathcal{N}_{n-1}^{(O)}|$ more than the number of unknowns in (4.4) and (4.5), so the system is overspecified.

To overcome this difficulty, we add each equation of $\mathcal{N}_n^{(I)}$ or $\mathcal{N}_{n-1}^{(O)}$ to the equation at its adjacent node, or equivalently add each test function associated with a node in $\mathcal{N}_n^{(I)}$ or $\mathcal{N}_{n-1}^{(O)}$ to the test function at its adjacent node, within the same finite element cell which is in Ω or in $\overline{\Gamma_n^{(O)}} - \overline{\partial\Omega_{n-1}^{(O)}}$. In this way we obtain N_E number of basis functions for the test functions, which close the system and satisfy the above condition. We denote these functions by $\hat{w}_i$ $(i = 1, 2, \ldots, N_E)$ and the test space by

$$\hat{S}^{\kappa}(\mathcal{T}_{h,\Delta t}) := \operatorname{span}\{\hat{w}_i\}_{i=1}^{N_E}. \tag{4.6}$$

A family of ELLAM schemes. A family of ELLAM schemes (of different degrees κ) can be formulated as follows: Seek $C \in S^{\kappa}(\mathcal{T}_{h,\Delta t})$, which is of the form (4.4) and (4.5), such that for any $w \in \hat{S}^{\kappa}(\mathcal{T}_{h,\Delta t})$

$$\begin{aligned}
&\int_{\Omega} \Phi(\mathbf{x}, t_n) C(\mathbf{x}, t_n) w(\mathbf{x}, t_n)\, d\mathbf{x} + \int_{\Gamma_n^{(O)}} \mathbf{v}(\mathbf{x}, t) \cdot \mathbf{n}(\mathbf{x}) C(\mathbf{x}, t) w(\mathbf{x}, t)\, dS \\
&\quad = \int_{\Omega} \Phi(\mathbf{x}, t_{n-1}) C(\mathbf{x}, t_{n-1}) w(\mathbf{x}, t_{n-1}^{+})\, d\mathbf{x} + \int_{\Omega} \Psi^{(1)}(\mathbf{x}, t_n) q(\mathbf{x}, t_n) w(\mathbf{x}, t_n)\, d\mathbf{x} \\
&\qquad + \int_{\Gamma_n^{(O)}} \Psi^{(2)}(\mathbf{x}, t) \mathbf{v}(\mathbf{x}, t) \cdot \mathbf{n}(\mathbf{x}) q(\mathbf{x}, t) w(\mathbf{x}, t)\, dS \\
&\qquad - \int_{\Gamma_n^{(I)}} \mathbf{v}(\mathbf{x}, t) \cdot \mathbf{n}(\mathbf{x}) g(\mathbf{x}, t) w(\mathbf{x}, t)\, dS.
\end{aligned} \tag{4.7}$$

Remark 4.3. Unlike many Eulerian and characteristic methods that often require an artificial outflow boundary condition to be added, the ELLAM schemes (4.7) naturally incorporate the inflow boundary condition (3.4) into their formulations and provide a systematic way to treat the outflow boundary. Because all the test functions sum to one on $\overline{\Omega}$ at time t_n and at the outflow boundary $\Gamma_n^{(O)}$, dropping the last term on the left-hand side of Eq. (3.7) does not affect mass conservation [61]. Thus, the ELLAM schemes conserve mass.

Remark 4.4. Using characteristic tracking, the ELLAM schemes symmetrize the governing PDE, and generate a well-conditioned, symmetric, and positive definite coefficient matrix. Thus, the discrete system can be solved efficiently by, for example, the conjugate gradient method in an optimal order without any preconditioning needed. Moreover, the

ELLAM schemes eliminate the majority of the time truncation errors, so they allow large time steps to be used in a simulation without loss of accuracy.

Remark 4.5. The extension (3.11) of the test functions into the space-time strip $\bar{\Omega} \times [t_{n-1}, t_n]$ is needed only to derive the reference equation (3.23), based on which the ELLAM schemes (4.7) are developed. Nevertheless, the ELLAM schemes (4.7) only need the values of trial and test functions on $\overline{\Omega \cup \Gamma_n^{(O)}}$ and do not need the extension at all.

4.3. *A Concrete Example: An ELLAM Scheme on a Unit Square*

As a concrete example to the ELLAM schemes developed in subsections 4.1 and 4.2, we present an ELLAM scheme for solving problem (3.1) and (3.4) over a two-dimensional unit square $\Omega := (-0.5, 0.5) \times (-0.5, 0.5)$. For simplicity, we assume a uniform velocity field $\mathbf{v}(\mathbf{x}, t) = (V_1, 0)$ with V_1 being a positive constant, i.e., the velocity field is parallelto the x_1-axis.

We define a uniform partition on $[0, T]$ and a uniform rectangular partition $\mathcal{T}_h$ on Ω

$$\begin{aligned} t_n &:= n\Delta t, && n = 0, 1, \ldots, N_t,\ \Delta t := \frac{T}{N_t}, \\ x_1^i &:= -0.5 + i\Delta x_1, && i = 0, 1, \ldots, N,\ \ \Delta x_1 := \frac{1}{N}, \\ x_2^j &:= -0.5 + j\Delta x_2, && j = 0, 1, \ldots, N,\ \Delta x_2 := \frac{1}{N}. \end{aligned} \tag{4.8}$$

The space-time inflow, noflow, and outflow boundaries $\Gamma_n^{(I)}$, $\Gamma_n^{(N)}$, and $\Gamma_n^{(O)}$ (defined in (3.12)) can be identified by

$$\begin{aligned} \Gamma_n^{(I)} &= \{(x_1, x_2, t) \mid x_1 = -0.5, x_2 \in [-0.5, 0.5], t \in [t_{n-1}, t_n]\}, \\ \Gamma_n^{(N)} &= \{(x_1, x_2, t) \mid x_1 \in (-0.5, 0.5), x_2 = \pm 0.5, t \in [t_{n-1}, t_n]\}, \\ \Gamma_n^{(O)} &= \{(x_1, x_2, t) \mid x_1 = 0.5, x_2 \in [-0.5, 0.5], t \in [t_{n-1}, t_n]\}. \end{aligned} \tag{4.9}$$

In other words, the $\Gamma_n^{(I)}$ consists of the left face of the space-time strip $\bar{\Omega} \times [t_{n-1}, t_n] = [-0.5, 0.5] \times [-0.5, 0.5] \times [t_{n-1}, t_n]$, $\Gamma_n^{(O)}$ consists of the right face of the cube, and Γ_n^N consists of the front and back faces of the cube.

The Courant number $Cr^{(O)}$ at the outflow boundary $\Gamma_n^{(O)}$ is

$$Cr^{(O)} = \max_{(\mathbf{x},t)\in\Gamma_n^{(O)}} \frac{V_1 \Delta t}{\Delta x_1}. \tag{4.10}$$

Because the time step in the ELLAM schemes (4.7) is taken very large, the Courant number $Cr^{(O)}$ is typically much larger than one. Let $[Cr^{(O)}]$ be the integer part of the Courant number; we define a uniform local refinement in time at the outflow boundary $\Gamma_n^{(O)}$ (cf. Remark 4.1)

$$t_{n,i} := t_n - i\Delta t_f, \qquad i = 0, 1, \ldots, N_f, \text{ with } \Delta t_f := \frac{\Delta t}{N_f}, \tag{4.11}$$

where $N_f = [Cr^{(O)}]$ if $Cr^{(O)}$ is an integer and $N_f = [Cr^{(O)}] + 1$ otherwise. In this way, the last equation in (4.8) and Eq. (4.11) define a partition on the right face $\Gamma_n^{(O)}$.

The above partition and the uniform rectangular partition $\mathcal{T}_h$ (given by the last two equations in (4.8)) define the partition $\mathcal{T}_{h,\Delta t}$ over $\overline{\Omega \cup \Gamma_n^{(O)}}$. $\mathcal{N}(\overline{\Omega \cup \Gamma_n^{(O)}})$ is reduced to

$$\begin{aligned} \mathcal{N}\big(\overline{\Omega \cup \Gamma_n^{(O)}}\big) = {} & \big\{\big(x_1^i, x_2^j, t_n\big) \,\big|\, i, j = 0, 1, 2, \ldots, N\big\} \\ & \cup \big\{\big(0.5, x_2^j, t_n^i\big) \,\big|\, j = 0, 1, 2, \ldots, N, i = 1, 2, \ldots, N_f\big\}. \end{aligned} \tag{4.12}$$

It is easy to see that its cardinality $|\mathcal{N}(\overline{\Omega \cup \Gamma_n^{(O)}})| = (N + N_f + 1) * (N + 1)$.

The trial space $S^1(\mathcal{T}_{h,\Delta t})$ is the space of continuous and piecewise-bilinear functions defined on $\overline{\Omega}$ at time t_n and the space-time outflow boundary $\overline{\Gamma_n^{(O)}}$ with the nodes given in (4.12). Corresponding to (4.2) we have

$$\begin{aligned} \mathcal{N}_n^{(I)} &:= \mathcal{N}\big(\overline{\partial\Omega^{(I)}(t_n)}\big) = \big\{\big(-0.5, x_2^j, t_n\big) \,\big|\, j = 0, 1, 2, \ldots, N\big\}, \\ \mathcal{N}_n^{(O)} &:= \mathcal{N}\big(\overline{\partial\Omega^{(O)}(t_n)}\big) = \big\{\big(0.5, x_2^j, t_n\big) \,\big|\, j = 0, 1, 2, \ldots, N\big\}, \\ \mathcal{N}_n^{\Omega} &:= \mathcal{N}\big(\overline{\Omega} - \overline{\partial\Omega^{(I)}(t_n)} - \overline{\partial\Omega^{(O)}(t_n)}\big) \\ &= \big\{\big(x_1^i, x_2^j, t_n\big) \,\big|\, i = 1, 2, \ldots, N - 1, j = 0, 1, 2, \ldots, N\big\}, \\ \mathcal{N}_n^{\Gamma} &:= \mathcal{N}\big(\overline{\Gamma_n^{(O)}} - \overline{\partial\Omega^{(O)}(t_n)} - \overline{\partial\Omega^{(O)}(t_{n-1})}\big) \\ &= \big\{\big(0.5, x_2^j, t_n^i\big) \,\big|\, j = 0, 1, 2, \ldots, N, i = 1, 2, \ldots, N_f - 1\big\}, \\ \mathcal{N}_{n-1}^{(O)} &:= \mathcal{N}\big(\overline{\partial\Omega^{(O)}(t_{n-1})}\big) = \big\{\big(0.5, x_2^j, t_{n-1}\big) \,\big|\, j = 0, 1, 2, \ldots, N\big\}. \end{aligned} \tag{4.13}$$

Remark 4.6. At the nodes $(x_1^i, x_2^j, t_n) \in \mathcal{N}_n^{(I)} \cup \mathcal{N}_n^{\Omega}$, the basis functions $w_{i,j} = w_{i,j}(x_1, x_2)$. At the nodes $(0.5, x_2^j, t_n^i) \in \mathcal{N}_n^{\Gamma} \cup \mathcal{N}_{n-1}^{(O)} = \{(0.5, x_2^j, t_n^i) \mid j = 0, 1, 2, \ldots, N, i = 1, 2, \ldots, N_f\}$, the basis functions $w_{N+i,j} = w_{N+i,j}(x_2, t)$. Because $\mathcal{N}_n^{(O)}$ is the intersection of the domain Ω at time t_n and the space-time outflow boundary $\Gamma_n^{(O)}$, at the nodes $(0.5, x_2^j, t_n) \in \mathcal{N}_n^{(O)}$, the basis function $w_{N,j} = w_{N,j}(x_1, x_2)$ on Ω and $w_{N,j} = w_{N,j}(x_2, t)$ on $\Gamma_n^{(O)}$.

The inflow boundary condition (3.4) is imposed at the nodes in $\mathcal{N}_n^{(I)}$ while the C is known at the nodes in $\mathcal{N}_{n-1}^{(O)}$ from the solution at the previous time t_{n-1}. Therefore, the degrees of freedoms are at the nodes in

$$\begin{aligned} \mathcal{N}_n^{\Omega} \cup \mathcal{N}_n^{(O)} \cup \mathcal{N}_n^{\Gamma} = {} & \big\{\big(x_1^i, x_2^j, t_n\big) \,\big|\, i = 1, 2, \ldots, N, j = 0, 1, 2, \ldots, N\big\} \\ & \cup \big\{\big(0.5, x_2^j, t_n^i\big) \,\big|\, j = 0, 1, 2, \ldots, N, i = 1, 2, \ldots, N_f - 1\big\}, \end{aligned} \tag{4.14}$$

whose cardinality $N_E = |\mathcal{N}_n^{\Omega}| + |\mathcal{N}_n^{(O)}| + |\mathcal{N}_n^{\Gamma}| = (N + N_f - 1) * (N + 1)$.

To obtain $N_E = (N + N_f - 1) * (N + 1)$ basis functions for the test functions, which leads to a mass conservative scheme, we add the test functions $w_{0,j}(x_1, x_2)$ at each node $(-0.5, x_2^j)$ on the inflow boundary $\partial\Omega^{(I)}(t_n)$ to the corresponding test functions $w_{1,j}(x_1, x_2)$ at its adjacent node (x_1^1, x_2^j). We denote the resulting functions by $\hat{w}_{1,j}(x_1, x_2)$. These functions $\hat{w}_{1,j}(x_1, x_2) = \hat{w}_{1,j}(x_1^1, x_2)$, i.e., they are constant over the interval $[-0.5, x_1^1]$. We also add the test functions $\hat{w}_{N+N_f,j}(x_2, t)$ at each node $(0.5, x_2^j, t_{n-1})$ on the outflow boundary $\partial\Omega^{(O)}(t_{n-1})$ to the corresponding test functions $\hat{w}_{N+N_f-1,j}(x_2, t)$ at its adjacent node $(0.5, x_2^j, t_n^{N_f-1})$. Thus, the test functions $\hat{w}_{N+N_f-1,j}(x_2, t) = \hat{w}_{N+N_f-1,j}(x_2, t_n^{N_f-1})$, i.e., they are constant in the direction of time over the interval $[t_{n-1}, t_n^{N_f-1}]$ on the outflow boundary $\Gamma_n^{(O)}$. In this way, we obtain $N_E = (N + N_f - 1) * (N + 1)$ basis functions for the test functions, which sum exactly to one on Ω at time t_n and $\Gamma_n^{(O)}$ and lead to a mass conservative scheme,

$$\hat{S}^1(\mathcal{T}_{h,\Delta t}) = \operatorname{span}\bigl\{\{\hat{w}_{0,j}\}_{j=0}^{N}, \{w_{i,j}\}_{i=1,j=0}^{N+N_f-2,N}, \{\hat{w}_{N+N_f-1,j}\}_{j=0}^{N}\bigr\}. \tag{4.15}$$

The numerical scheme (4.7) can now be written as

$$\begin{aligned}
&\int_{-0.5}^{0.5}\int_{-0.5}^{0.5} \Phi(x_1, x_2, t_n)C(x_1, x_2, t_n)\hat{w}_{i,j}(x_1, x_2, t_n)\,dx_1\,dx_2 \\
&\quad + \int_{t_{n-1}}^{t_n}\int_{-0.5}^{0.5} V_1 C(0.5, x_2, t)\hat{w}_{i,j}(0.5, x_2, t)\,dx_2\,dt \\
&\;= \int_{-0.5}^{0.5}\int_{-0.5}^{0.5} \Phi(x_1, x_2, t_{n-1})C(x_1, x_2, t_{n-1})w(x_1, x_2, t_{n-1}^{+})\,dx_1\,dx_2 \\
&\quad + \int_{-0.5}^{0.5}\int_{-0.5}^{0.5} \Psi^{(1)}(x_1, x_2, t_n)q(x_1, x_2, t_n)w(x_1, x_2, t_n)\,dx_1\,dx_2 \\
&\quad + \int_{t_{n-1}}^{t_n}\int_{-0.5}^{0.5} \Psi^{(2)}(0.5, x_2, t)V_1 q(0.5, x_2, t)w(0.5, x_2, t)\,dx_2\,dt \\
&\quad + \int_{t_{n-1}}^{t_n}\int_{-0.5}^{0.5} V_1 g(-0.5, x_2, t)w(-0.5, x_2, t)\,dx_2\,dt.
\end{aligned} \tag{4.16}$$

5. IMPLEMENTATIONAL ISSUES

5.1. *Approximation of Characteristics and Test Functions*

To evaluate the first term on the right-hand side of Eq. (4.7), one needs to track the characteristics defined by the ordinary differential equation (3.9) and then use the expression (3.11) to calculate $w(\mathbf{x}, t_{n-1}^{+})$. However, Eq. (3.9) cannot be solved exactly for a general variable-velocity field, and numerical means have to be used. We can define an approximate characteristic $\mathbf{y} = \mathbf{r}_{Num}(\theta; \bar{\mathbf{x}}, \bar{t})$, which passes through $(\bar{\mathbf{x}}, \bar{t})$, either by an Euler formula for simplicity

$$\mathbf{r}_{Num}(\theta; \bar{\mathbf{x}}, \bar{t}) := \bar{\mathbf{x}} + \mathbf{v}(\bar{\mathbf{x}}, \bar{t})(\theta - \bar{t}), \tag{5.1}$$

or by a second-order Runge–Kutta (Heun's) method for better accuracy

$$\mathbf{r}_{Num}(\theta; \bar{\mathbf{x}}, \bar{t}) := \bar{\mathbf{x}} + \frac{(\theta - \bar{t})}{2}[\mathbf{v}(\bar{\mathbf{x}}, \bar{t}) + \mathbf{v}(\bar{\mathbf{x}} + (\theta - \bar{t})\mathbf{v}(\bar{\mathbf{x}}, \bar{t}), \theta)]. \tag{5.2}$$

Moreover, within a global time step $\Delta t := t_n - t_{n-1}$ for Eq. (4.7), we can also use a micro-time step

$$\Delta t_m := \frac{\Delta t}{N_m} \tag{5.3}$$

(with N_m being a positive integer) to track characteristics defined by (5.1) or (5.2).

Remark 5.1. Because the approximate characteristics (5.1) or (5.2) do not satisfy the adjoint equation (3.8) exactly, the last term on the left-hand side of Eq. (3.7) does not vanish but is small. The authors previously proved an optimal-order error estimate for an ELLAM scheme (with this term dropped) for a one-dimensional analogue of Eq. (3.1) [30]. Moreover, dropping this term does not affect mass conservation because all the test functions sum to one on Ω at time t_n and on the space-time outflow boundary $\Gamma_n^{(O)}$ [61].

5.2. *Evaluation of Nonstandard Integrals and a Forward Tracking Algorithm*

In the ELLAM schemes (4.7), the trial functions $C(\mathbf{x}, t) \in S^\kappa(\mathcal{T}_{h,\Delta t})$ and the test functions $w(\mathbf{x}, t) \in \hat{S}^\kappa(\mathcal{T}_{h,\Delta t})$ are defined as standard piecewise polynomials on Ω at time t_n and the space-time outflow boundary $\Gamma_n^{(O)}$. Therefore, the two terms on the left-hand side and the second and third terms on the right-hand side of Eqs. (4.7) are standard in finite element methods and can be evaluated in a straightforward manner.

The first and last terms on the right-hand side of Eqs. (4.7) are due to the application of the Lagrangian coordinates and are non-conventional in any Eulerian methods. We take the first term as an example to address any potential problems and to describe the corresponding algorithm that overcomes these problems. In this term, the value of $C(\mathbf{x}, t_{n-1})$ is known from the solution at time t_{n-1}. However, keep in mind that (cf. Eq. (3.11))

$$w(\mathbf{x}, t_{n-1}^+) = w(\tilde{\mathbf{x}}, t_n) e^{-\int_{t_{n-1}}^{t_n} R^\Phi(\mathbf{r}_{Num}(\gamma; \mathbf{x}, t_{n-1}), \gamma)\, d\gamma}, \tag{5.4}$$

with proper modification when $\mathbf{r}_{Num}(\theta; \mathbf{x}, t_n)$ intersects the boundary $\partial\Omega$ during the time interval $[t_{n-1}, t_n]$. Here $\tilde{\mathbf{x}} := \mathbf{r}_{Num}(t_n; \mathbf{x}, t_{n-1})$ is the point at the head corresponding to $\mathbf{x}$ at the foot. The evaluation of this term can be potentially difficult and causes serious numerical artifacts in characteristic methods [3, 49].

In the modified method of characteristics [21, 24, 62] and some one-dimensional ELLAM schemes [12, 30, 60], this term was rewritten as an integral at time t_n, with the standard value of $w(\mathbf{x}, t_n)$ but backtracking to evaluate $C(\mathbf{x}^*, t_{n-1})$ where $\mathbf{x}^* := \mathbf{r}_{Num}(t_{n-1}; \mathbf{x}, t_n)$ is the point at the foot corresponding to $\mathbf{x}$ at the head. In 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 [3, 49]. However, the evaluation of this term becomes much more challenging for multiple dimensional problems due to the multi-dimensional deformation of each finite element cell on which the test functions are defined as the geometry backtracked from time t_n to time t_{n-1}. This requires mapping of points along the boundary of the cell and subsequent interpolation and mapping onto the fixed spatial grid at the previous time level t_{n-1}. Binning and Celia [6] used such a mapping in a two-dimensional ELLAM scheme that was computationally very intensive, especially when part or all of the cell being mapped intersects a space-time boundary Γ_n.

The most practical approach for evaluating this term is to use a forward tracking algorithm proposed by Russell and Trujillo [61] and was implemented by Healy and Russell [37] and some of the authors [72, 75] for one- and two-dimensional advection-diffusion PDEs. This would enforce the integration quadrature on each cell in Ω at t_{n-1} with respect to the fixed spatial grid $\mathcal{T}_h$ on which $\Phi(\mathbf{x}, t_{n-1})$ and $C(\mathbf{x}, t_{n-1})$ are defined; the difficult evaluation is the test function $w(\mathbf{x}, t_{n-1}^+)$ given by (5.4). Rather than backtracking the geometry and estimating the test functions by mapping the deformed geometry onto the fixed grid $\mathcal{T}_h$, the discrete quadrature points $\mathbf{x}_p$ chosen on each cell of the fixed grid $\mathcal{T}_h$ on Ω at t_{n-1} in a regular fashion can be forward-tracked to $\tilde{\mathbf{x}}_p := \mathbf{r}_{Num}(t_n; \mathbf{x}_p, t_{n-1})$ at time t_n. Then, we determine which test functions are nonzero at $\tilde{\mathbf{x}}_p$ at time 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. Because this forward tracking does not change the solution grid or the data structure, the algorithm does not suffer from the complication of distorted grids, which complicates many forward tracking algorithms.

6. DESCRIPTION OF SOME OTHER NUMERICAL METHODS

In this section, we briefly describe some well studied and widely used numerical methods for advection-reaction PDEs, including the upwind finite difference method, the Galerkin finite element method (Gal), the quadratic Petrov–Galerkin method (QPG) [4, 11, 14], the cubic Petrov–Galerkin method (CPG) [8, 76], the streamline diffusion finite element method (SDM) [9, 40], the monotonic upstream-centered scheme for conservation laws (MUSCL) [15, 70], and the Minmod scheme [36, 64]. In the next section, we carry out numerical experiments to observe the performance of the ELLAM schemes (4.7) and their comparisons with these methods. For simplicity of illustration, we assume $\Phi(\mathbf{x}, t) \equiv 1$ and present these methods for Eq. (3.1) or its nonconservative analogue

$$Lc := \frac{\partial c}{\partial t} + \mathbf{v}(\mathbf{x}, t) \cdot \nabla c(\mathbf{x}, t) + (R(\mathbf{x}, t) + \nabla \cdot \mathbf{v}(\mathbf{x}, t))c = q(\mathbf{x}, t), \qquad \mathbf{x} \in \Omega, t \in (0, T], \tag{6.1}$$

on a two-dimensional rectangular domain with a rectangular partition (4.8).

6.1. *The Upwind Finite Difference Method*

For any $\mathbf{x}_{i,j} := (x_1^i, x_2^j) \in \mathcal{N}_n^{\Omega} \cup \mathcal{N}_n^{(O)}$ with x_1^i and x_2^j being defined in (4.8), let $C_{i,j}^n := C(\mathbf{x}_{i,j}, t_n)$, $V_{i,j}^{1,n} := V_1(\mathbf{x}_{i,j}, t_n)$, $V_{i,j}^{2,n} := V_2(\mathbf{x}_{i,j}, t_n)$, and $\mathbf{v}_{i,j}^n := \mathbf{v}(\mathbf{x}_{i,j}, t_n)$. The upwind finite difference method (UFDM) for Eq. (6.1) can be formulated as

$$C_{i,j}^n = \left[1 - \Delta t\left(R_{i,j}^{n-1} + \nabla \cdot \mathbf{v}_{i,j}^{n-1}\right)\right] C_{i,j}^{n-1} + \frac{V_{i,j}^{1,n-1}\Delta t}{\Delta x_1} \Delta^{x_1} C_{i,j}^{n-1} + \frac{V_{i,j}^{2,n-1}\Delta t}{\Delta x_2} \Delta^{x_2} C_{i,j}^{n-1}, \tag{6.2}$$

where $C_{i,j}^n = g(\mathbf{x}_{i,j}, t_n)$ for $\mathbf{x}_{i,j} \in \mathcal{N}_n^{(I)}$ and

$$\begin{aligned} \Delta^{x_1} C_{i,j}^{n-1} &:= \begin{cases} C_{i,j}^{n-1} - C_{i-1,j}^{n-1}, & \text{if } V_{i,j}^{1,n-1} \geq 0, \\ C_{i+1,j}^{n-1} - C_{i,j}^{n-1}, & \text{otherwise,} \end{cases} \\ \Delta^{x_2} C_{i,j}^{n-1} &:= \begin{cases} C_{i,j}^{n-1} - C_{i,j-1}^{n-1}, & \text{if } V_{i,j}^{2,n-1} \geq 0, \\ C_{i,j+1}^{n-1} - C_{i,j}^{n-1}, & \text{otherwise.} \end{cases} \end{aligned} \tag{6.3}$$

Remark 6.1. The upwind finite difference method (UFDM) is one of the earliest methods designed to eliminate the non-physical oscillations present in standard finite difference methods (FDMs). To date, the UFDM is still the primary underlying scheme for many large-scale production simulators in petroleum reservoir simulation or in subsurface contaminant transport and remediation, partially due to the following reasons: (i) The UFDM is extremely stable and could generate solutions with correct qualitative physical trend for extremely complicated multiphase (e.g., gas, oil, water, and rock phases in reservoir simulation, or aqueous, non-aqueous, and rock phases in subsurface contaminant transport and remediation) and multicomponent (e.g., methane, ethane, propane, and n-butane in reservoir simulation, or brine and trace-species radionuclides in subsurface contaminant transport of hazardous nuclear waste, or organic contaminants and nutrients in bioremediation) fluid flows in porous media, even if strongly heterogeneous and deformable media and singular sources/sinks (wells) are present. (ii) Because of the extreme complexities and difficulties of these problems, robustness and stability of its numerical simulation have been emphasized very much in industrial applications. Unfortunately, since it is impossible to find the analytical solutions for these problems or to have some clear understanding on the quantative behavior of these solutions, the issue of accuracy of a simulation has been largely ignored. Consequently, although it is well known that the UFDM generates solutions with numerical dispersion qualitatively, not many people have actually realized how severe the numerical dispersion could be in the UFDM scheme. This is one of the most important reasons why the UFDM scheme is included in this paper.

6.2. *The Galerkin and Petrov–Galerkin Finite Element Methods*

Let $S^1(\mathcal{T}_h)$ be the trial function space that consists of continuous and piecewise bilinear functions on the partition $\mathcal{T}_h$ given by (4.8). Then the Galerkin finite element method (GAL), the quadratic Petrov–Galerkin finite element method (QPG) [4, 11, 14], and the cubic Petrov–Galerkin finite element method (CPG) [8, 76] for Eq. (3.1) can be formulated as follows: Find $C(\mathbf{x}, t_n) \in S^1(\mathcal{T}_h)$, which is of the form (4.4), such that

$$\begin{aligned}
&\int_\Omega [1 + \lambda \Delta t R(\mathbf{x}, t_n)] C(\mathbf{x}, t_n) w_{i,j}(\mathbf{x})\, d\mathbf{x} - \lambda \Delta t \int_\Omega \mathbf{v}(\mathbf{x}, t_n) C(\mathbf{x}, t_n) \cdot \nabla w_{i,j}(\mathbf{x})\, d\mathbf{x} \\
&\quad + \lambda \Delta t \int_{\partial\Omega^{(O)}} \mathbf{v}(\mathbf{x}, t_n) \cdot \mathbf{n}(\mathbf{x}) C(\mathbf{x}, t_n) w_{i,j}(\mathbf{x})\, ds \\
&\quad = \int_\Omega [1 - (1-\lambda)\Delta t] C(\mathbf{x}, t_{n-1}) w_{i,j}(\mathbf{x})\, d\mathbf{x} \\
&\quad + (1-\lambda)\Delta t \int_\Omega \mathbf{v}(\mathbf{x}, t_{n-1}) C(\mathbf{x}, t_{n-1}) \cdot \nabla w_{i,j}(\mathbf{x})\, d\mathbf{x} \qquad (6.4)\\
&\quad - (1-\lambda)\Delta t \int_{\partial\Omega^{(O)}} \mathbf{v}(\mathbf{x}, t_{n-1}) \cdot \mathbf{n}(\mathbf{x}) C(\mathbf{x}, t_{n-1}) w_{i,j}(\mathbf{x})\, ds \\
&\quad + \Delta t \int_\Omega [\lambda q(\mathbf{x}, t_n) + (1-\lambda) q(\mathbf{x}, t_{n-1})] w_{i,j}(\mathbf{x})\, d\mathbf{x} \\
&\quad - \Delta t \int_{\partial\Omega^{(I)}} [\lambda \mathbf{v}(\mathbf{x}, t_n) g(\mathbf{x}, t_n) + (1-\lambda) \mathbf{v}(\mathbf{x}, t_{n-1}) g(\mathbf{x}, t_{n-1})] \cdot \mathbf{n}(\mathbf{x}) w_{ij}(\mathbf{x})\, ds.
\end{aligned}$$

Here $\lambda \in [0, 1]$ is the weighting parameter between the time levels t_{n-1} and t_n. In particular, $\lambda = 0, 1$, and 0.5 yield the explicit-Euler, implicit-Euler, and the Crank–Nicolson temporal discretizations, respectively. The test functions are also in the tensor product form $w_{i,j}(\mathbf{x}) := w_i(x_1) w_j(x_2)$. In the GAL scheme, $w_i(x_1)$ and $w_j(x_2)$ are the standard one-dimensional hat functions. In the QPG scheme, $w_i(x_1)$ and $w_j(x_2)$ are constructed by adding an asymmetric perturbation to the original piecewise-linear hat functions

$$w_i(x) := \begin{cases} \dfrac{x - x^{i-1}}{\Delta x} + \dfrac{3(x - x^{i-1})(x^i - x)}{(\Delta x)^2}, & x \in [x^{i-1}, x^i], \\ \dfrac{x^{i+1} - x}{\Delta x} - \dfrac{3(x - x^i)(x^{i+1} - x)}{(\Delta x)^2}, & x \in [x^i, x^{i+1}], \\ 0, & \text{otherwise}. \end{cases} \qquad (6.5)$$

The CPG method was derived for the Crank–Nicolson time discretization. In the CPG method, the $w_i(x_1)$ and $w_j(x_2)$ are defined as the original piecewise linear hat functions with a symmetric cubic perturbation added to each nonzero piece

$$w_i(x) := \begin{cases} \dfrac{x - x^{i-1}}{\Delta x} + \gamma \dfrac{(x - x^{i-1})(x^i - x)(x^{i-1} + x^i - 2x)}{(\Delta x)^3}, & x \in [x^{i-1}, x^i], \\ \dfrac{x^{i+1} - x}{\Delta x} - \gamma \dfrac{(x - x^i)(x^{i+1} - x)(x^i + x^{i+1} - 2x)}{(\Delta x)^3}, & x \in [x^i, x^{i+1}], \\ 0, & \text{otherwise}. \end{cases} \tag{6.6}$$

Here $\gamma = 5Cr^2$ with $Cr = \frac{V\Delta t}{\Delta x}$ being the Courant number.

6.3. *The Streamline Diffusion Finite Element Method*

The streamline diffusion finite element method (SDM) can be formulated as follows: Find a piecewise-trilinear (linear in time) function $C(\mathbf{x}, t)$ on the space-time strip $\Omega \times [t_{n-1}, t_n]$, which is discontinuous in time at t_{n-1} and t_n and satisfies $C(\mathbf{x}, t)|_{\partial\Omega^{(I)}(t_n)} = g(\mathbf{x}, t)$, such that

$$\begin{aligned} &\int_{t_{n-1}}^{t_n} \int_\Omega [C_t(\mathbf{x}, t) + \mathbf{v} \cdot \nabla C(\mathbf{x}, t) + (\nabla \cdot \mathbf{v} + R)C(\mathbf{x}, t)] \\ &\qquad \times [w(\mathbf{x}, t) + \delta(w_t + \mathbf{v} \cdot \nabla w)(\mathbf{x}, t)]\, d\mathbf{x}\, dt + \int_\Omega C(\mathbf{x}, t_{n-1}^+) w(\mathbf{x}, t_{n-1}^+)\, d\mathbf{x} \\ &\quad = \int_{t_{n-1}}^{t_n} \int_\Omega q(\mathbf{x}, t)[w(\mathbf{x}, t) + \delta(w_t + \mathbf{v} \cdot \nabla w)(\mathbf{x}, t)]\, d\mathbf{x}\, dt \\ &\qquad + \int_\Omega C(\mathbf{x}, t_{n-1}^-) w(\mathbf{x}, t_{n-1}^+)\, d\mathbf{x}. \end{aligned} \tag{6.7}$$

Here $w(\mathbf{x}, t_{n-1}^+) := \lim_{t \to t_{n-1}^+} w(\mathbf{x}, t)$ and $w(\mathbf{x}, t_{n-1}^-) := \lim_{t \to t_{n-1}^-} w(\mathbf{x}, t)$, $C(\mathbf{x}, t_0^-) := c_0(\mathbf{x})$, and δ is typically chosen to be

$$\delta := \frac{K\sqrt{h^2 + (\Delta t)^2}}{\sqrt{1 + |\mathbf{v}|^2}}, \tag{6.8}$$

where h is the diameter of the space partition and Δt is the size of the time step.

Remark 6.3. The choice of δ has significant effects on the numerical solutions. If δ is chosen too small, the numerical solutions will exhibit oscillations. If δ is too big, the SDM method will damp the numerical solutions seriously. Unfortunately, an optimal choice of δ is not clear and is heavily problem-dependent. In the numerical experiments in the next section, we use the formula (6.8) for δ, which is a generally accepted choice but may not be best possible for a given problem. In the formula (6.8), the constant K is typically chosen to be 1 or 0.5, which will be used along with several others in the next section to indicate the general behavior.

6.4. *The High Resolution Methods (Minmod and MUSCL)*

High resolution methods are well suited for the solution of nonlinear hyperbolic conservation laws and resolve shock discontinuities in the solutions without excessive smearing or spurious oscillations. We consider two such schemes in this section: the monotone upstream-centered scheme for conservation laws (MUSCL), which was developed by van Leer [70], and the Minmod scheme, which was based on a generalization of the first and was developed by Harten *et al.* [36, 64]. These methods apply to the following nonlinear hyperbolic conservation law

$$Lc := \frac{\partial c}{\partial t} + \frac{\partial u(c)}{\partial x_1} + \frac{\partial v(c)}{\partial x_2} = 0, \qquad \mathbf{x} \in \Omega, t \in (0, T], \tag{6.9}$$

which is closed by the inflow boundary condition (3.4) and the initial condition (3.5). If $u(c)$ and $v(c)$ in Eq. (6.9) are chosen to be $u(c) = V_1 c$ and $v(c) = V_2 c$, then Eq. (6.9) is reduced to Eq. (3.1) with $\Phi = 1$ and $R = q = 0$.

In these methods, the unknowns are defined at the cell center $\mathbf{x}_{i-1/2,j-1/2} := (x_1^{i-1/2}, x_2^{j-1/2})$ of each cell $[x_1^{i-1}, x_1^i] \times [x_2^{j-1}, x_2^j]$ for $i = 1, \ldots, I$ and $j = 1, \ldots, J$. Let $C^n_{i-1/2,j-1/2} := C(\mathbf{x}_{i-1/2,j-1/2}, t_n)$. These two schemes can be uniformly formulated as

$$C^n_{i-\frac{1}{2},j-\frac{1}{2}} = C^{n-1}_{i-\frac{1}{2},j-\frac{1}{2}} - \frac{\Delta t}{\Delta x_1}\big(\tilde{u}_{i,j-\frac{1}{2}} - \tilde{u}_{i-1,j-\frac{1}{2}}\big) - \frac{\Delta t}{\Delta x_2}\big(\tilde{v}_{i-\frac{1}{2},j} - \tilde{v}_{i-\frac{1}{2},j-1}\big). \tag{6.10}$$

To define the numerical flux functions $\tilde{u}$ and $\tilde{v}$ in Eq. (6.10), we first define the left and right states of the solution

$$\begin{aligned} C^{L,n-1}_{i,j-\frac{1}{2}} &:= C^{n-1}_{i-\frac{1}{2},j-\frac{1}{2}} + \left(\frac{\Delta x_1}{2} - \frac{\Delta t}{2}\eta^+_{i-\frac{1}{2},j-\frac{1}{2}}\right)\delta_{x_1} C^{n-1}_{i-\frac{1}{2},j-\frac{1}{2}} - \frac{\Delta t}{2\Delta x_2}\big(\xi^{n-1}_{i-\frac{1}{2},j} - \xi^{n-1}_{i-\frac{1}{2},j-1}\big), \\ C^{R,n-1}_{i,j-\frac{1}{2}} &:= C^{n-1}_{i+\frac{1}{2},j-\frac{1}{2}} - \left(\frac{\Delta x_1}{2} + \frac{\Delta t}{2}\eta^-_{i+\frac{1}{2},j-\frac{1}{2}}\right)\delta_{x_1} C^{n-1}_{i+\frac{1}{2},j-\frac{1}{2}} - \frac{\Delta t}{2\Delta x_2}\big(\xi^{n-1}_{i+\frac{1}{2},j} - \xi^{n-1}_{i+\frac{1}{2},j-1}\big), \end{aligned} \tag{6.11}$$

where

$$\xi^{n-1}_{i-\frac{1}{2},j} := H^v\big(C^{L,n-1}_{i-\frac{1}{2},j}, C^{R,n-1}_{i-\frac{1}{2},j}\big), \tag{6.12}$$

with the Godunov flux $H^v(c_L, c_R)$ being defined by

$$H^v(c_L, c_R) := \begin{cases} \min\limits_{[c_L, c_R]} v(c), & \text{if } c_L \le c_R, \\ \max\limits_{[c_R, c_L]} v(c), & \text{otherwise}, \end{cases} \tag{6.13}$$

and

$$\begin{aligned} C^{L,n-1}_{i-\frac{1}{2},j} &:= C^{n-1}_{i-\frac{1}{2},j-\frac{1}{2}} + \frac{\Delta x_2}{2}\delta_{x_2} C^{n-1}_{i-\frac{1}{2},j-\frac{1}{2}}, \\ C^{R,n-1}_{i-\frac{1}{2},j} &:= C^{n-1}_{i-\frac{1}{2},j+\frac{1}{2}} - \frac{\Delta x_2}{2}\delta_{x_2} C^{n-1}_{i-\frac{1}{2},j+\frac{1}{2}}. \end{aligned} \tag{6.14}$$

The MUSCL and Minmod schemes differ in the choices of the left and right states η^+ and η^- in Eq. (6.11) as well as the slopes $\delta_x C^{n-1}_{i-1/2,j-1/2}$. In the Minmod scheme, the left and right states are defined by

$$\eta^+_{i-\frac{1}{2},j-\frac{1}{2}} = \eta^-_{i-\frac{1}{2},j-\frac{1}{2}} := V^{1,n-1}_{i-\frac{1}{2},j-\frac{1}{2}} := V_1(\mathbf{x}_{i-\frac{1}{2},j-\frac{1}{2}}, t_{n-1}), \tag{6.15}$$

while in the MUSCL method they are defined by

$$\eta^{+}_{i-\frac{1}{2},j-\frac{1}{2}} := \max\bigl\{0, V^{1,n-1}_{i-\frac{1}{2},j-\frac{1}{2}}\bigr\}, \qquad \text{and} \qquad \eta^{-}_{i-\frac{1}{2},j-\frac{1}{2}} := \min\bigl\{0, V^{1,n-1}_{i-\frac{1}{2},j-\frac{1}{2}}\bigr\}. \tag{6.16}$$

In the Minmod scheme, the numerical slope in the x_1-direction is defined by

$$\delta_{x_1} C^{n-1}_{i-\frac{1}{2},j-\frac{1}{2}} := \begin{cases} \Delta^{x_1}_{+} C^{n-1}_{i-\frac{1}{2},j-\frac{1}{2}}, & \text{if } \bigl|\Delta^{x_1}_{+} C^{n-1}_{i-\frac{1}{2},j-\frac{1}{2}}\bigr| \le \bigl|\Delta^{x_1}_{-} C^{n-1}_{i-\frac{1}{2},j-\frac{1}{2}}\bigr|, \\ \Delta^{x_1}_{-} C_{i-\frac{1}{2},j-\frac{1}{2}}, & \text{otherwise,} \end{cases} \tag{6.17}$$

where the difference operators are given by

$$\begin{aligned} \Delta^{x_1}_{-} C^{n-1}_{i-\frac{1}{2},j-\frac{1}{2}} &:= \begin{cases} 2\dfrac{C^{n-1}_{i-\frac{1}{2},j-\frac{1}{2}} - g^{n-1}_{i-1,j-\frac{1}{2}}}{\Delta x_1}, & i = 1, \\ \dfrac{C^{n-1}_{i-\frac{1}{2},j-\frac{1}{2}} - C^{n-1}_{i-3/2,j-\frac{1}{2}}}{\Delta x_1}, & i = 2, \ldots, I, \end{cases} \\ \Delta^{x_1}_{+} C^{n-1}_{i-\frac{1}{2},j-\frac{1}{2}} &:= \begin{cases} \dfrac{C^{n-1}_{i+\frac{1}{2},j-\frac{1}{2}} - C^{n-1}_{i-\frac{1}{2},j-\frac{1}{2}}}{\Delta x_1}, & i = 1, \ldots, I-1, \\ 2\dfrac{g^{n-1}_{i,j-\frac{1}{2}} - C^{n-1}_{i-\frac{1}{2},j-\frac{1}{2}}}{\Delta x_1}, & i = I. \end{cases} \end{aligned} \tag{6.18}$$

The numerical slope in the x_2-direction is given by symmetry.

In the MUSCL method, the numerical slopes are defined by

$$\delta_{x_1} C^{n-1}_{i-\frac{1}{2},j-\frac{1}{2}} := \min\bigl\{\Delta_{\text{lim}} C^{n-1}_{i-\frac{1}{2},j-\frac{1}{2}}, \bigl|\Delta_c C^{n-1}_{i-\frac{1}{2},j-\frac{1}{2}}\bigr|\bigr\} \times \operatorname{sgn}\bigl(\Delta_c C^{n-1}_{i-\frac{1}{2},j-\frac{1}{2}}\bigr), \tag{6.19}$$

where $\operatorname{sgn}(x) := 1$ for $x > 0$, $\operatorname{sgn}(x) := -1$ for $x < 0$, and $\operatorname{sgn}(0) := 0$, and Δ_{lim} is defined by

$$\begin{aligned} &\Delta_{\text{lim}} := \\ &\begin{cases} \alpha_i \min\bigl\{\bigl|\Delta^{x_1}_{+} C^{n-1}_{i-\frac{1}{2},j-\frac{1}{2}}\bigr|, \bigl|\Delta^{x_1}_{-} C^{n-1}_{i-\frac{1}{2},j-\frac{1}{2}}\bigr|\bigr\}, & \text{if } \Delta^{x_1}_{+} C^{n-1}_{i-\frac{1}{2},j-\frac{1}{2}} \times \Delta^{x_1}_{-} C^{n-1}_{i-\frac{1}{2},j-\frac{1}{2}} > 0, \\ 0, & \text{otherwise,} \end{cases} \end{aligned} \tag{6.20}$$

and

$$\Delta_c C^{n-1}_{i-\frac{1}{2},j-\frac{1}{2}} := \begin{cases} \dfrac{C^{n-1}_{i+\frac{1}{2},j-\frac{1}{2}} + 3C^{n-1}_{i-\frac{1}{2},j-\frac{1}{2}} - 4g^{n-1}_{i-1,j-\frac{1}{2}}}{3\Delta x_1}, & i = 1, \\ \dfrac{C^{n-1}_{i+\frac{1}{2},j-\frac{1}{2}} - C^{n-1}_{i-3/2,j-\frac{1}{2}}}{2\Delta x_1}, & I = 2, \ldots, I-1, \\ \dfrac{4g^{n-1}_{i,j-\frac{1}{2}} - 3C^{n-1}_{i-\frac{1}{2},j-\frac{1}{2}} - C^{n-1}_{i-3/2,j-\frac{1}{2}}}{3\Delta x_1}, & i = I. \end{cases} \tag{6.21}$$

The parameter α_i in Eq. (6.20) is 2 for $i = 1, \ldots, I-1$ and 1, otherwise. It is the upper bound that allows the steeper representation of sharp fronts. The numerical flux $\tilde{u}_{i,j-\frac{1}{2}}$ is then defined by

$$\tilde{u}_{i,j-\frac{1}{2}} := H^u\big(C^{L,n-1}_{i,j-\frac{1}{2}}, C^{R,n-1}_{i,j-\frac{1}{2}}\big), \tag{6.22}$$

where the Godunov flux H^u is given in Eq. (6.13) with the flux function v being replaced by u. The numerical flux function $\tilde{v}$ is then defined by symmetry.

7. NUMERICAL RESULTS

In this section we present two-dimensional numerical experiments to investigate the performance of the ELLAM scheme (4.7) with piecewise-bilinear trial and test functions, and to compare it with the numerical methods described in Section 6.

7.1. *Model Problem: A Two-Dimensional Rotating Gaussian Pulse*

This example considers the transport of a two-dimensional Gaussian pulse. The spatial domain is $\Omega := (-0.5, 0.5) \times (-0.5, 0.5)$, a rotating velocity field is imposed as $V_1(x_1, x_2) = -4x_2$, and $V_2(x_1, x_2) = 4x_1$. The time interval is $[0, T] = [0, \pi/2]$, which is the time period required for one complete rotation. The initial condition $c_0(x_1, x_2)$ is given by

$$c_0(x_1, x_2) := \exp\left(-\frac{(x_1 - x_{1c})^2 + (x_2 - x_{2c})^2}{2\sigma^2}\right), \tag{7.1}$$

where x_{1c}, x_{2c}, and σ are the centered and standard deviations, respectively. The corresponding analytical solution for Eq. (3.1) with $\Phi = 1$ and $f = 0$ is given by

$$c(x_1, x_2, t) = \exp\left(-\frac{(\overline{x}_1 - x_{1c})^2 + (\overline{x}_2 - x_{2c})^2}{2\sigma^2} - \int_0^t R(\mathbf{r}(\theta; \overline{\mathbf{x}}, 0), \theta)\, d\theta\right), \tag{7.2}$$

where $\overline{x}_1 := x_1 \cos(4t) + x_2 \sin(4t)$, $\overline{x}_2 := -x_1 \sin(4t) + x_2 \cos(4t)$, and $\mathbf{r}(\theta; \overline{\mathbf{x}}, 0) := (\overline{x}_1 \cos(4\theta) - \overline{x}_2 \sin(4\theta), \overline{x}_1 \sin(4\theta) + \overline{x}_2 \cos(4\theta))$.

Remark 7.1. This problem provides an example for a homogeneous two-dimensional advection-reaction PDE with a variable velocity field and a known analytical solution and can be viewed as an incompressible fluid flow in a two-dimensional homogeneous porous medium. Consequently, this example has been used widely to test for numerical artifacts of different schemes, such as numerical stability, numerical dispersion, spurious oscillations, deformation, and phase errors as well as other numerical effects arising in porous medium fluid flows. The analytical solution $c(x_1, x_2, t)$ after one complete rotation is identical to the initial condition $c_0(x_1, x_2)$, which is centered at (x_{1c}, x_{2c}) with a minimum value 0 and a maximum value 1.

In the numerical experiments, the data are chosen as follows: $\Phi = 1$, $R = 0$, $f = 0$, $x_{1c} = -0.25$, $x_{2c} = 0$, $\sigma = 0.0447$ which gives $2\sigma^2 = 0.0040$. A uniform spatial grid of the form (4.8), where $h = \Delta x_1 = \Delta x_2 = \frac{1}{64}$ is chosen such that the analytical solution can be represented with a reasonable resolution, is used in the ELLAM scheme and used as a base spatial grid size in all the schemes in Section 6. Then the grid size h is further

TABLE I

The Performance of the ELLAM_Euler, the ELLAM_RK, and the Upwind Finite Difference (UFDM) Methods

	h	No. of cells	Δt	Δt_m	Max	Min	CPU	CFL No.	Fig. No.
Analytical	$\frac{1}{64}$	N/A	N/A	N/A	1	0	N/A	N/A	1a–1b
ELLAM_Euler	$\frac{1}{64}$	4,096	$\frac{\pi}{10}$	$\frac{\Delta t}{20}$	0.9308	0	31 s	57	—
	$\frac{1}{64}$	4,096	$\frac{\pi}{10}$	$\frac{\Delta t}{25}$	0.9367	0	33 s	57	—
ELLAM_RK	$\frac{1}{64}$	4,096	$\frac{\pi}{8}$	$\frac{\Delta t}{20}$	0.9987	0	1 min 5 s	71.25	1c–1d
UFDM	$\frac{1}{64}$	4,096	$\frac{\pi}{580}$	N/A	0.1491	0	1.7 s	0.98	—
	$\frac{1}{64}$	4,096	$\frac{\pi}{800}$	N/A	0.1387	0	2.4 s	0.71	—
	$\frac{1}{64}$	4,096	$\frac{\pi}{4000}$	N/A	0.1229	0	12 s	0.14	—
	$\frac{1}{64}$	4,096	$\frac{\pi}{6000}$	N/A	0.1218	0	18 s	0.09	—
	$\frac{1}{96}$	9,216	$\frac{\pi}{860}$	N/A	0.2092	0	6 s	0.99	—
	$\frac{1}{96}$	9,216	$\frac{\pi}{1000}$	N/A	0.2016	0	7 s	0.85	—
	$\frac{1}{96}$	9,216	$\frac{\pi}{2000}$	N/A	0.1831	0	14 s	0.43	—
	$\frac{1}{96}$	9,216	$\frac{\pi}{4000}$	N/A	0.1756	0	28 s	0.21	—
	$\frac{1}{192}$	36,864	$\frac{\pi}{1700}$	N/A	0.3475	0	52 s	1.0	—
	$\frac{1}{192}$	36,864	$\frac{\pi}{2400}$	N/A	0.3261	0	1 min 33 s	0.71	—
	$\frac{1}{256}$	65,536	$\frac{\pi}{2400}$	N/A	0.4105	0	2 min 18 s	0.95	—
	$\frac{1}{256}$	65,536	$\frac{\pi}{3000}$	N/A	0.3954	0	2 min 52 s	0.76	—
	$\frac{1}{384}$	147,456	$\frac{\pi}{3600}$	N/A	0.5109	0	9 min	0.95	—
	$\frac{1}{512}$	262,144	$\frac{\pi}{6000}$	N/A	0.5661	0	58 min	0.76	2a–2b
	$\frac{1}{1024}$	1,048,576	$\frac{\pi}{12000}$	N/A	0.7225	0	8 h 27 min	0.76	2c–2d

Note. Δt_m is the micro time step defined in (5.3).

refined for all the schemes in Section 6 if needed. We have systematically varied the time steps to examine the performance of each method, because the temporal errors dominate the numerical solutions with all the methods other than the ELLAM schemes. Except for the upwind finite difference method (UFDM), the MUSCL scheme, and the Minmod scheme that are explicit, all other comparative methods tested yield strongly non-symmetric 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 I, we present the minimum and maximum values of the numerical solutions with the ELLAM and the UFDM and the overall CPU each method consumed, which is measured on a SGI Indigo Workstation. We present the same results for the backward-Euler Galerkin and quadratic Petrov–Galerkin finite element methods, the Crank–Nicolson Galerkin and (quadratic and cubic) Petrov–Galerkin finite element methods, and the streamline diffusion finite element methods in Table II, and those for the MUSCL and Minmod schemes in Table III. 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 Tables I–III are presented in Figs. 1–12.

TABLE II
The Performance of the Galerkin, the Petrov–Galerkin, and the Streamline Diffusion Finite Element Methods

	h	No. of cells	Δt	Max	Min	CPU	CFL No.	Fig. No.
BE_GAL	$\frac{1}{64}$	4,096	$\frac{\pi}{200}$	0.2755	0	18 min 8 s	2.84	—
	$\frac{1}{64}$	4,096	$\frac{\pi}{800}$	0.4959	0	1 h 13 min	0.71	3a–3b
	$\frac{1}{96}$	9,216	$\frac{\pi}{800}$	0.4960	0	2 h 48 min	1.07	—
	$\frac{1}{64}$	4,096	$\frac{\pi}{2000}$	0.6697	0	3 h	0.28	—
	$\frac{1}{96}$	9,216	$\frac{\pi}{2000}$	0.6700	0	6 h 50 min	0.43	—
	$\frac{1}{64}$	4,096	$\frac{\pi}{4000}$	0.7864	0	6 h 3 min	0.14	—
	$\frac{1}{64}$	4,096	$\frac{\pi}{6000}$	0.8412	0	9 h 4 min	0.09	3c–3d
BE_QPG	$\frac{1}{64}$	4,096	$\frac{\pi}{200}$	0.2143	−0.0005	22 min 59 s	2.84	—
	$\frac{1}{64}$	4,096	$\frac{\pi}{800}$	0.3780	−0.0011	1 h 32 min	0.71	4a–4b
	$\frac{1}{96}$	9,216	$\frac{\pi}{800}$	0.4221	0	3 h 24 min	1.07	—
	$\frac{1}{64}$	4,096	$\frac{\pi}{2{,}000}$	0.4945	−0.0022	3 h 57 min	0.28	—
	$\frac{1}{96}$	9,216	$\frac{\pi}{2{,}000}$	0.5621	0	8 h 42 min	0.43	—
	$\frac{1}{64}$	4,096	$\frac{\pi}{4{,}000}$	0.5677	−0.0036	7 h 55 min	0.14	—
	$\frac{1}{64}$	4,096	$\frac{\pi}{6{,}000}$	0.6040	−0.0042	12 h 5 min	0.09	4c–4d
CN_GAL	$\frac{1}{64}$	4,096	$\frac{\pi}{200}$	0.8872	−0.2092	19 min 42 s	2.84	5a–5b
	$\frac{1}{96}$	9,216	$\frac{\pi}{200}$	0.8900	−0.2118	48 min 55 s	4.27	—
	$\frac{1}{64}$	4,096	$\frac{\pi}{400}$	0.9724	−0.0362	36 min 33 s	1.42	5c–5d
	$\frac{1}{96}$	9,216	$\frac{\pi}{400}$	0.9804	−0.0278	1 h 25 min	2.13	—
CN_QPG	$\frac{1}{64}$	4,096	$\frac{\pi}{200}$	0.6845	−0.1282	22 min 51 s	2.84	6a–6b
	$\frac{1}{96}$	9,216	$\frac{\pi}{200}$	0.7626	−0.1654	52 min 15 s	4.27	—
	$\frac{1}{64}$	4,096	$\frac{\pi}{400}$	0.7117	−0.0150	46 min 12 s	1.42	6c–6d
	$\frac{1}{96}$	9,216	$\frac{\pi}{400}$	0.8273	−0.0247	1 h 43 min	2.13	—
CN_CPG	$\frac{1}{64}$	4,096	$\frac{\pi}{200}$	Unbounded	Unbounded	—	2.84	—
	$\frac{1}{64}$	4,096	$\frac{\pi}{400}$	0.9824	−0.0004	53 min 36 s	1.42	7a–7b
	$\frac{1}{96}$	9,216	$\frac{\pi}{400}$	Unbounded	Unbounded	—	1.42	—
SDM, $K = 0.5$	$\frac{1}{64}$	4,096	$\frac{\pi}{200}$	0.7881	−0.0203	49 min 28 s	2.84	7c–7d
$K = 0.01$	$\frac{1}{64}$	4,096	$\frac{\pi}{200}$	0.9456	−0.0037	57 min 32 s	2.84	8a–8b
$K = 0.001$	$\frac{1}{64}$	4,096	$\frac{\pi}{200}$	0.9502	−0.0033	57 min 55 s	2.84	8c–8d

7.2. *The ELLAM Simulation*

The numerical results of ELLAM schemes are presented in Table I, where a global time step of Δt (given in (4.8)) is used in solving the ELLAM scheme (4.7). The ELLAM_Euler solutions and the ELLAM_RK solutions specify whether the Euler tracking (5.1) or the Runge–Kutta tracking (5.2) with a micro-time step of Δt_m given in (5.3) is used to track the characteristics. Even though the (global) time steps are very coarse, the resulting ELLAM solutions are much more accurate than the solutions with all the comparative methods with much finer spatial and temporal grid sizes and significantly increased CPU time and storage. The surface and contour plots for an ELLAM_RK solution are presented in Figs. 1c–1d.

TABLE III

The Performance of the Minmod and MUSCL Schemes

Scheme	h	No. of cells	Δt	Max	Min	CPU	CFL No.	Fig. No.
Minmod	$\frac{1}{64}$	4,096	$\frac{\pi}{580}$	0.4223	0	1 min 5 s	0.98	9a–9b
	$\frac{1}{64}$	4,096	$\frac{\pi}{800}$	0.4049	0	1 min 30 s	0.71	—
	$\frac{1}{64}$	4,096	$\frac{\pi}{4000}$	0.3745	0	7 min 31 s	0.14	—
	$\frac{1}{64}$	4,096	$\frac{\pi}{6000}$	0.3723	0	11 min 16 s	0.09	—
	$\frac{1}{96}$	9,216	$\frac{\pi}{860}$	0.5722	0	3 min 37 s	0.99	—
	$\frac{1}{96}$	9,216	$\frac{\pi}{1000}$	0.5629	0	4 min 12 s	0.85	—
	$\frac{1}{96}$	9,216	$\frac{\pi}{2000}$	0.5371	0	8 min 24 s	0.43	—
	$\frac{1}{96}$	9,216	$\frac{\pi}{4000}$	0.5257	0	16 min 48 s	0.21	—
	$\frac{1}{196}$	36,864	$\frac{\pi}{1700}$	0.7826	0	28 min 25 s	1.0	10a–10b
	$\frac{1}{196}$	36,864	$\frac{\pi}{2400}$	0.7702	0	40 min 7 s	0.71	—
	$\frac{1}{256}$	65,536	$\frac{\pi}{2400}$	0.8393	0	1 h 11 min	0.95	—
	$\frac{1}{256}$	65,536	$\frac{\pi}{3000}$	0.8332	0	1 h 29 min	0.76	—
	$\frac{1}{384}$	147,456	$\frac{\pi}{3600}$	0.8987	0	4 h 2 min	0.95	—
	$\frac{1}{512}$	262,144	$\frac{\pi}{6000}$	0.9250	0	11 h 58 min	0.76	—
	$\frac{1}{1024}$	1,048,576	$\frac{\pi}{12000}$	0.9681	0	3 days 23 h	0.76	11a–11b
MUSCL	$\frac{1}{64}$	4,096	$\frac{\pi}{580}$	0.6604	−0.0010	1 min 9 s	0.98	9c–9d
	$\frac{1}{64}$	4,096	$\frac{\pi}{800}$	0.6485	−0.0003	1 min 49 s	0.71	—
	$\frac{1}{64}$	4,096	$\frac{\pi}{4000}$	0.6268	0	9 min 6 s	0.14	—
	$\frac{1}{64}$	4,096	$\frac{\pi}{6000}$	0.6252	0	13 min 39 s	0.09	—
	$\frac{1}{96}$	9,216	$\frac{\pi}{860}$	0.8012	−0.0003	4 min 22 s	0.99	—
	$\frac{1}{96}$	9,216	$\frac{\pi}{1000}$	0.7965	−0.0002	5 min 5 s	0.85	—
	$\frac{1}{96}$	9,216	$\frac{\pi}{2000}$	0.7832	0	10 min 10 s	0.43	—
	$\frac{1}{96}$	9,216	$\frac{\pi}{4000}$	0.7780	0	20 min 21 s	0.21	—
	$\frac{1}{196}$	36,864	$\frac{\pi}{1700}$	0.9326	0	34 min 42 s	1.0	10c–10d
	$\frac{1}{196}$	36,864	$\frac{\pi}{2400}$	0.9291	0	49 min 1 s	0.71	—
	$\frac{1}{256}$	65,536	$\frac{\pi}{2400}$	0.9571	0	1 h 27 min	0.95	—
	$\frac{1}{256}$	65,536	$\frac{\pi}{3000}$	0.9556	0	1 h 49 min	0.76	—
	$\frac{1}{384}$	147,456	$\frac{\pi}{3600}$	0.9775	0	4 h 55 min	0.95	—
	$\frac{1}{512}$	262,144	$\frac{\pi}{6000}$	0.9851	0	14 h 39 min	0.76	—
	$\frac{1}{1024}$	1,048,576	$\frac{\pi}{12000}$	0.9948	0	4 days 21 h	0.76	11c–11d

Remark 7.2. The results in Table I and Figs. 1c–1d show that the ELLAM schemes generate accurate numerical solutions even if very large time steps are used, leading to significantly improved efficiency. The figures also show that the ELLAM solutions do not have deformation or phase errors.

7.3. *The Upwind Finite Difference Simulation*

As we discussed in Remark 6.1, the UFDM has been a primary underlying scheme for many large-scale production simulators in petroleum reservoir simulation and in subsurface

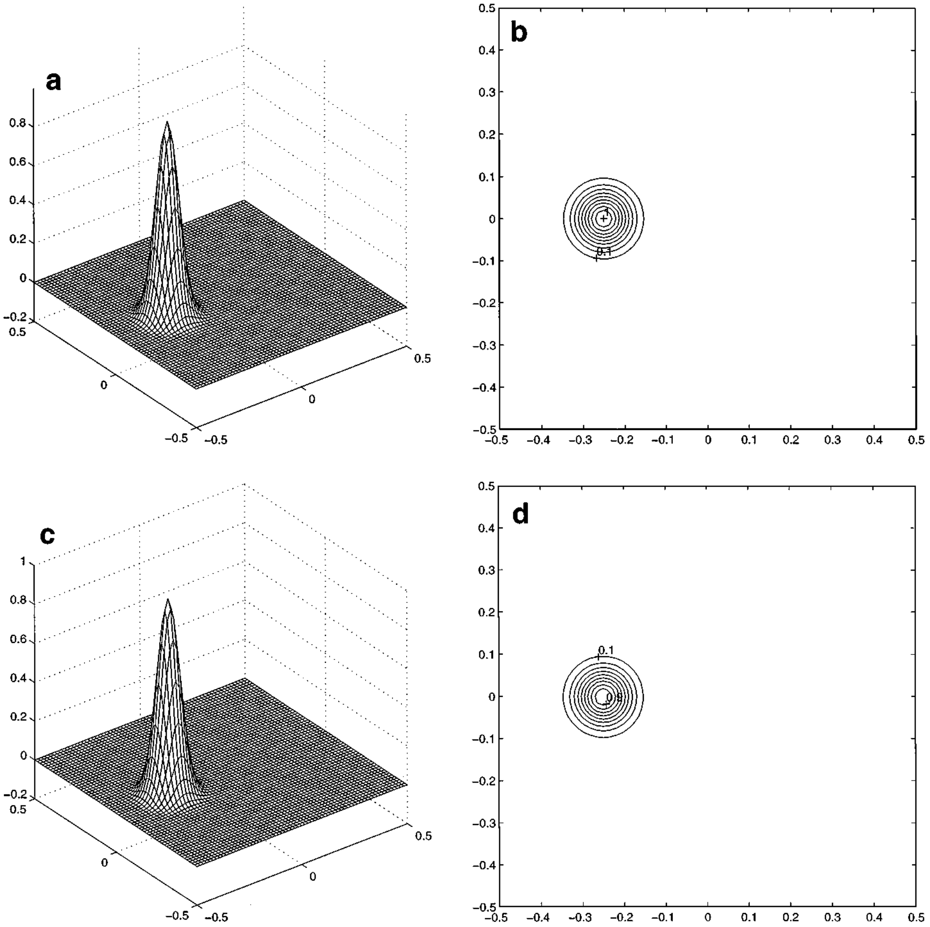

FIG. 1. The analytical solution and the ELLAM solution at $T = \frac{\pi}{2}$, $h = \frac{1}{64}$. (a) Analytical, min = 0, max = 1. (b) Analytical solution. (c) ELLAM_RK, min = 0, max = 0.9987. (d) $\Delta t = \frac{\pi}{8}$, $\Delta t_f = \frac{\Delta t}{20}$.

contaminant transport and remediation, because it is explicit and fairly easy to implement and can generate very stable solutions with correct qualitative physical trend even for very complex multiphase and multicomponent fluid flows in porous media. Although it is well known that the UFDM generates solutions with numerical dispersion, not many people have actually realized how severe the numerical dispersion could be quantatively in the UFDM. The experiments in this section serve to illustrate the quantative behavior of the UFDM.

With the base spatial grid size of $h = \frac{1}{64}$, the time step $\Delta t = \frac{\pi}{580}$ is the largest admissible step size that meets the CFL condition (the Courant number is 0.98). The UFDM scheme generates an extremely diffusive solution with a maximal value of only 0.1491, even though it is extremely efficient per time step (it took 1.7 s for 290 time steps). With a comparable CPU time which the ELLAM scheme consumed, the UFDM can generate a solution using a spatial grid size of $h = \frac{1}{192}$ (or equivalently, 36,864 elements) and a time step of $\Delta t = \frac{\pi}{1700}$. However, the resulting solution has a maximal value of only 0.3475. The finest grids used are $\Delta t = \frac{\pi}{12000}$, and $h = \frac{1}{1024}$ (i.e., 1,048,576 elements). It took CPU time of 8 h and 27 min for the UFDM to generate a solution with a maximal value of 0.7255, whose surface and contour plots are presented in Figs. 2c and 2d.

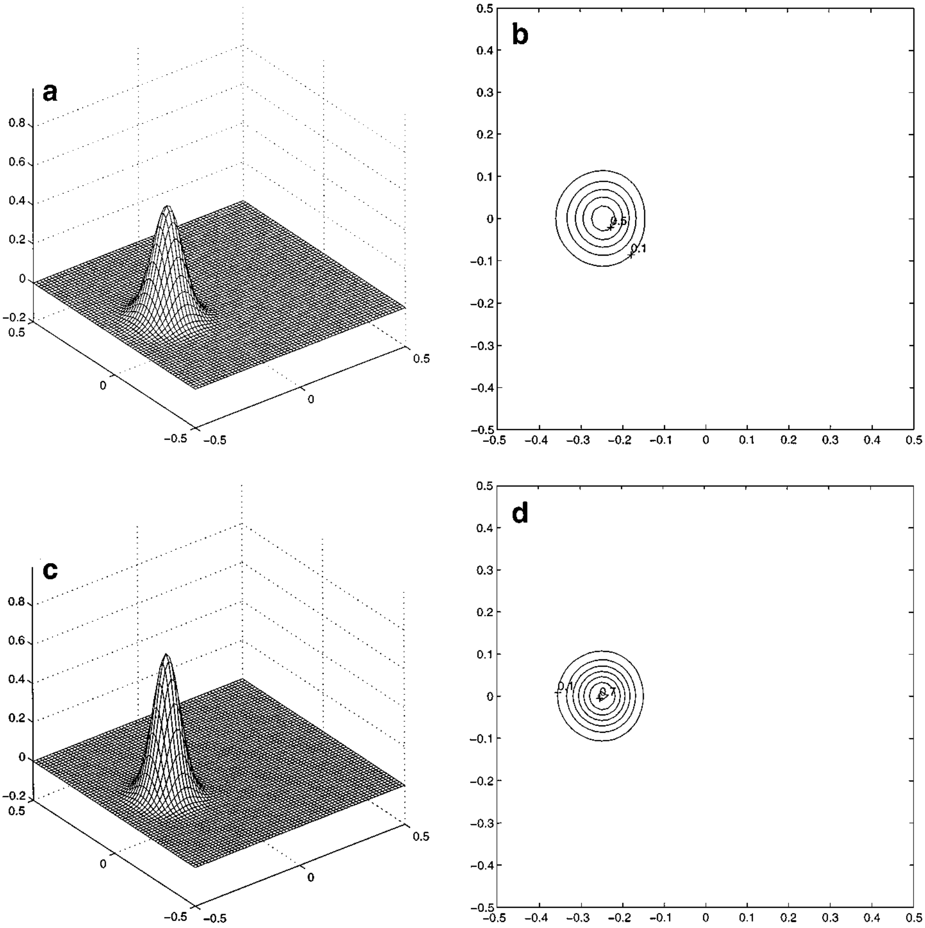

FIG. 2. The upwind finite difference solutions with different spatial and temporal grids at $T = \frac{\pi}{2}$. (a) min $= 0$, max $= 0.5661$. (b) $h = \frac{1}{512}$, $\Delta t = \frac{\pi}{6000}$. (c) min $= 0$, max $= 0.7225$. (d) $h = \frac{1}{1024}$, $\Delta t = \frac{\pi}{12000}$.

Remark 7.3. Although the UFDM is explicit and extremely efficient per time step, it requires an extremely large number of time steps; therefore, it takes a significant amount of overall CPU time to obtain a reasonably accurate solution. Furthermore, the UFDM needs an extremely refined grid, which means a significant increase of the computer memory. Finally, Figs. 2a–2d show that the UFDM generates solutions with no undershoots or oscillations, but with a slight deformation due to the grid orientation effect [24].

Remark 7.4. While the UFDM is subject to the CFL condition, the results in Table I show that it produces slightly more accurate solutions with larger time steps that satisfy the CFL condition. In contrast, the backward-Euler temporal discretization tends to generate more accurate solutions for smaller time steps. This phenomenon can be explained through the following one-dimensional analogue of problem (3.1)

$$\begin{aligned} &\frac{\partial c}{\partial t} + V\frac{\partial c}{\partial x} = 0, \qquad && x \in (-0.5, 0.5),\ t \in (0, T], \\ &c(-0.5, t) = g(t), && t \in [0, T], \end{aligned} \tag{7.3}$$

where V is a positive constant. Let the grid be defined by the first two equations in (4.8) with x_1 being replaced by x, and $C_n^i := C(x^i, t_n)$, the one-dimensional UFDM is

$$\begin{aligned} \frac{C_n^i - C_{n-1}^i}{\Delta t} + V\frac{C_{n-1}^i - C_{n-1}^{i-1}}{\Delta x} &= 0, \qquad 1 \le i \le N,\ 1 \le n \le N_t, \\ C_n^0 &= g(t_n), \qquad 0 \le n \le N_t. \end{aligned} \tag{7.4}$$

The local truncation error for the scheme (7.4) is

$$\begin{aligned} &\frac{c(x^i, t_n) - c(x^i, t_{n-1})}{\Delta t} + V\frac{c(x^i, t_{n-1}) - c(x^{i-1}, t_{n-1})}{\Delta x} \\ &= \left(\frac{\partial c}{\partial t} + V\frac{\partial c}{\partial x}\right)(x^i, t_{n-1}) + \frac{\Delta t}{2}\frac{\partial^2 c(x^i, t_{n-1})}{\partial t^2} \\ &\quad - \frac{V\Delta x}{2}\frac{\partial^2 c(x^i, t_{n-1})}{\partial x^2} + \mathcal{O}((\Delta x)^2 + (\Delta t)^2) \\ &= \frac{V\Delta x(Cr-1)}{2}\frac{\partial^2 c(x^i, t_{n-1})}{\partial x^2} + \mathcal{O}((\Delta x)^2 + (\Delta t)^2), \end{aligned} \tag{7.5}$$

where $Cr := \frac{V\Delta t}{\Delta x}$ is the Courant number. In the scheme (7.4), the temporal spatial errors cancel each other and a second-order local truncation error can be achieved with the Courant number equals to one. In fact, in this case the particle at the node x^{i-1} at time t_{n-1} meets the node x^i at time t_n. Hence, the scheme (7.4) becomes $C_n^i = C_{n-1}^{i-1}$ which is exact for the Courant number $Cr = 1$. In contrast, in a backward-in-time scheme the temporal error and the spatial error add up. This will be demonstrated in the next subsection.

7.4. *The BE_GAL and BE_QPG Simulation*

Because of their unconditional stability and robustness, fully implicit temporal discretizations have been widely used in many large-scale production simulators in industrial applications. Nevertheless, the accuracy issue has been traditionally overlooked. In this section, we conduct numerical experiments to investigate the performance of the backward-Euler temporal discretization, including the backward-Euler Galerkin finite element method (BE_GAL) and quadratic Petrov–Galerkin finite element method (BE_QPG). The numerical results are presented in Table II, with the surface and contour plots for selected example runs being given in Figs. 3a–4d.

With the base spatial grid size of $h = \frac{1}{64}$ and a time step of $\Delta t = \frac{\pi}{200}$ (which gives a Courant number of 2.84), the BE_GAL and BE_QPG methods generate solutions with minimal values of 0 and -0.0005 and maximal values of 0.2755 and 0.2143 (cf. Table II), respectively, which are excessively over-damped. Moreover, the BE_GAL and BE_QPG methods require more iterations in the PCGS solver than the ELLAM does, because they yield strongly non-symmetric coefficient matrices. The BE_GAL and BE_QPG solutions with a much finer time step of $\Delta t = \frac{\pi}{800}$ are presented in Figs. 3a–3b and 4a–4b, which show that these solutions are still very diffusive and deformed. They have minimal values of 0 and -0.0011, and maximal values of 0.4959 and 0.3780, respectively. 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 [24]. With the same time step of $\Delta t = \frac{\pi}{800}$ but reduced spatial grid size $h = \frac{1}{96}$, it takes significantly increased CPU times for the BE_GAL and the

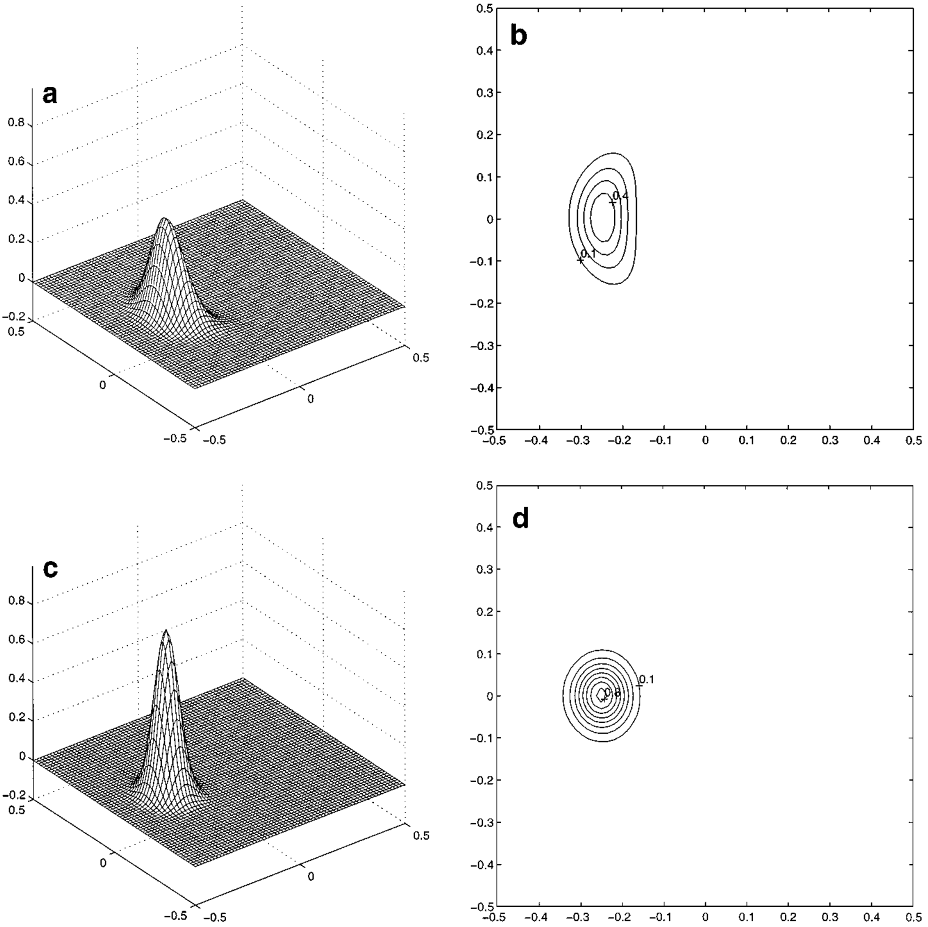

FIG. 3. The BE_GAL solutions with different time step sizes at $T = \frac{\pi}{2}$, $h = \frac{1}{64}$. (a) min = 0, max = 0.4959. (b) $\Delta t = \frac{\pi}{800}$. (c) min = 0, max = 0.8412. (d) $\Delta t = \frac{\pi}{6000}$.

BE_QPG methods to generate solutions with essentially no improvement. 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 $h = \frac{1}{64}$, leading to a Courant number of 0.28 and numerical solutions with more visible improvement.

With the same time step of $\Delta t = \frac{\pi}{2000}$ and a reduced spatial grid size of $h = \frac{1}{96}$, the BE_GAL and BE_QPG methods again consume significantly increased CPU time but generate solutions with essentially no improvement. Using less overall CPU time, we could use the base spatial grid of $h = \frac{1}{64}$ but a finer time step of $\Delta t = \frac{\pi}{4000}$, yielding solutions with more visible improvement. This shows that with $h = \frac{1}{64}$ and $\Delta t = \frac{\pi}{2000}$ that gives a Courant number of 0.28, the temporal error still dominates the BE_GAL and BE_QPG solutions. Finally, we use a spatial grid of $h = \frac{1}{64}$ and a time step of $\Delta t = \frac{\pi}{6000}$ and generate the BE_GAL and BE_QPG solutions with minimal values of 0 and −0.0042 and maximal values of 0.8412 and 0.6040, respectively. Their surface and contour plots are presented in Figs. 3c–3d and 4c–4d. The CPU times consumed by the BE_GAL and the BE_QPG solutions are 9 h and 4 min and 12 h and 5 min.

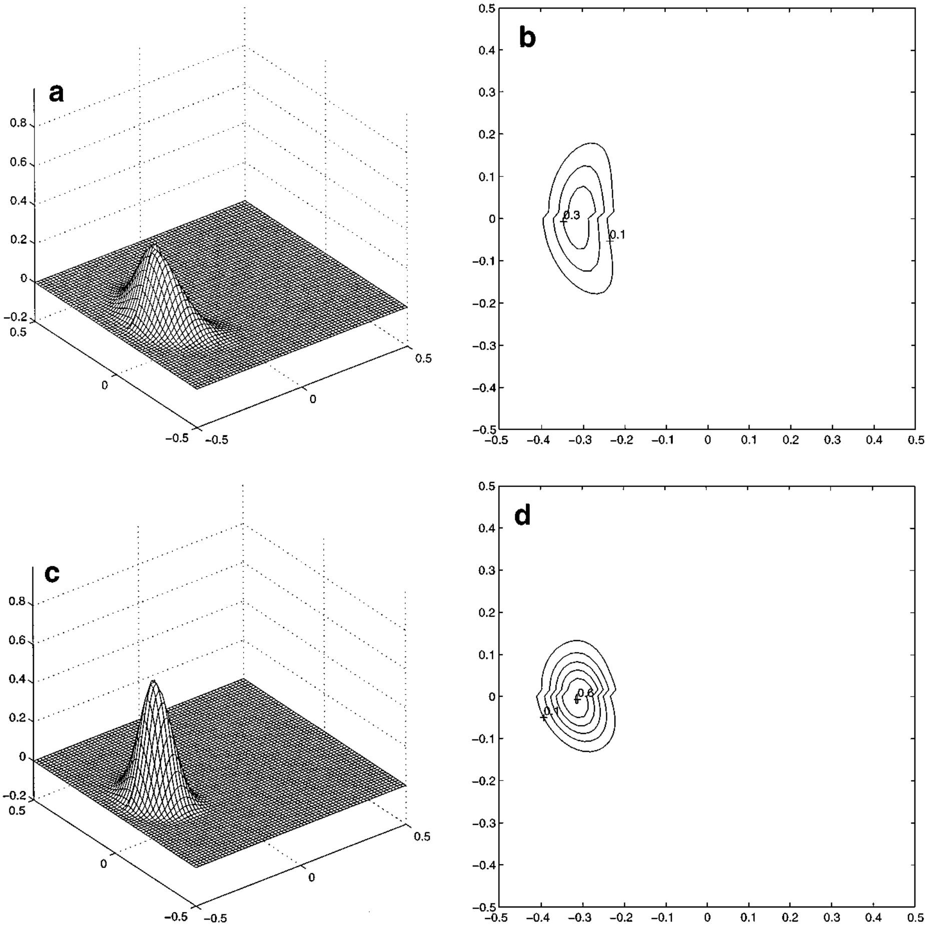

FIG. 4. The BE_QPG solutions with different time step sizes at $T = \frac{\pi}{2}$, $h = \frac{1}{64}$. (a) min $= -0.0011$, max $= 0.3780$. (b) $\Delta t = \frac{\pi}{800}$. (c) min $= -0.0042$, max $= 0.6040$. (d) $\Delta t = \frac{\pi}{6000}$.

Remark 7.5. With a time step of $\Delta t = \frac{\pi}{6000}$ (or equivalently, 9 h of CPU for the BE_GAL or 12 h of CPU for the BE_QPG), the BE_GAL and BE_QPG methods still cannot generate solutions that are comparable with the ELLAM solutions using $\Delta t = \frac{\pi}{10}$ (or about 0.5 min of CPU). In fact, the BE_QPG solution still has severe deformation. These results show that even though fully implicit methods are unconditionally stable and allow large time steps to be used in a simulation while maintaining its stability, extremely small time steps have to be used in these schemes, not for the reason of stability but for the reason of a reasonable accuracy. Consequently, this significantly reduces the efficiency of the simulation.

Remark 7.6. For a given spatial grid, the backward-Euler schemes produce more accurate numerical solutions with finer time steps. This is in contrast to the explicit upwind_FD scheme. For simplicity, we explain this for an implicit space-centered scheme, which can be viewed as an analogue of the BE_GAL scheme

$$\frac{C_n^i - C_{n-1}^i}{\Delta t} + V\frac{C_n^{i+1} - C_n^{i-1}}{2\Delta x} = 0, \qquad 1 \le i \le N-1,\ 1 \le n \le N_t, \tag{7.6}$$

or a backward, one-point upwind scheme, which can be viewed as an analogue of the BE_QPG scheme

$$\frac{C_n^i - C_{n-1}^i}{\Delta t} + V \frac{C_n^i - C_n^{i-1}}{\Delta x} = 0, \qquad 1 \le i \le N,\ 1 \le n \le N_t. \tag{7.7}$$

The local truncation error of the space-centered scheme (7.6) is

$$\begin{aligned} &\frac{c\big(x^i, t_n\big) - c\big(x^i, t_{n-1}\big)}{\Delta t} + V \frac{c\big(x^{i+1}, t_n\big) - c\big(x^{i-1}, t_n\big)}{2\Delta x} \\ &\quad = -\frac{\Delta t}{2} \frac{\partial^2 c\big(x^i, t_n\big)}{\partial t^2} + \mathcal{O}((\Delta x)^2 + (\Delta t)^2), \end{aligned} \tag{7.8}$$

while the local truncation error of the upwind scheme (7.7) is

$$\begin{aligned} &\frac{c\big(x^i, t_n\big) - c\big(x^i, t_{n-1}\big)}{\Delta t} + V \frac{c\big(x^i, t_n\big) - c\big(x^{i-1}, t_n\big)}{\Delta x} \\ &= \left(\frac{\partial c}{\partial t} + V \frac{\partial c}{\partial x}\right)\big(x^i, t_n\big) - \frac{\Delta t}{2} \frac{\partial^2 c\big(x^i, t_n\big)}{\partial t^2} - \frac{V \Delta x}{2} \frac{\partial^2 c\big(x^i, t_n\big)}{\partial x^2} + \mathcal{O}((\Delta x)^2 + (\Delta t)^2) \\ &= -\frac{V \Delta x (Cr + 1)}{2} \frac{\partial^2 c\big(x^i, t_n\big)}{\partial x^2} + \mathcal{O}((\Delta x)^2 + (\Delta t)^2). \end{aligned} \tag{7.9}$$

Unlike those in (7.5) that cancel with each other, the spatial and temporal errors in (7.8) and (7.9) add up. Therefore, with a reduced time step Δt, the local truncation error is reduced. Hence, the numerical solutions are more accurate with smaller time steps.

7.5. *The CN_GAL, CN_QPG, and CN_CPG Simulation*

In this subsection we investigate the performance of the Crank–Nicolson Galerkin (CN_GAL), quadratic Petrov–Galerkin (CN_QPG), and cubic Petrov–Galerkin Galerkin (CN_CPG) finite element methods. The results are presented in Table II, and the surface and contour plots for selected runs are given in Figs. 5a–7b.

With a spatial grid size of $h = \frac{1}{64}$ and a time step size of $\Delta t = \frac{\pi}{200}$, the CN_GAL solution has a minimal value of -0.2092 and a maximal value of 0.8872. Severe undershoot, deformation, and phase errors are observed in the plots for the solution in Figs. 5a–5b. The CN_QPG solution has a minimum value of -0.1282, a maximum value of 0.6845, and serious damping, phase error, and deformation. The CN_CPG method generates an unbounded solution for the given time step and spatial grid size that yields a Courant number of 2.84, since the CN_CPG method requires the Courant number not too far from 1. Next we keep the time step size but reduce the spatial grid size to $h = \frac{1}{96}$. The CN_GAL solution does not change much. The maximal value of the CN_QPG solution increases from 0.6845 to 0.7626, but the undershoot also increases from -0.1282 to -0.1654. Alternatively, we keep the spatial grid of $h = \frac{1}{64}$ but reduce the time step size to $\Delta t = \frac{\pi}{400}$. Both the CN_GAL and CN_QPG solutions are improved considerably. The CN_CPG solution is now available and has a maximal value of 0.9824 and a negligible undershoot of -0.0004.

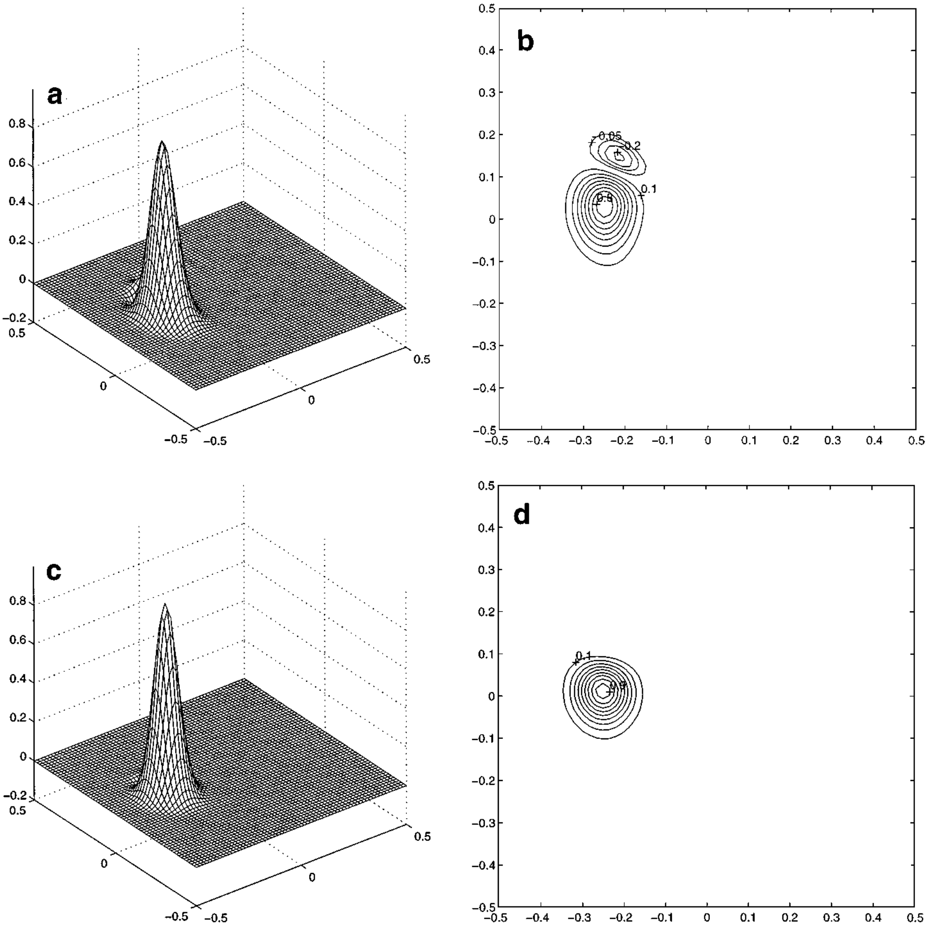

FIG. 5. The CN_GAL solutions with different time step sizes at $T = \frac{\pi}{2}$, $h = \frac{1}{64}$. (a) min = −0.2092, max = 0.8872. (b) $\Delta t = \frac{\pi}{200}$. (c) min = −0.0362, max = 0.9724. (d) $\Delta t = \frac{\pi}{400}$.

Remark 7.7. Because of their second-order accuracy in time, the CN_GAL, CN_QPG, and CN_CPG methods often generate more accurate solutions than the BE_GAL and BE_QPG methods do. However, the CN_GAL and CN_QPG methods yield solutions with severe undershoot and phase errors. The CN_CPG method generates more accurate solutions if the Courant number is around one but generates unbounded solutions for larger Courant numbers or solutions that are close to CN_GAL solutions for small Courant numbers. In any case, these methods are not compatible with the ELLAM solutions, in terms of accuracy, efficiency, and being free of deformation and phase errors.

7.6. *The SDM Simulation*

Table II also contains the numerical solutions of the streamline diffusion finite element method (SDM), whose surface and contour plots are presented in Figs. 7c–8d for a time step of $\Delta t = \frac{\pi}{200}$ and $h = \frac{1}{64}$. The undetermined parameters K in (6.8) are chosen to be 0.5, 0.1, and 0.001, respectively. As K decreases from 0.5 to 0.1 and then to 0.001, the maximal and minimal values of the corresponding SDM solutions change from 0.7881 and −0.0203

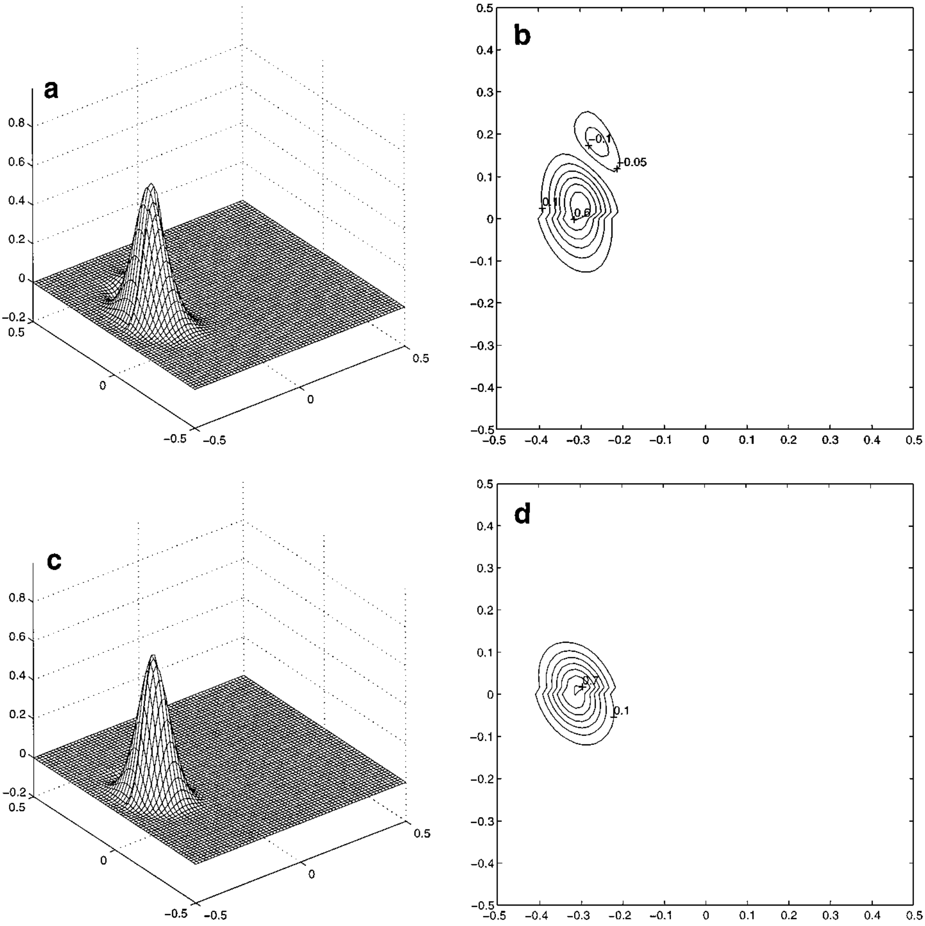

FIG. 6. The CN_QPG solutions with different time step sizes at $T = \frac{\pi}{2}$, $h = \frac{1}{64}$. (a) min $= -0.1282$, max $= 0.6845$. (b) $\Delta t = \frac{\pi}{200}$. (c) min $= -0.0150$, max $= 0.7117$. (d) $\Delta t = \frac{\pi}{400}$.

to 0.9456 and −0.0037, and then to 0.9502 and −0.0033. 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 it has 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.

Remark 7.8. The SDM can capture shock discontinuities in a thin region, but the numerical solutions may develop over- and under-shoots within this layer. A modified SDM with improved shock-capturing properties was proposed in [41, 45], which consists of adding a shock-capturing term to the diffusion by introducing a cross-wind control that is close to the steep fronts or shocks. This modified SDM scheme performs better in terms of catching steep fronts or jump discontinuities. However, it involves an additional undetermined parameter, and thus is not used in our comparison.

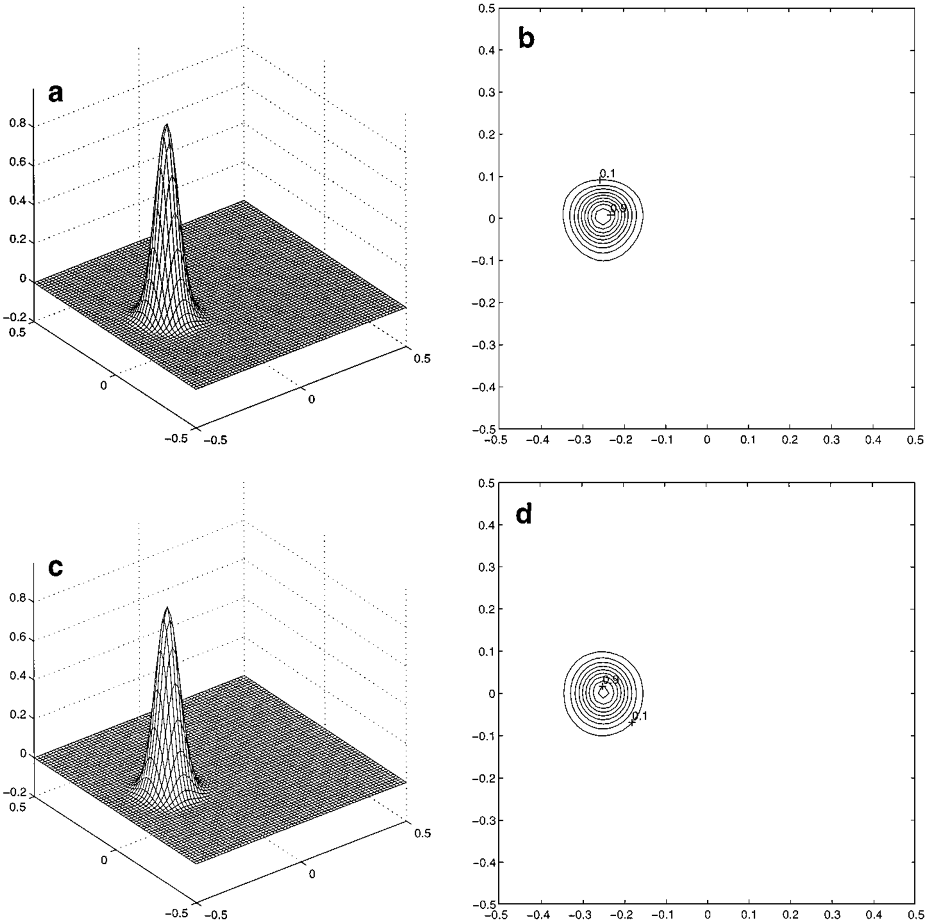

FIG. 7. The CN_CPG and SDM solutions at $T = \frac{\pi}{2}$, $h = \frac{1}{64}$. (a) CN_CPG, min $= -0.0004$, max $= 0.9824$. (b) CN_CPG, $\Delta t = \frac{\pi}{400}$. (c) SDM, min $= -0.0203$, max $= 1.7881$. (d) SDM, $K = 0.5$, $\Delta t = \frac{\pi}{200}$.

7.7. *The MUSCL and Minmod Simulations*

The numerical results of the MUSCL and Minmod schemes are presented in Table III. Since these methods are explicit, the same spatial and temporal grids are used in the numerical simulations as those used in UFDM to observe their performance. With the base spatial grid size of $h = \frac{1}{64}$ (i.e., 4,096 cells) and the time step size of $\Delta t = \frac{\pi}{580}$, the Minmod and MUSCL methods generate solutions with minimal values of 0 and -0.0010, and maximal values of 0.4223 and 0.6604, respectively. Their contour and surface plots are presented in Figs. 9a–9d. Using a CPU time of slightly more than one minute, they generate more accurate solutions than the UFDM with $h = \frac{1}{256}$ and $\Delta t = \frac{\pi}{2400}$ and the BE_GAL and BE_QPG solutions with more than 1 h of CPU time. With a spatial grid size of $h = \frac{1}{196}$ and a time step size of $\Delta t = \frac{\pi}{1700}$, the MUSCL and Minmod schemes already generate fairly accurate solutions, which are presented in Figs. 10a–10d. Finally, the MUSCL and Minmod solutions with a spatial grid size of $h = \frac{1}{1024}$ and a time step size of $\Delta t = \frac{\pi}{12000}$ are presented in Figs. 11a–11d.

Remark 7.9. The MUSCL and Minmod schemes generate more accurate solutions than the UFDM, Galerkin, and Petrov–Galerkin finite element methods with backward-Euler

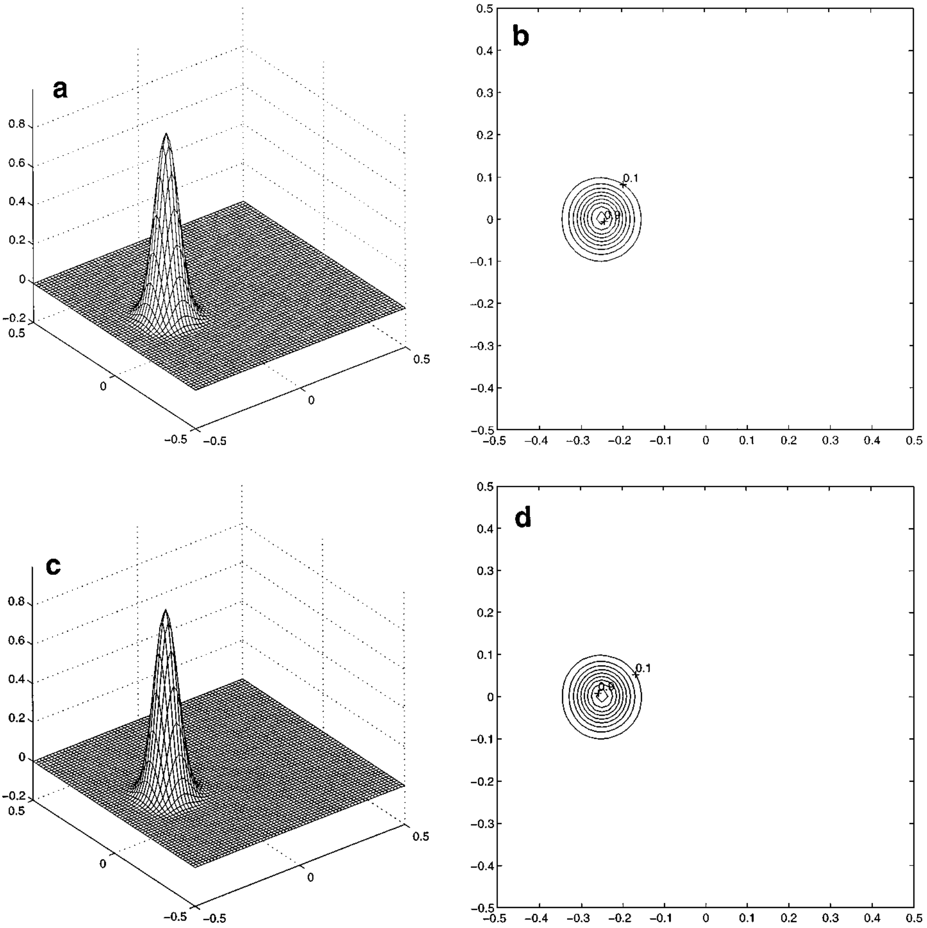

FIG. 8. The SDM solutions at $T = \frac{\pi}{2}$ with different values of K, $h = \frac{1}{64}$, and $\Delta t = \frac{\pi}{200}$. (a) $K = 0.01$, min $= -0.0037$, max $= 0.9456$. (b) $K = 0.01$. (c) $K = 0.001$, min $= -0.0033$, max $= 0.9502$. (d) $K = 0.001$.

or Crank–Nicolson temporal discretizations, and the streamline diffusion finite element methods. With the same spatial grid and temporal step sizes the MUSCL scheme generates more accurate solution than the Minmod scheme. However, the solutions generated by these methods are still not comparable with those generated by the ELLAM schemes that effectively discretize temporal derivatives and the advection term along characteristics.

Remark 7.10. There has widely been a misunderstanding that as long as the time step is chosen small enough to satisfy the CFL condition (or even smaller), an Eulerian method should generate a solution with the same accuracy as that produced by the ELLAM scheme with a possibly larger time step (but the same spatial grid size). This in turns implies that the ELLAM schemes will probably not improve computational efficiency much since reducing the size of time steps in (especially explicit) Eulerian methods will not increase computational cost much. However, the numerical results in this section show that this actually is *not* the case. The results in Tables I and III show that with the same spatial grid size of $h = \frac{1}{64}$, the solutions generated by the MUSCL and Minmod schemes with a (largest admissible) time step of $\Delta t = \frac{\pi}{580}$ are excessively diffusive and are not comparable with the

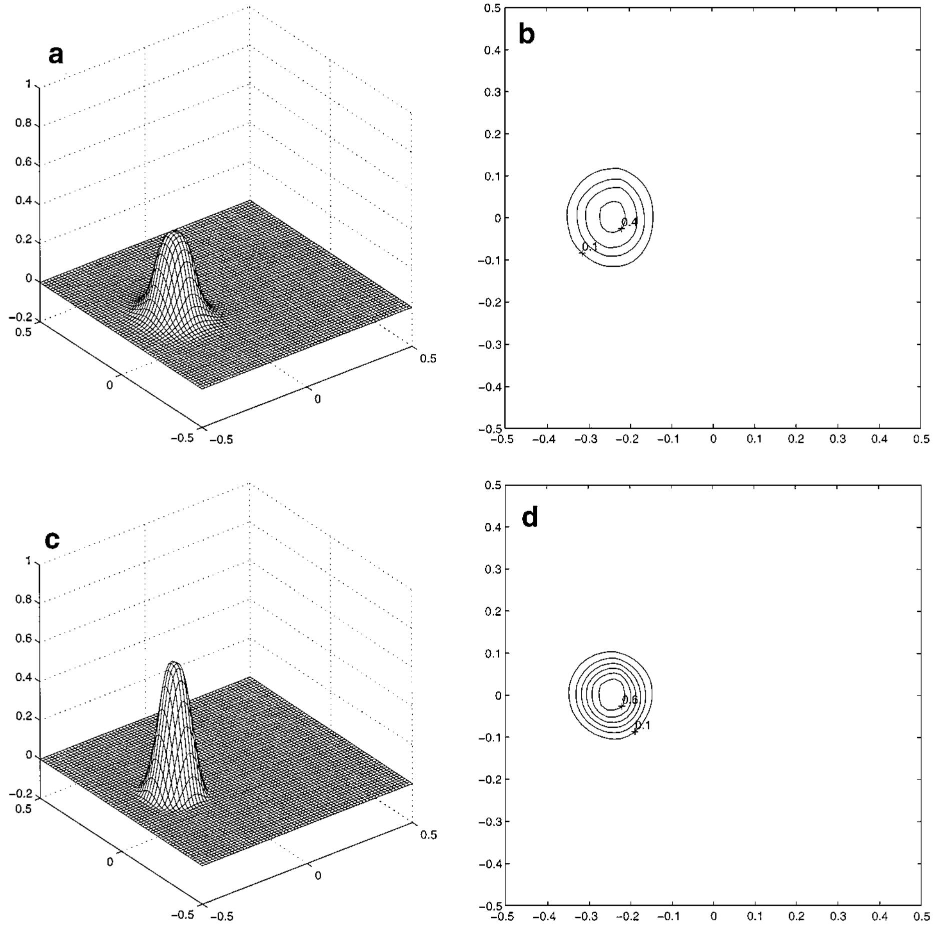

FIG. 9. The Minmod and MUSCL solutions with $h = \frac{1}{64}$ and $\Delta t = \frac{\pi}{580}$ at $T = \frac{\pi}{2}$. (a) min $= 0$, max $= 0.4223$. (b) The Minmod solution. (c) min $= -0.0010$, max $= 0.6604$. (d) The MUSCL solution.

solution generated by the ELLAM scheme with a time step of $\Delta t = \frac{\pi}{10}$ at all. Furthermore, if the time step is reduced in size, the MUSCL and Minmod schemes generate slightly less accurate solutions as we demonstrated for explicit methods in Remark 7.3. Therefore, to improve the accuracy of the numerical solutions generated by the MUSCL and Minmod schemes, one has to refine both the spatial grid and the time step. As a matter of fact, in order to generate solutions that are comparable to the ELLAM solution with $h = \frac{1}{64}$ and $\Delta t = \frac{\pi}{10}$, one has to significantly reduce the sizes of the spatial grids and temporal steps in the MUSCL and Minmod schemes from $h = \frac{1}{64}$ to $h = \frac{1}{1024}$ and the time step size from $\Delta t = \frac{\pi}{10}$ to $\Delta t = \frac{\pi}{12000}$. Consequently, this leads to a significantly increased cost of computational efficiency and memory storage.

7.8. *Additional Numerical Examples*

To observe the performance of the ELLAM scheme (4.7) in solving advection-reaction PDEs with non-homogeneous and/or discontinuous boundary conditions, we apply the ELLAM scheme to the following two model problems.

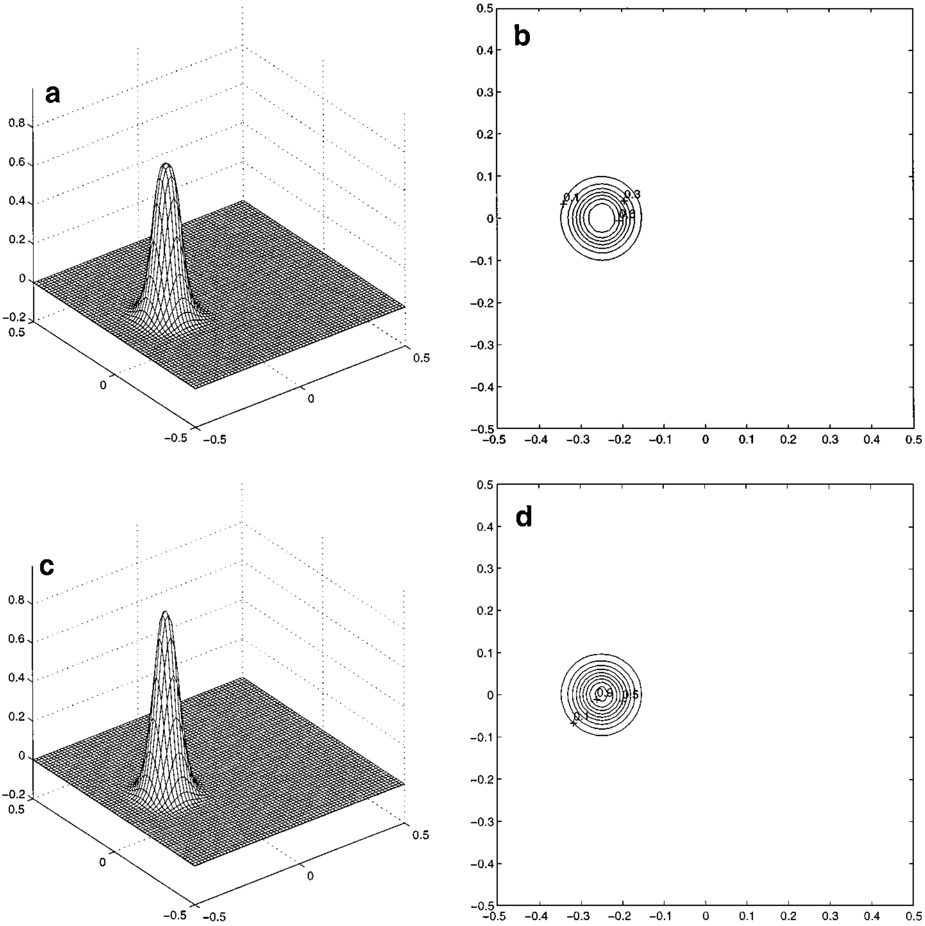

FIG. 10. The Minmod and MUSCL solutions with $h = \frac{1}{192}$ and $\Delta t = \frac{\pi}{1700}$ at $T = \frac{\pi}{2}$. (a) min = 0, max = 0.7826. (b) The Minmod solution. (c) min = 0, max = 0.9326. (d) The MUSCL solution.

Model Problem II. We consider the transport of a Gaussian pulse, which crosses over the boundary of the spatial domain $\Omega = (-0.5, 0.5) \times (-0.5, 0.5)$. The initial Gaussian pulse is still given by (7.1) with the same standard deviation $\sigma = 0.0447$ but different centered deviations $x_{1c} = x_{2c} = -0.5$. Hence, the initial Gaussian pulse is located at the left-bottom corner of the domain Ω. The rest of the Pulse that is not present is furnished via the inflow boundary condition. To obtain an analytical solution, we impose a velocity field of $V_1(x_1, x_2) = 1$ and $V_2(x_1, x_2) = 1$ and a time interval of $[0, T] = [0, 0.5]$. Thus, the transport is diagonal and terminates at the center of the domain Ω. A uniform spatial grid of $\Delta x_1 = \Delta x_2 = \frac{1}{64}$ and a time step of $\Delta t = \frac{1}{8}$ are chosen in the numerical simulation. An Euler method with no micro-time step tracking is used to evaluate the characteristics. The surface and contour plots are presented in Figs. 12a–12b, which show that the ELLAM scheme still generates accurate solutions even if a non-homogeneous boundary condition is present.

Model Problem III. To observe the performance of the ELLAM scheme (4.7) in solving advection-reaction PDEs with discontinuous boundary conditions and solutions, we consider the transport of a square pulse. A spatial domain of $\Omega = (-0.5, 0.5) \times (-0.5, 0.5)$, a time interval of $[0, T] = [0, 0.5]$, and a velocity field of $V_1(x_1, x_2) = 1$ and $V_2(x_1, x_2) = 0$

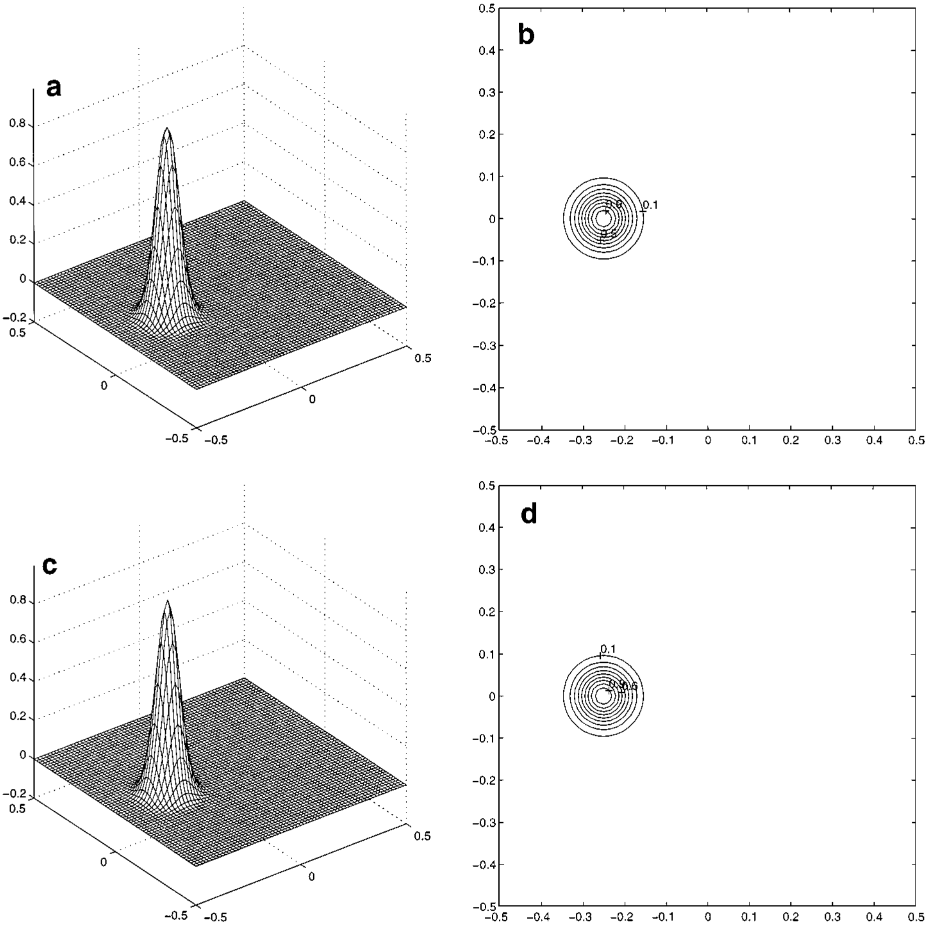

FIG. 11. The Minmod and MUSCL solutions with $h = \frac{1}{1024}$ and $\Delta t = \frac{\pi}{12000}$ at $T = \frac{\pi}{2}$. (a) min = 0, max = 0.9681. (b) The Minmod solution. (c) min = 0, max = 0.9948. (d) The MUSCL solution.

are used. A homogeneous initial condition is chosen. A discontinuous boundary condition is prescribed at the inflow boundary $x = -0.5$

$$g(y,t) = \begin{cases} 1, & \text{if } y \in (-0.25, 0.25), \\ 0, & \text{elsewhere.} \end{cases} \tag{7.10}$$

As in Model Problem II, a uniform spatial grid of $\Delta x_1 = \Delta x_2 = \frac{1}{64}$ and a time step of $\Delta t = \frac{1}{8}$ are chosen in the numerical simulation. An Euler method with no micro-time step tracking is used to evaluate the characteristics. The surface and contour plots are presented in Figs. 12c–12d, which show that the ELLAM scheme still generates satisfactory solutions even if a discontinuous boundary condition is prescribed.

8. SUMMARY AND DISCUSSIONS ON POSSIBLE EXTENSIONS

We develop a family of ELLAM schemes for first-order linear advection-reaction PDEs on general d-dimensional spatial domains. These schemes significantly reduce temporal truncation errors by using a forward characteristic tracking algorithm in evaluating their

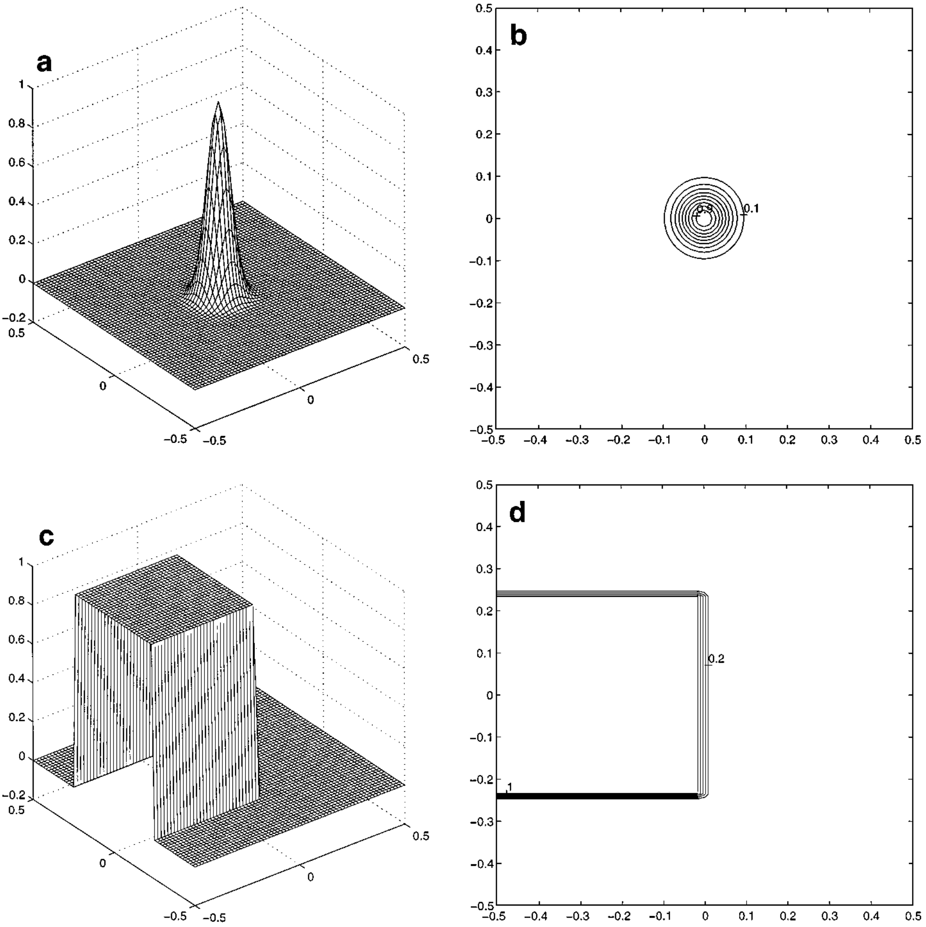

FIG. 12. The ELLAM solutions for Model Problems II and III with $h = \frac{1}{64}$ and $\Delta t = \frac{1}{8}$ at $T = 0.5$. (a) min $= 0$, max $= 0.9994$. (b)The ELLAM solution for Problem II. (c) min $= 0$, max $= 1$. (d) The ELLAM solution for Problem III.

right-hand sides and generate accurate numerical solutions even if large time steps are used. They naturally incorporate inflow boundary conditions into their formulations without any artificial outflow boundary conditions needed and conserve mass. Moreover, they yield well-conditioned, regularly structured, symmetric, and positive-definite coefficient matrices. Consequently, the resulting discrete algebraic systems can be solved efficiently by the conjugate gradient method in an optimal order number of operations without any preconditioning needed. Numerical results are presented to compare the performance of the ELLAM schemes with many well studied and widely used methods, including the upwind finite difference method, the Galerkin and the Petrov–Galerkin finite element methods with backward-Euler or Crank–Nicolson temporal discretization, the streamline diffusion finite element methods, the monotonic upstream-centered scheme for conservation laws (MUSCL), and the Minmod scheme. These results show that the ELLAM schemes often outperform these methods in the context of first-order linear advection-reaction PDEs. Of course, MUSCL and Minmod schemes are well suited for nonlinear hyperbolic conservation laws. Furthermore, Eulerian methods are easier to formulate and implement to characteristic methods.

In multiphase (e.g., immiscible) fluid flow processes in porous media, the corresponding governing equations are of the form of nonlinear hyperbolic conservation laws, which are similar to Eq. (1.2) but have a nonlinear advection term that is typically an S-shaped function of the unknown variable. Consequently, these equations could develop non-unique weak solutions [10, 47, 48], and characteristic methods do not apply directly. Previously, Espedal and Ewing [23] presented an operator-splitting technique to overcome this difficulty. The fractional flow function $f(S)$ is split into an advective concave hull $\bar{f}(S)$ of $f(S)$, which is linear in what would be the shock region of the governing PDE, and a residual anti-diffusive part. The modified advection PDE with the advection term given by $\bar{f}(S)$ yields the same entropy solution as the original PDE, and thus defines characteristic directions uniquely. The residual anti-diffusive advection term is treated numerically by a Petrov–Galerkin finite element method. This technique has been applied in numerical simulation of the modified method of characteristics for immiscible fluid flows [19]. Ewing [25] and Dahle *et al.* [20] have also applied the operator-splitting concept in developing an ELLAM scheme for one-dimensional immiscible fluid flow problems, which has shown very promising results. The authors are currently developing multi-dimensional ELLAM schemes for nonlinear advection-dominated PDEs.

When multiple components or species are involved in the fluid flow processes, many PDEs of form (1.2) are coupled together through the reaction terms. Previously, some of the authors and their collaborators generalized the ELLAM schemes for linear transport PDEs to subsurface contaminant transport with biodegradation or radionuclides and reservoir souring in one space dimension [32, 69, 74]. The numerical results showed their strong potential. The generalization of the ELLAM schemes developed in this paper to multi-dimensional transport systems (1.1)–(1.2) will be presented elsewhere.

Finally, we point out that in the system (1.1)–(1.2) the Darcy velocity $\mathbf{v}$ in the transport equation (1.2) is given as the numerical solution to the pressure equation (1.1). Consequently, the velocity field is given as piecewise polynomials at discrete time levels. Moreover, porous media could be strongly heterogeneous and deformable, the right-hand sources and sink terms are singular, and the pressure could be very high. All of these issues introduce further difficulties and complexities, which will be addressed in future publications.

REFERENCES

1. M. B. Allen III, G. A. Behie, and J. A. Trangenstein, *Multiphase Flow in Porous Media*, Lecture Notes in Engineering (Springer-Verlag, New York/Berlin, 1988), Vol. 34.
2. H. Aziz and A. Settari, *Petroleum Reservoir Simulation* (Applied Sci., Braking, Essex, 1979).
3. 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, 1987.
4. J. W. Barrett and K.W. Morton, Approximate symmetrization and Petrov–Galerkin methods for diffusion-convection problems, *Comput. Methods Appl. Mech. Eng.* **45**, 97 (1984).
5. J. Bear, *Hydraulics of Groundwater* (McGraw–Hill, New York, 1979).
6. 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 multiphase flow systems, *Water Resources Res.* **32**, 103 (1996).
7. R. C. Borden and P. B. Bedient, Transport of dissolved hydrocarbons influenced by oxygen-limited biodegradation. 1. Theoretical development, *Water Resources Res.* **22**, 1973 (1986).

8. 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. Eng.* **91**, 289 (1991).
9. 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. Eng.* **32**, 199 (1982).
10. S. E. Buckley and M. C. Leverret, Mechanism of fluid displacement in sands, *Trans. AIME* **146**, 107 (1942).
11. M. A. Celia, I. Herrera, E. T. Bouloutas, and J. S. Kindred, A new numerical approach for the advective-diffusive transport equation, *Numer. Methods PDEs* **5**, 203 (1989).
12. 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**, 187 (1990).
13. G. Chavent and J. Jaffré, *Mathematical Models and Finite Elements for Reservoir Simulation* (North-Holland, Amsterdam, 1986).
14. I. Christie, D. F. Griffiths, A. R. Mitchell, and O. C. Zienkiewicz, Finite element methods for second order differential equations with significant first derivatives, *Int. J. Numer. Eng.* **10**, 1389 (1976).
15. P. Colella, A direct Eulerian MUSCL scheme for gas dynamics, *SIAM J. Sci. Stat. Comp.* **6**, 104 (1985).
16. R. Courant, E. Isaacson, and M. Rees, On the solution of nonlinear hyperbolic differential equations by finite differences, *Comm. Pure Appl. Math.* **5**, 243 (1952).
17. R. A. Cox and T. Nishikawa, A new total variation diminishing scheme for the solution of advective-dominant solute transport, *Water Resources Res.* **27**, 2645 (1991).
18. M. G. Crandall and A. Majda, Monotone difference approximations for scalar conservation laws, *Math. Comp.* **34**, 1 (1980).
19. H. K. Dahle, M. S. Espedal, R. E. Ewing, and O. Sævareid, Characteristic adaptive sub-domain methods for reservoir flow problems, *Numer. Methods PDEs* **6**, 279 (1990).
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. Eng.* **122**, 223 (1995).
21. 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**, 871 (1982).
22. B. Einfeldt, On Godunov-type methods for gas dynamics, *SIAM J. Numer. Anal.* **25**, 294 (1988).
23. M. S. Espedal and R. E. Ewing, Characteristic Petrov–Galerkin subdomain methods for two-phase immiscible flow, *Comput. Methods Appl. Mech. Eng.* **64**, 113 (1987).
24. R. E. Ewing (Ed.), *The Mathematics of Reservoir Simulation*, Research Frontiers in Applied Mathematics (SIAM, Philadelphia, 1984), Vol. 1.
25. R. E. Ewing, Operator splitting and Eulerian–Lagrangian localized adjoint methods for multiphase flow, in *The Mathematics of Finite Elements and Applications*, edited by Whiteman (Academic Press, San Diego, CA, 1991), Vol. VII, p. 215.
26. R. E. Ewing and R. F. Heinemann, Mixed finite element approximation of phase velocities in compositional reservoir simulation, *Comput. Methods Appl. Mech. Eng.* **47**, 161 (1984).
27. 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**, 71 (1983).
28. R. E. Ewing, T. F. Russell, and M. F. Wheeler, Convergence analysis of an approximation of miscible displacement in porous media by mixed finite elements and a modified method of characteristics, *Comput. Methods Appl. Mech. Eng.* **47**, 73 (1984).
29. R. E. Ewing and H. Wang, Eulerian–Lagrangian localized adjoint methods for linear advection equations, in *Computational Mechanics 1991* (Springer-Verlag, New York/Berlin, 1991), p. 245.
30. R. E. Ewing and H. Wang, An optimal-order error estimate to Eulerian–Lagrangian localized adjoint method for variable-coefficient advection-reaction problems, *SIAM Numer. Anal.* **33**, 318 (1996).
31. R. E. Ewing, H. Wang, M. A. Celia, and R. Sharpley, A three-dimensional finite element simulation of nuclear waste contamination transport in porous media, in *Computer Methods and Advances in Geomechanics*, edited by Siriwardane and Zaman (Balkema, Rotterdam, 1995), Vol. 9, p. 2673.

32. R. E. Ewing, H. Wang, and R. C. Sharpley, Eulerian–Lagrangian localized adjoint methods for transport of nuclear-waste contamination in porous media, in *Computational Methods in Water Resources X*, Vol. 1, edited by Peters *et al.* Water Science and Technology Library (Kluwer Academic, Dordrecht, 1994), Vol. 12, p. 241.
33. R. E. Ewing, Y. Yuan, and G. Li, Time stepping along characteristics for a mixed finite element approximation for compressible flow of contamination by nuclear waste in porous media, *SIAM J. Numer. Anal.* **26**, 1513 (1989).
34. B. A. Finlayson, *Numerical Methods for Problems with Moving Fronts* (Ravenna Park Publishing, Seattle, 1992).
35. A. O. Garder, D. W. Peaceman, and A. L. Pozzi, Numerical calculations of multidimensional miscible displacement by the method of characteristics, *Soc. Pet. Eng. J.* **4**, 26 (1964).
36. A. Harten, B. Engquist, S. Osher, and S. Chakravarthy, Uniformly high order accurate essentially nonoscillatory schemes, III, *J. Comput. Phys.* **71**, 231 (1987).
37. 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**, 2399 (1993).
38. I. Herrera, R. E. Ewing, M. A. Celia, and T. F. Russell, Eulerian–Lagrangian localized adjoint methods: The theoretical framework, *Numer. Methods PDEs* **9**, 431 (1993).
39. D. Hillel, *Fundamentals of Soil Physics* (Academic Press, New York, 1980).
40. T. J. R. Hughes and A. N. Brooks, A multidimensional upwinding scheme with no crosswind diffusion, in *Finite Element Methods for Convection Dominated Flows*, edited by Hughes (ASME, New York, 1979), Vol. 34.
41. 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. Eng.* **58**, 305 (1986).
42. C. Johnson, U. Nävert, and J. Pitkäranta, Finite element methods for linear hyperbolic problems, *Comput. Methods Appl. Mech. Eng.* **45**, 285 (1984).
43. C. Johnson and J. Pitkäranta, An analysis of discontinuous Galerkin methods for a scalar hyperbolic equation, *Math. Comp.* **46**, 1 (1986).
44. C. Johnson and A. Szepessy, Adaptive finite element methods for conservation laws based on a posteriori error estimates, *Comm. Pure Appl. Math.* **98**, 199 (1995).
45. C. Johnson, A. Szepessy, and P. Hansbo, On the convergence of shock-capturing streamline diffusion finite element methods for hyperbolic conservation laws, *Math. Comp.* **54**, 107 (1990).
46. J. S. Kindred and M. A. Celia, Contaminant transport and biodegradation. II. Conceptual model and test simulations, *Water Resources Res.* **25**, 1149 (1989).
47. P. D. Lax, *Hyperbolic Systems of Conservation Laws and the Mathematical Theory of Shock Waves* (SIAM, Philadelphia, 1973).
48. R. J. LeVeque, *Numerical Methods for Conservation Laws* (Birkhäuser Verlag, Basel, 1992).
49. K. W. Morton, A. Priestley, and E. Süli, Stability of the Lagrangian–Galerkin method with nonexact integration, *RAIRO Model. Math. Anal. Numer.* **22**, 123 (1988).
50. S. P. Neuman, An Eulerian–Lagrangian numerical scheme for the dispersion-convection equation using conjugate space-time grids, *J. Comput. Phys.* **41**, 270 (1981).
51. D. W. Peaceman, *Fundamentals of Numerical Reservoir Simulation* (Elsevier, Amsterdam, 1977).
52. G. F. Pinder and H. H. Cooper, A numerical technique for calculating the transient position of the saltwater front, *Water Resources Res.* **6**, 875 (1970).
53. O. Pironneau, On the transport-diffusion algorithm and its application to the Navier–Stokes equations, *Numer. Math.* **38**, 309 (1982).
54. D.W. Pollock, Semianalytical computation of path lines for finite-difference models, *Ground Water* **26**, 743 (1988).
55. J. Pudykiewicz and A. Staniforth, Some properties and comparative performance of the semi-Lagrangian method of robert in the solution of the advection-diffusion equation, *Atmosphere-Ocean*. **22**, 283 (1984).
56. G. Qin, H. Wang, R. E. Ewing, and M. S. Espedal, Efficient numerical solution techniques for compositional model, in *Iterative Methods in Linear Algebra, II*, IMACS Series in Computational and Applied Mathematics, (Rutgers University, New Brunswick, NJ, 1996), p. 427.

57. M. Reeves and R. M. Cranwell, *User's Manual for the Sandia Waste-Isolation Flow and Transport Model (SWIFT) Release 4.81*, Sandia Report Nureg/CR-2324, SAND 81-2516, GR, 1981.
58. M. Reeves, D. S. Ward, N. D. Johns, and R. M. Cranwell, *Theory and Implementation for the Sandia Waste-Isolution Flow and Transport Model for Fractured Media (SWIFT) Release 4.84*, Sandia Report Nureg/CR-3328, SAND 83-1159, RW, 1986.
59. G. R. Richter, An optimal-order error estimate for the discontinuous Galerkin method, *Math. Comp.* **50**, 75 (1988).
60. T. F. Russell, Eulerian–Lagrangian localized adjoint methods for advection-dominated problems, in *Pitmann Research Notes in Mathematics Series*, edited by Griffiths and Watson (Longman, Harlow, 1990), Vol. 228, p. 206.
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*, edited by Gambolati, *et al.* (Springer-Verlag, Berlin, 1990), p. 357.
62. T. F. Russell and M. F. Wheeler, Finite element and finite difference methods for continuous flows in porous media, in *The Mathematics of Reservoir Simulation*, edited by Ewing, Research Frontiers in Applied Mathematics (SIAM, Philadelphia, 1984), Vol. 1, p. 35.
63. A. L. Schafer-Perini and J. L. Wilson, Efficient and accurate front tracking for two-dimensional groundwater flow models, *Water Resources Res.* **27**, 1471 (1991).
64. C. Shu and S. Osher, Efficient implementation of essentially non-oscillatory shock capturing schemes, *J. Comput. Phys.* **77**, 439 (1988).
65. P. K. Smolarkiewicz, The multidimensional crowley advection scheme, *Monthly Weather Rev.* **110**, 1968 (1982).
66. J. Smoller, *Shock Waves and Reaction-Diffusion Equations*, 2nd ed., Fundamental Principles of Mathematical Sciences (Springer-Verlag, New York, 1994), Vol. 258.
67. E. Sunde, T. Thorstenson, T. Torsvik, J. E. Våg, and M. S. Espedal, Field-related mathematical model to predict and reduce reservoir souring, in *Proceedings of SPE International Symposium on Oilfield Chemistry, New Orleans, 1993*, SPE 25197.
68. P. K. Sweby, High resolution schemes using flux limiters for hyperbolic conservation laws, *SIAM J. Numer. Anal.* **21**, 995 (1984).
69. J. E. Våg, H. Wang, and H. K. Dahle, Eulerian–Lagrangian localized adjoint methods for systems of nonlinear advection-diffusion-reaction equations, *Adv. Water Resources* **19**, 297 (1996).
70. B. van Leer, On the relation between the upwind-differencing schemes of Godunov, Engquist–Osher, and Roe, *SIAM J. Sci. Stat. Comp.* **5**, 1 (1984).
71. E. Varoglu and W. D. L. Finn, Finite elements incorporating characteristics for one-dimensional diffusion-convection equation, *J. Comput. Phys.* **34**, 371 (1980).
72. H. Wang, *Eulerian–Lagrangian Localized Adjoint Methods: Analyses, Numerical Implementations and Applications*, Ph.D. Thesis, Department of Mathematics, University of Wyoming, 1992.
73. H. Wang, M. Al-Lawatia, and S. A. Telyakovskiy, A Runge–Kutta characteristic method for first-order linear hyperbolic equations, *Numer. Methods PDEs* **13**, 617 (1997).
74. H. Wang, R. E. Ewing, and M. A. Celia, Eulerian–Lagrangian localized adjoint method for reactive transport with biodegradation, *Numer. Methods PDEs* **11**, 229 (1995).
75. 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*, edited by Aldama *et al.* (Computational Mechanics, Southampton/Boston, 1996), Vol. 2, p. 99.
76. J. J. Westerink and D. Shea, Consider higher degree Petrov–Galerkin methods for the solution of the transient convection-diffusion equation, *Int. J. Numer. Methods Eng.* **28**, 1077 (1989).