SIAM J. SCI. COMPUT.
Vol. 34, No. 4, pp. B447–B478

A FULLY CONSERVATIVE EULERIAN-LAGRANGIAN STREAM-TUBE METHOD FOR ADVECTION-DIFFUSION PROBLEMS*

TODD ARBOGAST†, CHIEH-SEN HUANG‡, AND CHEN-HUI HUNG§

Abstract. We present a new method for a two-dimensional linear advection-diffusion problem of a "tracer" within an ambient fluid. The problem should have isolated external sources and sinks, and the bulk fluid flow is assumed to be governed by an elliptic problem approximated by a standard locally conservative scheme. The new method, the fully conservative Eulerian-Lagrangian stream-tube method, combines the volume corrected characteristics-mixed method with the use of a stream-tube mesh. Advection of the tracer is approximated using characteristic tracing in time of regions of space, which maintains mass conservation. However, the shape of a characteristic trace-back region is numerically approximated, so its volume must also be correct to maintain accurate approximation of the tracer density (i.e., the mass of the ambient fluid must be conserved during the advection step). Our new method has the advantages that it is fully locally conservative (both tracer and ambient fluid mass is conserved locally), has low numerical diffusion overall and no numerical cross-diffusion between stream-tubes, can use very large time steps (perhaps 20 to 30 times the CFL limited step), and can use a very coarse mesh, since it is tailored to the flow pattern. Because advection is approximated within stream-tubes, it is essentially one-dimensional, making it relatively easy to implement and computationally efficient. We also present a grid transfer technique to approximate more simply the physical diffusion on a rectangular grid rather than on the stream-tube mesh. The new method can be used for many applications, but especially problems of flow and transport in porous media, which have sources and sinks isolated to wells. Examples include the modeling of groundwater contaminant migration, petroleum production, and carbon sequestration.

Key words. Eulerian-Lagrangian, stream-tube, finite volume, locally conservative, characteristics, hyperbolic transport, porous media, grid transfer

AMS subject classifications. 65M08, 65M25, 65M60, 76M12, 76R50, 76S05

DOI. 10.1137/110840376

1. Introduction.

We consider the linear hyperbolic transport problem with isolated external sources and sinks in which one component (say, a tracer) is predominately advected but also mildly diffused within an ambient fluid. For such problems, characteristic or Eulerian-Lagrangian (or semi-Lagrangian) methods have the advantages that long time steps can be used without loss of stability, numerical diffusion can be low, and relatively coarse computational meshes can be used effectively. An important application is to flow in a porous medium with isolated wells, modeling, e.g., groundwater contaminant migration, petroleum production, and carbon seques-

*Submitted to the journal's Computational Methods in Science and Engineering section July 11, 2011; accepted for publication (in revised form) May 14, 2012; published electronically August 9, 2012.

http://www.siam.org/journals/sisc/34-4/84037.html

†Department of Mathematics, University of Texas at Austin, Austin, TX 78712. Current address: Institute for Computational Engineering and Sciences, University of Texas at Austin, Austin, TX 78712 (arbogast@ices.utexas.edu). This author's work was supported by U.S. National Science Foundation grants DMS-0713815 and DMS-0835745 and by a 2012 Moncrief Grand Challenge Faculty Award from the University of Texas at Austin.

‡Corresponding author. Department of Applied Mathematics, National Sun Yat-sen University, Kaohsiung 804, Taiwan, R.O.C. (huangcs@math.nsysu.edu.tw). This author's work was supported in part under Taiwan National Science Council grant 99-2115-M-110-006-MY3.

§Department of Mathematic and Physical Sciences, R.O.C. Air Force Academy, No. Sisou 1, Jieshou W. Rd., Gangshan Dist., Kaohsiung City 82047, Taiwan (hungch@math.nsysu.edu.tw).

tration. We will use terminology suitable for this case, although the problem and our method are more general.

To be specific, we consider an incompressible bulk fluid moving with a velocity $\mathbf{u}(x)$ in a bounded, two-dimensional domain $\Omega \subset \mathbb{R}^2$. We have assumed that $\mathbf{u}$ is independent of time for simplicity. A dilute tracer of concentration $c(x,t)$ within the bulk fluid over the time interval of interest $J = (0,T]$ satisfies the advection-diffusion equation

$$(\phi c)_t + \nabla \cdot (c\mathbf{u} - D\nabla c) = c_I q_+ + c q_- \equiv q_c(c) \quad \text{in } \Omega \times J, \tag{1.1}$$

$$(c\mathbf{u} - D\nabla c) \cdot \nu = 0 \quad \text{on } \partial\Omega \times J, \tag{1.2}$$

$$c(\cdot, 0) = c^0 \quad \text{in } \Omega, \tag{1.3}$$

where $\phi(x)$ is the storage factor of the medium called *porosity,* subscript t is time partial differentiation, $D(x,t) \geq 0$ is the diffusion/dispersion tensor (that may depend on $\mathbf{u}$), $q(x)$ is a given and isolated external source or sink function (i.e., the function modeling wells, a sum of Dirac measures), $q_+(x) \geq 0$ is q when $q > 0$ and 0 otherwise, $q_-(x) = q - q_+ \leq 0$, $c_I(x,t)$ is the given concentration of injected fluid, $\nu(x)$ is the outer unit normal vector to $\partial\Omega$, and $c^0(x)$ is the initial tracer concentration. For simplicity, we will take $\phi = 1$ and so drop it from the equations, but the method we propose would apply to the full system with c below replaced by ϕc and any "volume" being pore volume.

Many schemes have been developed to solve the transport equation (1.1) using an Eulerian-Lagrangian approach [22, 32, 25, 30, 15, 2, 7, 18, 23, 21, 20, 38, 3, 13, 14, 37, 12]. Each of these schemes treats advection using a characteristic tracing algorithm (a Lagrangian frame of reference) from a fixed Eulerian mesh over each time step. In this paper, we use ideas from stream-tube methods [29, 39, 17, 11, 31] to better tailor the fixed Eulerian mesh used in the method. We then apply a modified version of the fully conservative, volume corrected characteristics-mixed method (VCCMM) developed by two of the authors [3], which uses characteristics for finite volume advective transport and a mixed finite element method for diffusion. It has been shown that this method, which conserves mass locally for both the tracer and the ambient fluid, has less numerical diffusion than competing methods [3, 4, 5]. This results in a new method, the fully conservative Eulerian-Lagrangian stream-tube method.

If we omit diffusion for the immediate discussion, VCCMM, in brief, is defined at each time level t^{n+1} by first tracing the edges of a fixed Eulerian mesh element backward in time along the flow velocity $\mathbf{u}$ to the previous time level t^n. Next, (1.1) is integrated over this finite volume, space-time characteristic trace-back region. Since the fluxes across the two space-time side-lateral boundaries vanish, ignoring wells for simplicity, the computation reduces to two terms. One term is the tracer mass at time t^{n+1} within the mesh element, which is set equal to the other term, an integration of the tracer at the earlier time level t^n over the trace-back region. As long as the volumes of the original mesh element and the trace-back element agree, the method works well.

VCCMM is defined no matter what Eulerian mesh is used. Rectangular grids are often taken in practice, since they are simple to work with. However, there can be numerical cross-diffusion between mesh elements when the Eulerian mesh is fixed arbitrarily, independent of the flow. If instead we define the fixed mesh so that many elements have boundaries that coincide with the characteristics of the flow field, then we avoid much of this numerical cross-diffusion. Perhaps the simplest way to accomplish this is to use characteristic streamlines that travel from injection to production wells. A stream-tube is the region bounded by two such streamlines. Thus our method, in the absence of physical diffusion, will properly restrict the tracer

to be transported within a single stream-tube. Moreover, the computation simplifies, since it basically reduces to a one-dimensional transport along the stream-tube.

The one-dimensional nature of the scheme turns out to be important in improving the efficiency of the method. As defined in [3], the VCCMM includes a critical geometric volume balance bisection algorithm that can be computationally costly. The structure of the method was elucidated in [6], which pointed out that the volume balance can be formulated purely as an algebraic constraint within the method. However, in multiple dimensions, it is not clear how to balance the volume to obtain this algebraic constraint without resorting to geometric techniques. But it is trivial to apply the constraint in one dimension, and thereby avoid the costly geometric volume balance calculation, improving the implementation of the VCCMM considerably. Moreover, the complex and problem specific selection of layers in the volume adjustment algorithm of VCCMM is no longer needed.

We will develop the method through a computational test example (and a variation of it in section 8), so that it is easier to explain the details of the method. One difficulty will be to treat the physical diffusion, which will be handled by using operator splitting between advection and diffusion. The diffusion can be approximated on the stream-tube mesh used for advection. However, it is perhaps more natural to use a regular grid, since it is less computationally costly to set up and solve diffusion on a regular grid. We present such a method. We will see that it works well as long as our advection-diffusion problem (1.1)–(1.3) is advection dominated. (In the diffusion dominated case, a standard method on the stream-tube mesh should be used.) Overall, our numerical results will show that we obtain good resolution of the advection on very coarse grids, that we can use extremely large time steps (on the order of 10 to 30 times the CFL limited step size), and that the concentration exhibits a very small amount of numerical diffusion.

In porous medium applications, the velocity $\mathbf{u}$ will be governed by the mass conservation principle, and it will be given by Darcy's law from the pressure p, that is,

$$\nabla \cdot \mathbf{u} = q \qquad \text{in } \Omega, \tag{1.4}$$

$$\mathbf{u} = -k\nabla p \quad \text{in } \Omega, \tag{1.5}$$

$$\mathbf{u} \cdot \nu = 0 \qquad \text{on } \partial\Omega, \tag{1.6}$$

where $k(x)$ is the *permeability* of the medium divided by the fluid viscosity. We will use a standard locally conservative method to approximate this equation.

We close the introduction with a brief discussion of potential generalizations. First, Eulerian-Lagrangian methodology is generally difficult to implement in two and three dimensions. The use of stream-tubes reduces essentially to one dimension and thus allows solution of two-dimensional problems, as the rest of the paper shows. However, our method can perhaps be seen as a natural approach to a feasible three-dimensional implementation. It is perhaps clear, however, that a simple generalization would likely be nontrivial and computationally expensive. Second, the method has potential to be generalized to formally higher order methods using, e.g., one-dimensional WENO reconstructions [28], although there are unresolved questions about how to handle accurately the variation in the stream-tube cross-sectional width. Finally, the simplicity of the methodology suggests that it may generalize to more complex flows, such as two-phase flows [5] and miscible displacement. These two cases would involve time-varying velocity fields and therefore stream-tube mesh generation at each time step. Even though these potential generalizations might be computationally costly, it is possible that the ability to use long time steps and coarse grids would make them competitive with simpler Eulerian methods.

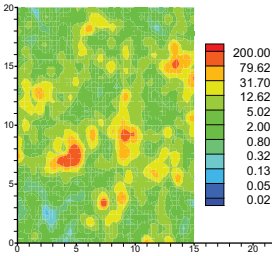


FIG. 2.1. *The permeability k of our test problem on a log scale.*

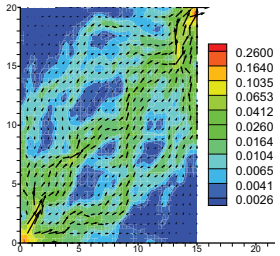


FIG. 2.2. *The speed in contour on a log scale and the velocity vectors as arrows.*

2. A locally conservative method for the flow velocity. Approximation of our advection-diffusion problem begins with an approximation of the pressure equation (1.4)–(1.6) for the velocity $\mathbf{u}$ (and pressure p). Since later we will be interested in local mass conservation of the tracer concentration, it is recommended that we use a locally conservative method for the pressure equation. There are many possibilities, including the use of finite volumes, mixed finite elements, discontinuous Galerkin, mimetic finite differences, and multipoint flux approximations. We will use mixed finite elements [33, 19, 34].

As noted in the introduction, we present the method in the context of a specific numerical example, to better explain our overall method. In fact, the example is taken from [6]. In this test case, as well as in this paper, the well term q is independent of time t, so we need only compute $\mathbf{u}(x)$ once at the beginning of the computations instead of at the beginning of each time step. The domain $\Omega = (0, 15) \times (0, 20)$ has a permeability k defined as a piecewise constant function on a 50×50 rectangular grid, as depicted in Figure 2.1 on a log scale. It is geostatistically generated and mildly correlated, with a mean of 10 and varying from 0.022 to 365. We place an injection well in the lower left corner grid cell and a production well in the top right corner cell of the opposite strength. That is, if $\ell_1 = 15/50$ and $\ell_2 = 20/50$ are the grid spacings, then $q(x) = 0$ everywhere except that $q = \tilde{q} = 1.033$, a positive constant, for $x \in (0, \ell_1) \times (0, \ell_2)$ and $q = -\tilde{q}$ for $x \in (15 - \ell_1, 15) \times (20 - \ell_2, 20)$.

The flow was computed using the lowest order Raviart–Thomas mixed method on the 50×50 rectangular grid. The velocity and speed on a log scale are depicted in Figure 2.2. We also computed the flow on a refined 100×100 grid with the same permeability $k(x)$ (i.e., the value in each original 50×50 grid cell was repeated in the 2×2 refinement), but the source term $q(x)$ is still defined only over the corner grid cells, and so it is both four times smaller and four times stronger.

3. Operator splitting for the advection-diffusion problem. If the diffusion tensor D vanishes, we need only compute advection. When nonzero physical diffusion is present, we use operator splitting [35] to handle advection and diffusion separately. We use an appropriate numerical method and mesh for each process. Let $\Delta t > 0$ be the time step and consider advancing the solution from time $t^n = n\Delta t$ to $t^{n+1} = (n + 1)\Delta t$. (We use a constant Δt for simplicity, but this is not necessary for the definition of the method.)

A first order in time Strang splitting would solve the advection problem

$$c_t + \nabla \cdot (c\mathbf{u}) = q_c(c) \quad \text{in } \Omega \times (t^n, t^{n+1}], \tag{3.1}$$

$$c(\cdot, t^n) = c^n \quad \text{in } \Omega \tag{3.2}$$

over the time interval $(t^n, t^{n+1}]$, starting from the computed tracer concentration $c^n(x) \approx c(x, t^n)$, to obtain $c_{\mathrm{A}}^{n+1}(x)$. For this advection step, we use a fully conservative stream-tube method with the tracer defined on the stream-tube mesh, as described in the next section. For the diffusion problem,

$$c_t - \nabla \cdot (D\nabla c) = 0 \qquad \text{in } \Omega \times (t^n, t^{n+1}], \tag{3.3}$$

$$D\nabla c \cdot \nu = 0 \qquad \text{on } \partial\Omega \times (t^n, t^{n+1}], \tag{3.4}$$

$$c(\cdot, t^n) = c_{\mathrm{A}}^{n+1} \quad \text{in } \Omega, \tag{3.5}$$

we use a standard mixed finite element method on a rectangular grid, as described in section 5, although other methods could be used for this part of the computation. The difficulty will be to define the forward and inverse transfers of the concentration between the stream-tube mesh and the rectangular grid.

A second order in time Strang splitting would solve the advection problem over half the time interval $(t^n, t^{n+1/2} \equiv (n+1/2)\Delta t]$, the diffusion problem over the entire interval $(t^n, t^{n+1}]$, and then the advection problem over the other half of the time interval $(t^{n+1/2}, t^{n+1}]$.

For our numerical example, we used a first order Strang splitting.

4. The fully conservative stream-tube method for advection. In this section, we describe our fully conservative Eulerian-Lagrangian stream-tube method for solving the advection problem (3.1)–(3.2). We first describe the tracing of streamlines and the construction of our stream-tube mesh. We then describe the approximation of tracer transport over each time step.

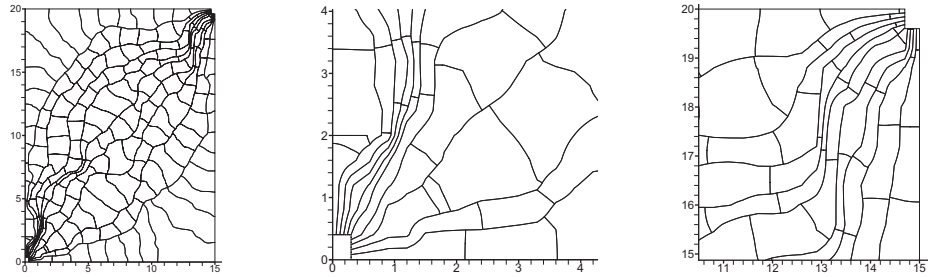
4.1. Streamlines and the stream-tube mesh. In the absence of diffusion and wells, tracer particles governed by our advection equation (3.1) travel along streamlines or characteristics. These lines will be parametrized by a time-like variable $\tau \in \mathbb{R}$. The streamline $\hat{x}(\tau, x)$ that passes initially through a given point $x \in \Omega$ satisfies the ordinary differential equation

$$\frac{\partial \hat{x}}{\partial \tau} = \mathbf{u}(\hat{x}(\tau, x)), \quad \tau \in \mathbb{R}, \tag{4.1}$$

$$\hat{x}(0, x) = x, \qquad x \in \Omega. \tag{4.2}$$

In our context, a *stream-tube* is a region bounded by streamlines and the boundaries of two wells. In the absence of diffusion, particles cannot cross the streamline boundaries, so particles within a stream-tube travel from an injection well to a production well.

We now describe the construction of the stream-tube mesh. For our numerical example, its final form is shown in Figure 4.1. This is the mesh on which the tracer concentration is defined for advection purposes. We first define the stream-tubes themselves. In terms of our test example, recall that $\ell_1 = 15/50$ and $\ell_2 = 20/50$ are the grid spacings. We locate initial points, evenly or unevenly spaced as we choose, on the boundary of the injection well, i.e., on the set $\{\ell_1\} \times (0, \ell_2) \cup (0, \ell_1) \times \{\ell_2\}$, including the endpoints on $\partial\Omega$ at $(\ell_1, 0)$ and $(0, \ell_2)$. We took 21 points, including the corner (ℓ_1, ℓ_2), so that we had 11 equally spaced points on both the horizontal and the



(a) *The entire domain* (b) *Near the injection well* (c) *Near the production well*

FIG. 4.1. *The stream-tube mesh, consisting of stream-tubes and stream-cells.*

vertical parts of the boundary. We next trace the streamline according to (4.1)–(4.2) from the given initial point on the injection boundary until it touches the production well boundary, i.e., the set $\{15-\ell_1\}\times(20-\ell_2,20)\cup(15-\ell_1,15)\times\{20-\ell_2\}$. We will elaborate on how we trace and represent the streamlines shortly. Each stream-tube is composed of three consecutive streamlines; the exterior two act as the boundary of the stream-tube, and the middle streamline is called the center-line of the stream-tube. We thus have defined 10 stream-tubes and 10 center-lines.

Once the stream-tubes are formed, we decompose each stream-tube into a mesh of *stream-cells,* which are bounded by the two exterior streamlines and two lines transverse to the flow. To determine the transverse lines, we begin by choosing points along the center-line. We took these to be nearly equally spaced in terms of the arc length, from the injection well downstream to the production well. These will be modified near the injection well shortly, so a few of them do not appear to be equally spaced in Figure 4.1. We include the initial point on the injection well and a final point on the production well. The transverse lines at these endpoints are the well boundaries. At the interior center-line points, the transverse lines are given by tracing in the direction perpendicular to the velocity $\mathbf{u}$ both ways to the stream-tube boundaries. That is, the transverse line $x^\dagger(\tau,x)$ solves an equation similar to (4.1), but with $\mathbf{u}=(u_1,u_2)$ replaced by $(u_2,-u_1)$, as in

$$\frac{\partial x^\dagger}{\partial\tau}=\mathbf{u}(x^\dagger(\tau,x)),\quad \tau\in\mathbb{R}, \tag{4.3}$$

$$x^\dagger(0,x)=x,\qquad\qquad x\in\Omega. \tag{4.4}$$

In Figure 4.1, we have 10 stream-tubes and approximately 20 stream-cells within each stream-tube, for a total of about 200 mesh cells, which is far fewer that the 2500 mesh cells in the rectangular grid used for the pressure equation. Since the streamlines decompose the domain according to the physical flow, they provide good information on how the tracer is being transported, even when a very coarse stream-tube mesh is used.

The injection well is treated in a special way, since it acts as a source of tracer mass. We treat each stream-tube independently. Given the time step $[t^n,t^{n+1}]$, the total volume of fluid injected into the stream-tube along its injection well boundary W can be computed according to

$$V=\int_{t^n}^{t^{n+1}}\int_W \mathbf{u}\cdot\nu\,ds\,dt. \tag{4.5}$$

We then determine the point on the center-line of the stream-tube for which the transverse line there, the streamline boundaries, and the well boundary enclose a region R of the correct volume V, i.e., so that this first region R is flooded entirely by fluid from the well. That is, we require that $V = |R|$, where for any set S, $|S|$ is its volume (actually its area, since we discuss the two-dimensional version of the method, but we prefer to use the term "volume" of the general case). A simple bisection strategy can be used to find this center-line point and transverse line (similar to what is used in the VCCMM [3], but here only for a single stream-cell at a time and not entire layers of cells). This first region R we call the *injection well stream-cell.* We note in passing that it could be determined equivalently by tracing streamlines forward from the well [27, 3].

The stream-tube mesh is now modified to include this injection well stream-cell, possibly replacing some stream-cells and reducing the size of one stream-cell that is only partially flooded by the well. (Such stream-cells are clearly evident in the stream-tube mesh shown in Figure 4.1(b).)

4.2. Aspects of streamline and transverse line tracing. We approximate each streamline and transverse line numerically as a polyline, which is a doubly linked list of vertices. Within the 50×50 rectangular grid used to solve the pressure equation for $\mathbf{u}$, each streamline will cross many grid cells. Within each such grid cell, we represent the polyline using about five vertices. The exact number of vertices used is based on the total time needed for a particle, traveling at the velocity $\mathbf{u}$, to traverse the grid cell along the streamline, divided by some reference time, up to a maximum fixed number of vertices. (We used 10, and no difference was observed between this value and a maximum of 50.) The vertices are taken to be evenly distributed with respect to the traversal time.

Since we use the lowest order Raviart–Thomas mixed finite element space to approximate $\mathbf{u}$, the streamlines can be computed analytically within each rectangular grid cell [26]. That is, on a single grid cell, the velocity is simply $\mathbf{u} = (a + bx, c + dy)$, and the equations for the streamlines (4.1)–(4.2) decouple into two independent ordinary differential equations which can be solved easily. The transverse lines must be computed numerically using an ordinary differential equations solver.

Raviart–Thomas spaces give continuous normal velocities but *not* fully continuous velocity fields. We also tested the use of the lowest order spaces proposed by Arbogast and Wheeler [8]. These are Stokes elements that provide accurate approximations to the second order Darcy system (1.4)–(1.6), and they give fully continuous velocities. In this case, we do not have an analytic solution to the streamlines. A numerical solution using microstepping is then necessary. However, because the streamlines are computed only approximately, it is easy to find that the polylines intersect, and correction action is needed. For example, a troublesome area is seen in Figure 4.1(c) at around $(13, 17.5)$. Moreover, there is no fluid exchange between stream-tubes for the pure advection problem, so the sum total of fluid volume that enters from the injection well (4.5) must equal that which leaves the domain at the production well within each stream-tube. This property is maintained automatically when using the Raviart–Thomas space for the velocity, since the computed velocity $\mathbf{u}$ is divergence free pointwise away from wells and streamlines are computed analytically. This is not the case when the Arbogast–Wheeler spaces are used, and some adjustment of the polylines is needed. In conclusion, the streamlines must be traced extremely accurately, and analytic tracing is preferred. The natural way to implement this is to solve the pressure equation using either the lowest order Raviart–Thomas spaces or

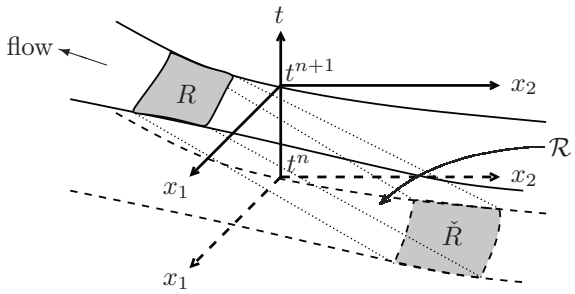


FIG. 4.2. *The stream-cell $R \subset \Omega$ at time t^{n+1} is traced back in time under the flow field* $\mathbf{u}$ *to the trace-back region $\check{R} \subset \Omega$. The space-time trace-back region $\mathcal{R} \subset \Omega \times [t^n, t^{n+1}]$ is also traced.*

another method combined with a projection to the Raviart–Thomas spaces in a mass conservative fashion (see, e.g., [16]).

When using the lowest order Raviart–Thomas spaces, a complication arises in tracing the transverse lines. The tangential component of velocity is not continuous, so it is possible that the transverse lines bifurcate when tracing from one element to another. In the rare case that this occurs, we average the two possibilities. No such bifurcation was observed for our main test example, but it was observed in other tests.

4.3. Theoretical basis for the advection computation. In this subsection, we interrupt the definition of our stream-tube method for advection to discuss some of its theoretical aspects. As noted in [2, 7, 3], we can transport tracer mass within a region of space R over time by simply tracing the motion of the boundary ∂R under the influence of the velocity field. Douglas and Russell [22] found it simpler to trace the streamline *backward* in time from $\tau = t^{n+1}$ to $\tau = t^n$, using (4.1)–(4.2). As depicted in Figure 4.2, given any time t^{n+1} and region of space R, we trace each point back in time to form a region $\check{R}$. In fact, we trace out a space-time region $\mathcal{R}$, bounded on "top" by $R \times \{t^{n+1}\}$, on the "bottom" by $\check{R} \times \{t^n\}$, and on the "sides" by a complex space-time region. In the pure advection problem, no fluid crosses the "sides" of $\mathcal{R}$, so fluid is transported from $\check{R}$ to R. For incompressible fluids, we also note the *volume balance* condition [3]

$$|R| = |\check{R}|, \tag{4.6}$$

which says that the volume of R agrees with the volume of $\check{R}$.

In theory, we could define the next step in our algorithm to be the tracing back of each stream-cell R to find $\check{R}$, as in [3]. We would then set the concentration in the stream-cell R at time t^{n+1} to be

$$c_R^{n+1} = \frac{1}{|R|} \int_{\check{R}} c^n(x)\, dx. \tag{4.7}$$

The key to obtaining an accurate characteristic method is to preserve the volume balance condition (4.6). It is difficult to do this in a geometric way in multiple dimensions, since the trace-back regions $\check{R}$ are computed only approximately. A complex volume adjustment algorithm was given in [3] to modify $\check{R}$ until (4.6) is satisfied for each R.

In this work, we advocate using the stream-tube mesh to define the transport regions; that is, the preferred regions R are the stream-cells. In this way, fluid is

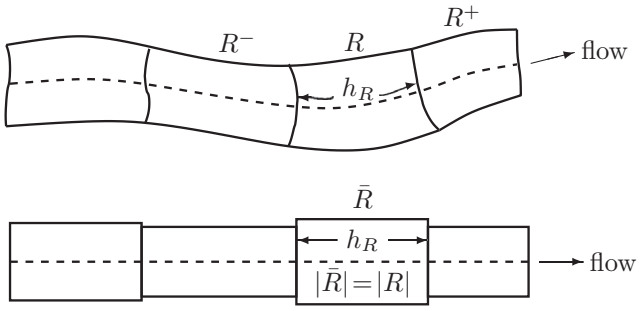


FIG. 4.3. *A stream-tube and its reference configuration. The reference stream-tube is defined as a union of rectangular reference stream-cells such that that the volume of each stream-cell R agrees with the volume of its reference stream-cell $\bar{R}$, i.e., $|R| = |\bar{R}| = h_R k_{\bar{R}}$.*

advected through the tube essentially in a one-dimensional way along each stream-tube's center-line, and no artificial numerical cross-diffusion will arise. Moreover, it turns out that the trace-back regions $\check{R}$ need not be computed in one dimension, since the fluid is constrained to flow through the stream-tube, and the complex geometric volume balance condition (4.6) can be replaced by a much simpler condition given in subsection 4.5 below. That is, the use of a stream-tube mesh allows us to avoid the complex and problem-specific geometric layer-by-layer volume adjustment technique of the VCCMM. In fact, we only need to balance the volume of the well stream-cells, and these independently of each other.

4.4. A reference stream-tube and postprocessed concentration. For a fixed stream-tube, as shown in Figure 4.3, we map it to a reference configuration consisting of rectangles, one for each stream-cell. The stream-cell R is mapped to $\bar{R}$, a rectangle of length h_R, where h_R is the arc length of the part of the center-line contained in R, and width $k_{\bar{R}}$ such that the volume of R agrees with that of $\bar{R}$, i.e., $|R| = |\bar{R}| = h_R k_{\bar{R}}$, so that we preserve volume under our map. In fact, we do not need an explicit representation of the map to perform the advection step.

The tracer concentration is approximated by piecewise discontinuous constants within each stream-cell. However, knowledge of the average concentration in the stream-cells can be used to reconstruct its distribution to a formally higher order of accuracy [36]. Since fluid is constrained within each stream-tube, this reconstruction or postprocessing should be done in a one-dimensional way along the stream-tube. The calculations are performed on the reference stream-tube. As is typical for hyperbolic advection schemes, such a reconstruction can significantly sharpen fronts, and this is exactly what we observed for our method as well.

Given concentration values

$$c_R^n \approx \frac{1}{|R|} \int_R c(x, t^n)\, dx \tag{4.8}$$

over each stream-cell R or $\bar{R}$, we define a postprocessed concentration $\tilde{c}^n(x)$ to be a piecewise linear function along the center-line and constant in the transverse direction. On the reference stream-cell $\bar{R}$, let y be the one-dimensional coordinate aligned with the center-line and increasing in the downstream direction, and let $\bar{y}_{\bar{R}}$ be the midpoint of the center-line of $\bar{R}$. Then we define on $\bar{R}$

$$\tilde{c}_{\bar{R}}^n(y) = c_R^n + \sigma_R^n L_{\bar{R}}(y), \tag{4.9}$$

where the slope σ_R^n is defined next and the function

$$L_{\bar{R}}(y) = y - \bar{y}_{\bar{R}}.$$

This implicitly defines $\tilde{c}_R^n(x)$ on R via the inverse mapping.

To define σ_R^n, let R^- be the stream-cell within the stream-tube adjacent to and upstream from R (i.e., toward the injection well), and similarly let R^+ be the stream-cell adjacent to and downstream from R (i.e., toward the production well), if these exist (otherwise, set them to be R). Recall that h_R is the arc length of the part of the center-line contained in R or, equivalently, $\bar{R}$. We define the initial slopes by

$$\sigma_{R^+}^n = \frac{c_{R^+}^n - c_R^n}{\frac{1}{2}h_R} \quad \text{and} \quad \sigma_{R^-}^n = \frac{c_R^n - c_{R^-}^n}{\frac{1}{2}h_R}, \tag{4.10}$$

and then slope limit these to define

$$\sigma_R^n = \begin{cases} 0 & \text{if } \sigma_{R^+}^n \sigma_{R^-}^n < 0, \\ \sigma_{R^+}^n & \text{if } |\sigma_{R^+}^n| < |\sigma_{R^-}^n|, \\ \sigma_{R^-}^n & \text{otherwise.} \end{cases} \tag{4.11}$$

We remark that other procedures could be used to define this postprocessing step besides this MUSCL reconstruction [36]. This procedure is simply the one we used for our numerical example.

4.5. The advection step. We are finally ready to define the advection step, i.e., we define the average concentration c_R^{n+1} at time t^{n+1} for each stream-cell R (recall (4.8)) from $\tilde{c}^n(x)$. We proceed for a fixed stream-tube.

Beginning from the injection side of the stream-tube, we set the concentration in the first stream-cell R_0, the injection well stream-cell, according to the injection concentration $c_I(t)$ of the well, since it entirely floods this stream-cell. More precisely,

$$c_{R_0}^{n+1} = \frac{1}{\Delta t} \int_{t^n}^{t^{n+1}} c_I(t)\, dt. \tag{4.12}$$

No other stream-cell is affected directly by the well.

For any noninjection well stream-cell R, inspired by (4.7), we require in theory that

$$|R|\, c_R^{n+1} = \int_{\check{R}} \tilde{c}^{\,n}(x)\, dx = \sum_S \int_{S \cap \check{R}} \tilde{c}_S^n(x)\, dx, \tag{4.13}$$

where the sum is taken over the stream-cells in the stream-tube. Fortunately, in one-dimensional flow, we need not compute according to (4.13), which would require a careful geometric representation of $\check{R}$, but rather we can compute on the reference stream-tube.

To continue, it is helpful to set some notation, as illustrated in Figure 4.4. Within the reference stream-tube, let us order the stream-cells downstream from the injection well stream-cell as $\bar{R}_0$, $\bar{R}_1 = \bar{R}_0^+$, $\bar{R}_2 = \bar{R}_1^+$, and so on, i.e., $\bar{R}_i = \bar{R}_{i-1}^+$ for $i \geq 1$. Define the points y_0, y_1, y_2, ... so that $\bar{R}_i$ occupies the part of the center-line from y_i to y_{i+1}, let $h_i = y_{i+1} - y_i = h_{R_i}$ be the length of $\bar{R}_i$, and let $k_i = k_{\bar{R}_i} = |\bar{R}_i|/h_i$ be the width of $\bar{R}_i$ for $i \geq 0$.

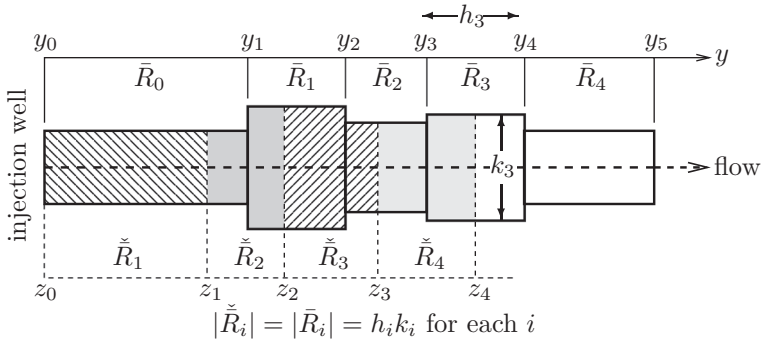


FIG. 4.4. *Advection computations within the reference stream-tube. The points y_i give the position of the transverse lines of the reference stream-cells. The point $z_0 = y_0$ gives the position of the boundary of the injection well. For $i = 1, 2, 3, \ldots$, the point z_i is determined by the condition that the reference stream-tube volume between z_{i-1} and z_i, $|\check{\bar{R}}_i|$ agrees with the volume of the reference stream-cell $|\bar{R}_i| = h_i k_i$. The mass in $\check{\bar{R}}_i$ is transported to $\bar{R}_i$ over the time step.*

Continuing with Figure 4.4, we define the reference trace-back regions $\check{\bar{R}}_i$ as follows. We fix the points $z_0 = y_0$, and then successively we define z_i, $i \geq 1$, so that the volume (i.e., area in two dimensions) of the stream-tube between z_{i-1} and z_i is the same as $|\bar{R}_i| = |R_i| = h_i k_i$. That is, we enforce the volume balance condition (4.6). We then define the trace-back region $\check{\bar{R}}_i$ to be the part of the stream-tube that occupies the part of the center-line from z_{i-1} to z_i. For example, in Figure 4.4, $\check{\bar{R}}_2$ is composed of parts of two stream-cells, $[z_1, y_1] \times [-k_0/2, k_0/2]$ and $[y_1, z_2] \times [-k_1/2, k_1/2]$, where z_2 was defined by the condition that $(y_1 - z_1)k_0 + (z_2 - y_1)k_1 = h_2 k_2$.

Returning to (4.13), since the center-line of $\bar{R}_j$ is $[y_j, y_{j+1}]$ and of $\check{\bar{R}}_i$ is $[z_{i-1}, z_i]$, we obtain that

$$
\begin{aligned}
(4.14)\quad c_{R_i}^{n+1} &= \sum_j \frac{1}{|R_i|} \int_{R_j \cap \check{R}_i} \tilde{c}_{R_j}^n(x)\, dx \\
&= \sum_j \frac{k_j}{|R_i|} \int_{[y_j, y_{j+1}] \cap [z_{i-1}, z_i]} \left(c_{R_j}^n + \sigma_{R_j}^n L_{\bar{R}_j}(y) \right) dy \\
&= \sum_j \left\{ \left(\frac{|\bar{R}_j \cap \check{\bar{R}}_i|}{|\bar{R}_i|} \right) c_{R_j}^n + \left(\frac{k_j}{|\bar{R}_i|} \int_{[y_j, y_{j+1}] \cap [z_{i-1}, z_i]} L_{\bar{R}_j}(y)\, dy \right) \sigma_{R_j}^n \right\}.
\end{aligned}
$$

Thus advection amounts to combining the values $c_{R_j}^n$ for various R_j both directly and through $\sigma_{R_j}^n$. We need only to compute volumes of rectangular regions and the linearly weighted integral $\int_{[y_j, y_{j+1}] \cap [z_{i-1}, z_i]} L_{\bar{R}_j}(y)\, dy$ over an interval. The description of the advection step is complete.

5. A locally conservative method for the diffusion. In this section we describe how to solve the diffusion problem (3.3)–(3.5) within the Strang splitting algorithm (assuming that D is nonzero). But first we note that we use a standard model for the the diffusion/dispersion coefficient D. It is defined by three parameters, the molecular diffusion coefficient d_{mol}, which we take to be 10^{-11} m^2/s in our main numerical example, and two parameters modeling dispersion in a porous medium with respect to the velocity $\mathbf{u}$. Along the flow direction, we need the longitudinal dispersion coefficient d_{long}, taken to be 10^{-5} m, and transverse to the flow, we need

the transverse dispersion coefficient d_{trans}, taken to be 10^{-6} m. The full model gives a tensor, defined by

$$D(\mathbf{u}) = \phi(x)\, d_{\text{mol}}\, I + |\mathbf{u}(x)|\big[d_{\text{long}}\, E(\mathbf{u}) + d_{\text{trans}}\big(I - E(\mathbf{u})\big)\big], \tag{5.1}$$

where $E(\mathbf{u})$ is the tensor projecting onto the vector $\mathbf{u}/|\mathbf{u}|$, i.e., onto the direction of the flow. Recall that we take $\phi = 1$ in this paper.

We remark that our diffusion/dispersion coefficients are generally quite small, which is typical in certain porous medium applications. Our goal here, though, is to emphasize the advection. But we do not omit diffusion/dispersion, since it adds certain numerical complications that need to be addressed.

Since the advected concentration is defined on the stream-tube mesh, it seems natural to approximate the diffusion problem on the same mesh. As always, we should use a locally mass conservative method for the diffusion problem, such as a discontinuous Galerkin, mimetic, or multipoint flux approximation. We would need to refine the stream-tube mesh, however, since it is much too coarse for resolving small diffusion. Moreover, the resulting mesh is likely to be nonconforming, since the original stream-tubes are divided into stream-cells independently of each other.

In this paper, however, we take a different point of view, and develop a diffusion approximation defined on a simple rectangular grid, since this is more natural for elliptic diffusion problems. Our approximation is subject to grid transfer error and so is not as accurate as a direct approximation on the stream-tube mesh. However, as we will show, it is a useful approximation when the physical diffusion is small.

5.1. Diffusion step. The diffusion step has three parts. We need to (1) develop a forward transfer operator from stream-cell concentrations to rectangular grid cell concentrations, (2) apply a diffusion operator on a rectangular grid, and finally (3) apply an inverse transfer from grid cell concentrations back to stream-cell concentrations.

We require four properties of the process. First, the transfers should be nonnegative, since negative concentrations are unphysical. Second, if we follow the forward transfer operator directly by the inverse transfer operator, without solving the physical diffusion problem in between, we obtain the original concentration on the stream-tube mesh. This is a requirement that the grid transfer operators minimize numerical diffusion associated purely with grid transfer effects. Third, each stage should be mass conservative. Finally, the entire step should model diffusion as a physical process, akin to what one would obtain if one solved the problem on the stream-tube mesh without any grid transfers. We develop two transfer operators with these four properties, a simple, locally constant one presented here and a somewhat improved locally linear version in subsection 5.4.

5.1.1. Locally constant forward transfer. The first part of the diffusion step is to transfer the concentration onto a simple mesh, which in our numerical example is the same as the 50×50 rectangular grid used earlier for solving the pressure equation. We start from the stream-tube concentrations, which are piecewise constant on the stream-tube mesh, and we need to define the average concentrations on the rectangular grid. A simple nonnegative, mass preserving, locally constant transfer operator can be defined geometrically by dividing the mass into the intersection of the rectangular and stream-tube meshes and then averaging the results to the rectangular grid. This procedure can be accomplished, for example, by using the Sutherland–Hodgman clipping algorithm [24] to find the intersection points of the two meshes.

At this stage of the Strang operator splitting procedure, each stream-cell concentration c_S^n is actually the concentration after applying the advection step. To define the first (locally constant) transfer operator, consider the rectangular grid element G and the stream-cell S. We define the transfer from the set of stream-tube concentrations $\{c_S^n\}$ to the rectangular grid concentration on G by

$$c_G^n = \sum_S \frac{|G \cap S|}{|G|}\, c_S^n, \tag{5.2}$$

where the sum is over all the stream-cells in the mesh.

5.1.2. Diffusion on the rectangular grid. We next need to solve the diffusion problem (3.3)–(3.5) over the time step, starting from the grid concentrations $\{c_G^n\}$ and ending with the diffused grid concentrations $\{\breve{c}_G^{n+1}\}$. The definition of the inverse transfer will require the original $\{c_G^n\}$, so these cannot be overwritten in computer memory. One should use a method that both is locally mass conservative and satisfies the maximum principle.

In fact, we use the expanded mixed finite element method with lowest order Raviart–Thomas finite elements, approximated as cell-centered finite differences [9], since this is easy to implement. However, because we use a scheme that does not satisfy the maximum principle, and because we use large time steps for the advection computation, we needed to use a smaller time step for the diffusion problem. That is, we solve the diffusion problem (3.3)–(3.5) over the time interval $[t^n, t^{n+1}]$ using some number M of microsteps, each of size $\Delta t/M$. In our test example, we generally used $\Delta t = 20$ or 30 and fixed M the same, so that always $\Delta t/M = 1$. The use of a proper locally conservative, maximum principle preserving scheme would avoid the need for such computational machinations. On the other hand, our numerical results show that a poor diffusion solver can still be used effectively in the overall approximation scheme.

5.1.3. Locally constant inverse transfer. We need to define the transfer back to the stream-tube mesh by defining the diffused concentrations $\{c_S^{n+1}\}$ at the advanced time level for each stream-cell S. No simple inverse transfer can satisfy the four requirements mentioned at the beginning of this subsection. We will define the inverse transfer operator here and leave it to subsection 5.3 to provide an heuristic physical justification.

Using the undiffused c_G^n and c_S^n, let

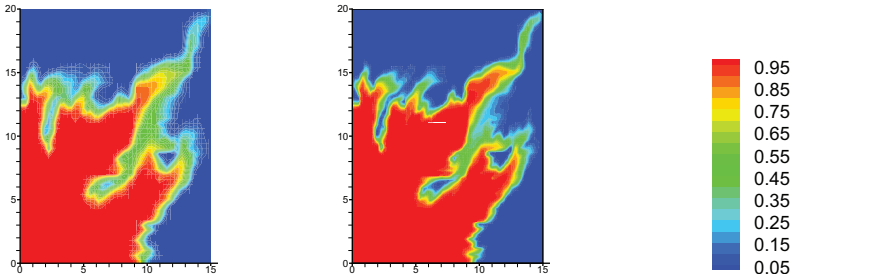
$$\rho_{SG}^n = \begin{cases} c_S^n / c_G^n & \text{if } c_G^n \neq 0, \\ 1 & \text{otherwise} \end{cases} \tag{5.3}$$

and

$$\alpha_G^n = \frac{4\, d_G\, \Delta t}{|G| + 4\, d_G\, \Delta t}, \quad d_G = \frac{1}{|G|} \int_G |D|\, dx. \tag{5.4}$$

Note that d_G is the average magnitude of the diffusion coefficient on G. The locally constant inverse transfer operator is defined by

$$c_S^{n+1} = \breve{c}_S^{n+1} = \sum_G \frac{|G \cap S|}{|S|} \big[(1 - \alpha_G^n)\rho_{SG}^n + \alpha_G^n\big] \breve{c}_G^{n+1}, \tag{5.5}$$



(a) 50×50 *grid,* $\Delta t = 0.97$ (b) 100×100 *grid,* $\Delta t = 0.24$ (c) *Concentration scale*

FIG. 5.1. *The concentration (from* 0 *to* 1*) as solved by a formally second order Godunov method using* (a) 50×50 *and* (b) 100×100 *rectangular grids at the final time* $T = 1200$ *with the CFL limited time steps* $\Delta t = 0.965$ *and* 0.242*, respectively. The concentration scale is shown in* (c).

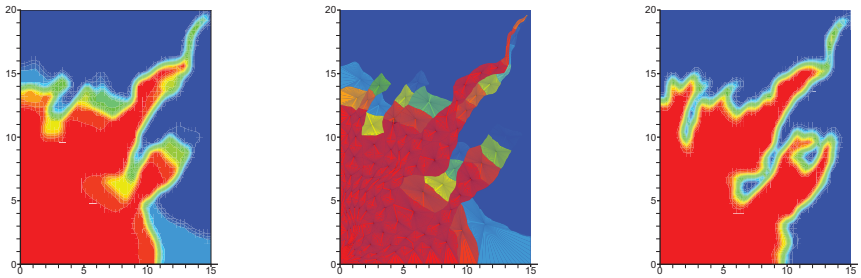
where the sum is over the rectangular grid cells. The operator breaks into a convex sum of two parts, since $0 \le \alpha_G^n \le 1$. When $\alpha_G^n = 0$, we have only the first term, which restores mass back to the stream-cell proportional to its contribution from the forward transfer. This part of the operator is nondiffusive, and it is the entire operator when there is no diffusion ($D = 0$ and $\breve{c}_G^{n+1} = c_G^n$). When $\alpha_G^n = 1$, we have only the second term, which distributes mass in G to the stream-cell proportional to its area. This is the appropriate grid transfer for diffused mass. The full operator, as explained in the next section, is a combination of these two operators, accounting for the artificial grid transfer diffusion created by the original forward transfer (5.2) between the stream-cell and rectangular grids.

Except for the modification of a locally linear transfer operator in subsection 5.4, the description of the diffusion step is complete.

5.2. Numerical results for our main test example. We finally report our first stream-tube results in this subsection. For reference, we solved our transport problem using a formally second order Godunov method [10, 1] on a rectangular grid to a final time of $T = 1200$ using the CFL limited time step. The concentration is shown in Figure 5.1. We use both a 50×50 grid with time step $\Delta t = 0.965$ and a 100×100 grid with time step $\Delta t = 0.242$. We compare the stream-tube results with these results.

The domain is initially free of tracer, meaning that $c_S^0 = 0$ for all stream-cells S. From the injection well in the lower left corner of the domain, pure tracer (of concentration 1) is injected. It migrates through time to the upper right corner of the domain, where the production well resides. Our plots show the concentration over the domain at a time that corresponds approximately to breakthrough (the time at which tracer enters the production well), since at that time the tracer plume is well developed.

In Figures 5.2(a)–(b), we report our stream-tube results using the 10 stream-tubes depicted in Figure 4.1, which has about 20 stream-cells in each stream-tube, for about only 200 total elements. Plot (a) shows the concentration as transferred to the 50×50 rectangular mesh used for the reference solution, and plot (b) shows a very sharp concentration profile on the stream-tube mesh. The time step used is considerably larger than that used by the reference solution: we use $\Delta t = 30 \approx 31.1 \Delta t_{\mathrm{CFL}}$. Since the time step is so large, there is less numerical diffusion compared to the Godunov result in Figure 5.1(a), which uses 2500 elements. Therefore, any numerical diffusion



(a) 10 *stream-tubes, each with* 20 *stream-cells,* $\Delta t = 30$*, rectangular grid (smoothed)*

(b) *Same as* (a)*, but shown on the stream-tube mesh (unsmoothed)*

(c) 40 *stream-tubes, each with* 60 *stream-cells, and* $\Delta t = 30$*, rectangular grid (smoothed)*

FIG. 5.2. *The concentration (from* 0 *to* 1*) as solved by the stream-tube method using constant transfer and* (a)*,* (b) 10 *stream-tubes with about* 20 *stream-cells each and* (c) 40 *stream-tubes with about* 60 *stream-cells each at the final time* $T = 1200$*, with* $\Delta t = 30$ *and* 20*, respectively. Plots* (a) *and* (c) *are shown on the rectangular grid, visually smoothed. Plot* (b) *is shown on the stream-tube mesh, and the concentration is shown as an unsmoothed piecewise constant function.*

caused by transfer to and from the rectangular diffusion grid seems to be a minor issue in this example.

In Figure 5.2(c), we use a refined stream-tube mesh of 40 stream-tubes, each with about 60 stream-cells, for a total of 2400 elements. The time step is still $\Delta t = 30$. Note that we have about the same number of elements as in Figure 5.1(a), but the stream-tube result is superior in terms of the amount of numerical diffusion, and so it shows very good details of the fingering effect. It is comparable and perhaps superior in quality to the 100×100 (10,000 element) reference solution shown in Figure 5.1(b).

5.3. Theoretical justification of the grid transfer operators. As stated in the following lemma, our transfer operator has the first three of the four properties we set out to obtain at the beginning of this subsection.

LEMMA 5.1. *Let the forward and inverse locally constant transfers be defined by* (5.2) *and* (5.3)–(5.5)*, respectively.*

(i) *If the input concentrations are nonnegative, then both transfers result in nonnegative concentrations.*

(ii) *If* $D = 0$*, then the composition of the forward and inverse transfer is the identity.*

(iii) *If a conservative, maximum principle preserving diffusion operator is applied between the two transfer operators, then mass is nonnegative and conserved.*

Proof. The nonnegativity of the forward and inverse transfer is clear, since the transfer coefficients are nonnegative.

We suppress the time level (superscript n) within the proof. Without diffusion, $\check{c}_G = c_G$ and $\alpha_G = 0$, and so the composition of the two operators results in

$$\check{c}_S = \sum_G \rho_{SG}\, c_G = \sum_{G, c_G \neq 0} \frac{|G \cap S|}{|S|} c_S = \sum_G \frac{|G \cap S|}{|S|} c_S = c_S,$$

since if $c_G = 0$, then for a given stream-cell S, either $|G \cap S| = 0$ or $c_S = 0$.

The nonnegativity is clear by (i) and the hypothesis. Let the total initial mass be

$$M = \sum_S c_S\, |S| = \sum_S \left(\sum_G \frac{|G \cap S|}{|S|} \right) c_S\, |S| = \sum_G \left(\sum_S \frac{|G \cap S|}{|G|} c_S \right) |G| = \sum_G c_G\, |G|,$$

which is preserved after the forward transfer. Let $\breve{c}_G$ be the grid concentrations after applying the forward transfer and the diffusion operator. The final mass after applying the inverse operator is

$$\begin{aligned}
\breve{M} = \sum_S \breve{c}_S\,|S| &= \sum_S \left(\sum_G \frac{|G\cap S|}{|S|} \big[(1-\alpha_G)\rho_{SG} + \alpha_G\big] \breve{c}_G \right) |S| \\
&= \sum_{G, c_G \neq 0} \sum_S |G\cap S| \left[(1-\alpha_G)\frac{c_S}{c_G} + \alpha_G \right] \breve{c}_G + \sum_{G, c_G = 0} \sum_S |G\cap S| \breve{c}_G \\
&= \sum_{G, c_G \neq 0} \left\{ \left(\sum_S \frac{|G\cap S|}{|G|}\, c_S \right) \frac{(1-\alpha_G)}{c_G} + \alpha_G \right\} \breve{c}_G\,|G| + \sum_{G, c_G = 0} \breve{c}_G\,|G| \\
&= \sum_{G, c_G \neq 0} \big\{ (1-\alpha_G) + \alpha_G \big\} \breve{c}_G\,|G| + \sum_{G, c_G = 0} \breve{c}_G\,|G| \\
&= \sum_G \breve{c}_G\,|G| = M,
\end{aligned}$$

because the diffusion operator is conservative. ∎

5.3.1. A heuristic model of diffusion. The fourth and final desired property, that the entire step should properly model diffusion as a physical process, will be shown heuristically. Thus for the rest of this section, we relax strict mathematical rigor and argue formally. The difficulty with the diffusion step is that the forward grid transfer (5.2) is itself diffusive. Thus a simple inverse transfer (say, $\alpha_G^n = 1$ in (5.5)) would provide a physically improper sum of the physical and numerical diffusions.

We argue locally near some stream-cell S and rectangular grid cell G, and we omit the time level (superscript n) in our argument. We model the grid transfer diffusion by the heat equation

$$\frac{\partial c_G}{\partial t} - D_{\text{grid}} \Delta c_G \approx 0$$

from the initial condition given by c_S, with a diffusion coefficient D_{grid} chosen so that we obtain spreading on the order of $|G|^{1/2}$, the size of the grid cells, since mass can spread only this far by the forward transfer operator. That is, since the diffusion length is $2\sqrt{D_{\text{grid}}\, t_{\text{grid}}}$, we define

$$D_{\text{grid}} = \frac{|G|}{4\, t_{\text{grid}}},$$

where t_{grid} is some "grid" time scale. In terms of this heat equation, over a "time step" of size t_{grid}, we have approximately that

$$c_G \approx c_S + D_{\text{grid}}\, t_{\text{grid}} \frac{\partial^2_{|S|} c_S}{|S|} \approx c_S + \frac{|G|}{4\,|S|}\, \partial^2_{|S|} c_S,$$

where $\partial^2_{|S|}$ is a standard second order difference (and $|S|$ is the length squared factor).

Next, the physical diffusion operator is applied to c_G. We again model this process as a simple heat equation, at least locally, but with the physically relevant diffusion coefficient D, taken here to be the constant d_G defined in (5.4). Thus, over the time step Δt,

$$\breve{c}_G \approx c_G + d_G \Delta t\, \frac{\partial^2_{|G|} c_G}{|G|}.$$

Combining with the previous result, we have that

$$\begin{aligned}\breve{c}_G \approx c_G + \frac{d_G\Delta t}{|G|}\,\partial^2_{|G|}c_G &\approx \left(c_S + \frac{|G|}{4\,|S|}\,\partial^2_{|S|}c_S\right) + \frac{d_G\Delta t}{|G|}\,\partial^2_{|G|}\left(c_S + \frac{|G|}{4\,|S|}\,\partial^2_{|S|}c_S\right)\\ &= c_S + \frac{|G|}{4\,|S|}\,\partial^2_{|S|}c_S + \frac{d_G\Delta t}{|G|}\,\partial^2_{|G|}c_S + \frac{d_G\Delta t}{4\,|S|}\,\partial^2_{|G|}\partial^2_{|S|}c_S\\ &\approx c_S + \left(\frac{|G|}{4\,|S|} + \frac{d_G\Delta t}{|S|}\right)\partial^2_{|S|}c_S,\end{aligned}$$

wherein we have made two additional approximations. First, we replaced $\partial^2_{|G|}c_S/|G|$ by $\partial^2_{|S|}c_S/|S|$, since these are both approximations of the Laplace operator. Second, we dropped the fourth order difference term, since its characteristic diffusion length scales as the fourth root rather than the square root.

Finally, the inverse transfer is applied to obtain

$$\begin{aligned}\breve{c}_S &= \sum_G \frac{|G\cap S|}{|S|}\left[(1-\alpha_G)\rho_{SG} + \alpha_G\right]\breve{c}_G\\ &\approx \sum_G \frac{|G\cap S|}{|S|}\rho_{SG}\,(1-\alpha_G)\,\breve{c}_G + \sum_G \frac{|G\cap S|}{|S|}\alpha_G\left[c_S + \left(\frac{|G|}{4\,|S|} + \frac{d_G\Delta t}{|S|}\right)\partial^2_{|S|}c_S\right]\\ &\approx \sum_G \frac{|G\cap S|}{|S|}\left((1-\alpha_G)\frac{\breve{c}_G}{c_G} + \alpha_G\right)c_S + \sum_G \frac{|G\cap S|}{|S|}\alpha_G\left(\frac{|G|}{4\,|S|} + \frac{d_G\Delta t}{|S|}\right)\partial^2_{|S|}c_S.\end{aligned}$$

The first term restores mass back to the original stream-tube mesh without diffusion, so the second term needs to be adjusted to attain the correct level of diffusion. To this end, if we solved the diffusion problem on the stream-tube mesh without any grid transfers (as our approximate heat equation), we would obtain

$$\breve{c}_S \approx c_S + \frac{d_G\Delta t}{|S|}\partial^2_{|S|}c_S,$$

and so we need to define α_G so that

$$\frac{d_G\Delta t}{|S|} = \alpha_G\left(\frac{|G|}{4\,|S|} + \frac{d_G\Delta t}{|S|}\right);$$

that is, (5.4) follows.

5.3.2. Numerical tests of the diffusion approximation. We now provide a numerical study of the performance of our treatment of diffusion, especially with respect to the grid transfer error. We reemphasize that our approach is designed for the case of small physical diffusion and that a more computationally expensive direct approximation on the stream-tube mesh is an alternative when high-fidelity simulation of the diffusion operator is required.

Shown in Figure 5.3 is a test example using the stream-tube mesh from our main test example (Figure 4.1). We set the velocity to zero, so there is no advection, and therefore also there is only molecular diffusion $D = d_{\text{mol}} = 10^{-3}$ in (5.1). Initially, the domain has no tracer except in the central stream-cell, where the concentration is 1. Over time, the tracer spreads into the surrounding domain. We use a time step of $\Delta t = 30$.

To start the test, we need to transfer the tracer to the rectangular grid using (5.2). The top row of Figure 5.3 is the reference solution on the 50×50 rectangular

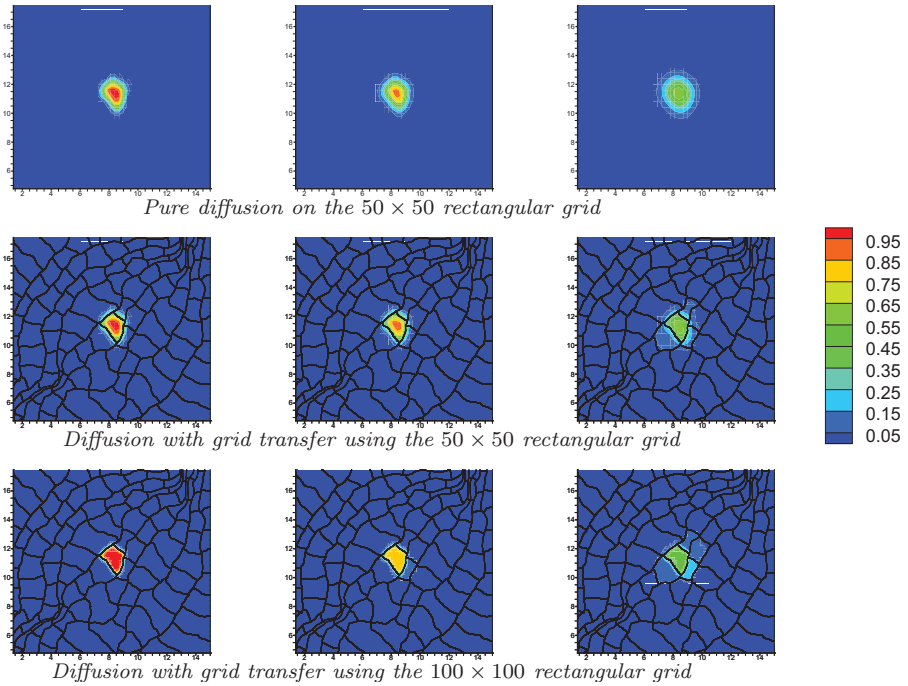


FIG. 5.3. *A test of diffusion from a central stream-cell originally at concentration* 1 *in an otherwise clean domain. From left to right are time steps* 1, 2, *and* 5 *(using* $\Delta t = 30$*). The results are magnified and shown on the rectangular grid, beginning from the concentration after forward transfer. The top set of plots shows the pure diffusion reference case; that is, the concentration is* never *transfered back to the stream-tube mesh. The other two sets show the case of cycling between one step of the diffusion operator, transfer to the stream-tube mesh, and forward transfer back to the rectangular grid. The middle set uses the* 50×50 *rectangular grid for the diffusion, and the bottom set uses the* 100×100 *rectangular grid.*

grid. It is the pure diffusion case without grid transfer, given by solving (3.3)–(3.4). We show time steps 1, 2, and 5. Note especially the size and shape of the plume.

The second row of Figure 5.3 is our proposed method of handling diffusion. Here each time step consists of three parts. First is an application of the diffusion operator solving (3.3)–(3.4) as for the reference solution. But instead of proceeding to the next diffusion step directly, we move the tracer to the stream-tube mesh using the inverse transfer (5.5). This is inherently numerically diffusive. We finally transfer the tracer back to the 50×50 rectangular grid using (5.2). In this case, $\alpha_G = \alpha = 0.5$, so we correct about half the grid diffusion, and the plume size and shape agree well with the reference.

The third row of Figure 5.3 is the same as the second, except that the underlying rectangular grid is refined to 100×100. Now $\alpha_G = \alpha = 0.8$, and more of the diffusion operator is used directly. Similar results are obtained (although the diffusion is resolved a bit better on this finer grid).

Figure 5.4 is a second test of the diffusion, with the same parameters but a modified stream-tube mesh. A very thin stream-tube is added, with a very small central stream-cell containing the initial tracer. These concentrations are shown on a log scale, so smaller concentrations can be seen more easily. Results similar to the previous test case are obtained.

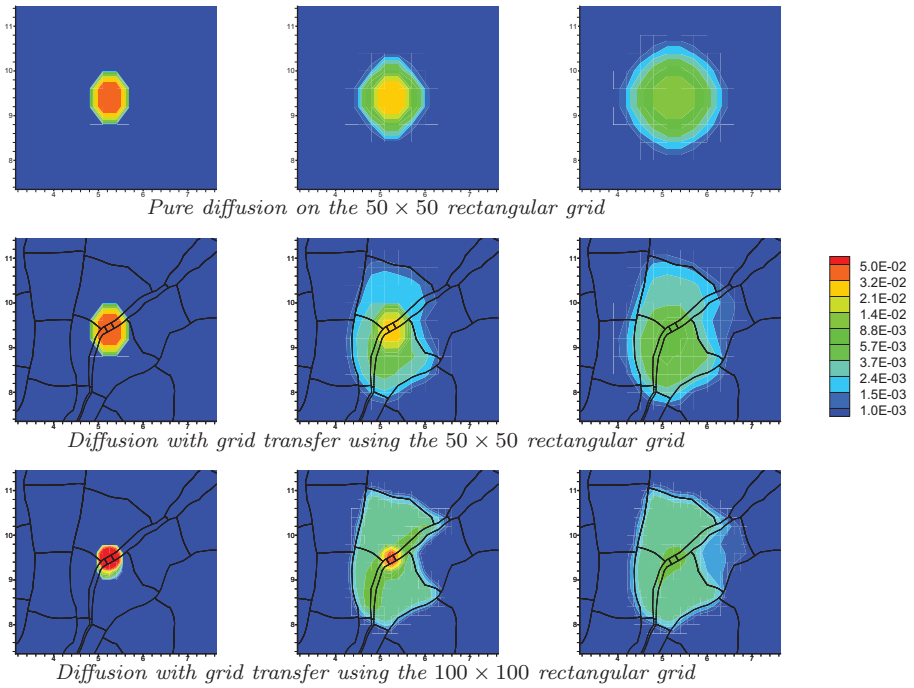


FIG. 5.4. *A second test of diffusion from a much smaller central stream-cell originally at concentration* 1 *in an otherwise clean domain. From left to right are time steps* 1, 2, *and* 5 *(using* $\Delta t = 30$*). The results are magnified and shown using a log scale on the rectangular grid, beginning from the concentration after forward transfer. The top set of plots shows the pure diffusion reference case. The other two sets show the case diffusion with grid transfer. The middle and bottom sets use the* 50×50 *and* 100×100 *rectangular grids, respectively.*

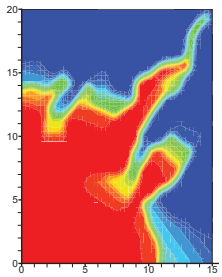


FIG. 5.5. *The concentration from the stream-tube method using linear transfer on a* 10 *stream-tube mesh with about* 20 *stream-cells each with* $T = 1200$ *and* $\Delta t = 30$.

5.4. A locally linear transfer operator. It is easily seen in Figure 5.2(a) that there is a lot of numerical diffusion near the bottom right corner of the domain. This is caused mainly by the extremely large stream-cell in that corner (see Figure 4.1(a)) but also by an interaction with the physical diffusion step. The problem is that a simple constant transfer from the stream-tube mesh to the rectangular grid places a lot of mass too far advanced within the stream-tube, which diffuses downstream. To help alleviate this problem, we devise a locally linear transfer operator. Figure 5.5 shows an improvement at the bottom right corner when the linear transfer operator is applied.

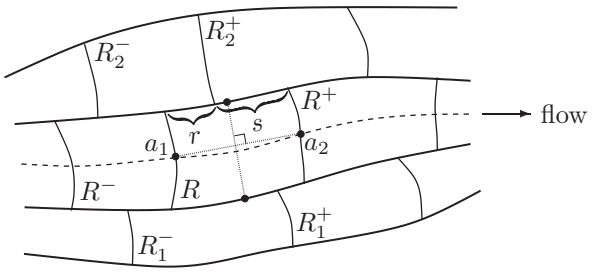


FIG. 5.6. *The linear transfer operator is based on a linear reconstruction depicted here.*

The linear transfer operator is defined as follows. For each stream-cell R, we begin by defining a locally piecewise linear reconstruction of the concentration, similar to that constructed on the reference stream-cell $\bar{R}$ in subsection 4.4. Here we reconstruct on the actual stream-tube, as depicted in Figure 5.6. For time level t^n, we define

$$\tilde{\bar{c}}_R^n(x) = c_R^n + \sigma_R^n \, \mathbf{a} \cdot (x - \bar{x}_R) + \tau_R^n \, \mathbf{a}^\perp \cdot (x - \bar{x}_R), \tag{5.6}$$

where σ_R^n is defined in (4.11) and

$$\bar{x}_R = \frac{1}{|R|} \int_R x \, dx$$

is the average of x on R, so as to preserve mass. To define the other quantities, we let $a_1 \in \Omega$ be the most upstream center-line point of R and $a_2 \in \Omega$ the most downstream point. The vector $\mathbf{a}$ is a unit vector along the flow defined as

$$\mathbf{a} = \frac{a_2 - a_1}{|a_2 - a_1|},$$

and the unit vector $\mathbf{a}^\perp$ is orthogonal to $\mathbf{a}$.

The slope τ_R^n can be set to zero or it can be used to approximate the transverse variation of the concentration. In the latter case, we identify the two stream-tubes adjacent to the one containing R as stream-tubes 1 and 2, where $\mathbf{a}^\perp$ points from stream-tube 1 to stream-tube 2. These stream-tubes exist unless we are near the boundary, in which case we can identify the missing stream-tube with the one containing R.

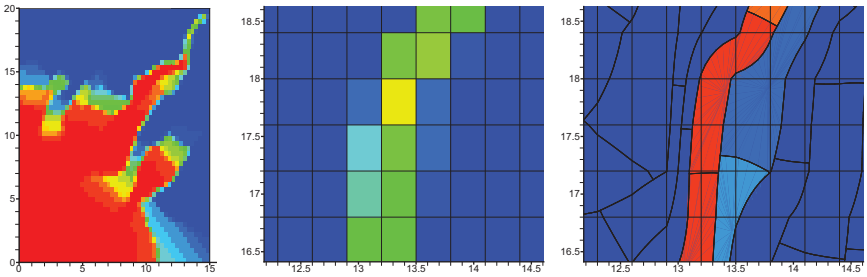
We define the concentration c_2^n associated with stream-tube 2 as a weighted average of its concentration values. The weighting is in terms of the arc length of the common streamline boundary between R and streamline 2. In the case depicted in Figure 5.6, we would set

$$c_2^n = \frac{r \, c_{R_2^-}^n + s \, c_{R_2^+}^n}{r + s}.$$

Similarly we define c_1^n, and finally

$$\tau_R^n = \frac{c_2^n - c_1^n}{k_{\bar{R}}},$$

where, we recall, $k_{\bar{R}}$ is the width of the reference cell $\bar{R}$ of R.



(a) *Piecewise constant* (b) *Rectangular grid* (c) *Stream-tube mesh*

FIG. 5.7. *Nonmonotonicity at* (13.35, 17.8) *shown as piecewise discontinuous constant concentration values* (a) *over the domain on the rectangular diffusion grid,* (b) *close-up on the rectangular diffusion grid, and* (c) *close-up on the stream-tube mesh.*

Once the linear reconstruction is complete, we can simply transfer the mass to the rectangular grid via

$$c_G^n = \sum_S \frac{1}{|G|} \int_{S\cap G} \tilde{\tilde{c}}_S^n(x)\, dx \tag{5.7}$$

in place of (5.2). The inverse transfer is defined similarly to the locally constant case (5.3)–(5.5), but now we keep (5.4) and replace (5.5) by

$$c_S^{n+1} = \breve{c}_S^{n+1} = \sum_G \frac{1}{|S|} \left[(1-\alpha_G^n) \frac{1}{c_G^n} \int_{G\cap S} \tilde{\tilde{c}}_S^n(x)\, dx + \alpha_G^n\, |G\cap S|\, \breve{c}_G^{n+1} \right]. \tag{5.8}$$

The proof of Lemma 5.1 can be modified easily to give a similar result for the linear transfer. The only difficulty is showing that $c_G = 0$ implies that $\int_{S\cap G} \tilde{\tilde{c}}_S^n(x)\, dx = 0$, which is trivial under the added assumption that $\tilde{\tilde{c}}_S^n(x) \geq 0$. We should require this relation anyway, and it is achieved through limiting the slopes in the definition (5.6).

LEMMA 5.2. *Let the forward and inverse locally linear transfers be defined by* (5.7) *and* (5.8) *with* (5.4), *respectively. If* $\tilde{\tilde{c}}_S^n(x) \geq 0$, *then the conclusions of Lemma* 5.1 *hold for the locally linear transfer operators.*

5.5. Aspects of monotonicity. Careful consideration of Figure 5.2(a) shows that the concentration appears to be nonmonotone at around $(x, y) = (12.7, 15.5)$. The solution is, in fact, monotone there, and the apparent nonmonotonicity is due to the smoothing algorithm of the graphical plotting software. We can remove it by viewing the piecewise constants defining the numerical concentration, as shown in Figure 5.7(a). However, when we do so, we discover a genuine nonmonotonicity at (13.35, 17.8), as shown enlarged in Figure 5.7(b).

Figure 5.7(c) explains the situation on the stream-tube mesh. The concentration is monotonic on the stream-tube mesh used for advection, as we expect. (The advection algorithm is both stable and monotonic—see [6].) However, the transfer operation creates a nonmonotonicity on the rectangular grid. The key facts are that the stream-tube width is smaller than the rectangular grid in the region of the domain near (13.35, 17.8) and that the stream-cell has a higher concentration than the stream-cells around it. When we apply the transfer operator from the stream-tube mesh to the rectangular grid, we inevitably create a slight nonmonotonicity. The use of the linear transfer operator, as in Figure 5.5, will lessen the nonmonotonicity, but

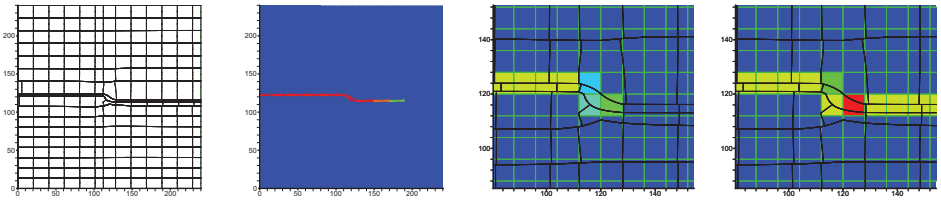


FIG. 5.8. *Test example of monotonicity using* $D = d_{mol} = 10^{-10}$. *Left is the stream-tube mesh, followed by the concentration shown on the stream-tube mesh at step* 9 *(plotted using the usual linear scale, Figure* 5.1(c)*). It remains monotone. The final two plots are magnified images of the concentration after transfer to the rectangular grid. Here the concentrations are clearly nonmonotone.*

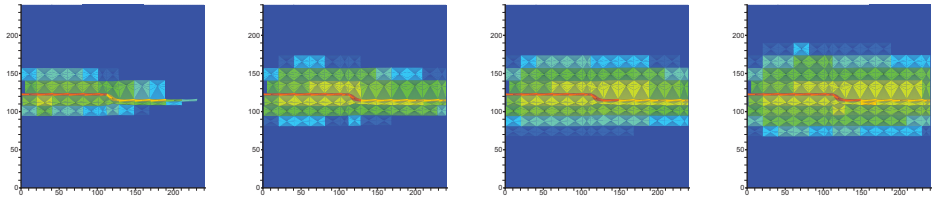


FIG. 5.9. *Test example of monotonicity using* $D = d_{mol} = 10^{-2}$. *The tracer concentration is plotted on a log scale (from* 10^{-3} *to* 1*). The concentration within the stream-tube remains monotone when shown on the stream-cell mesh.*

it cannot correct it. Neither can local stream-tube refinement, described in the next subsection, since the nonmonotonicity comes not from an insufficiently fine grid but rather from the way the stream-tubes and stream-cells intersect with the rectangular grid. Refining the rectangular grid used for diffusion would alleviate the problem. However, this nonmonotonicity is probably within the range of acceptable approximation error. It is both very slight and not expected to increase with time, since both the advection and diffusion steps are stable and the diffusion step is smoothing.

We present now two contrived test examples to further exhibit the problem of nonmonotonicity in a simple setting using only molecular diffusion in (5.1). On the left in Figure 5.8, we show a 240×240 domain with a single fine stream-tube with a kink, and tracer is injected only into that single stream-tube. In the first test, the diffusion is very small, $D = d_{\mathrm{mol}} = 10^{-10}$. Forward grid transfer near that kink must result in nonmonotone concentrations on the 30×30 rectangular grid, as seen magnified at two times (steps 6 and 9 using $\Delta t = 500$) in the right-most plots in Figure 5.8. However, the concentration within the stream-tube remains monotone, as seen for the later test in the second plot in Figure 5.8. In this test, $\alpha = \alpha_G \approx 0$ for the inverse transfer operator (5.4). The second test is similar, except that now $D = d_{\mathrm{mol}} = 10^{-2}$, so there is significantly more physical diffusion, and $\alpha = \alpha_G \approx 0.24$. It is shown for four times in Figure 5.9, plotted on a log scale. Again, the concentration within the stream-tube remains monotone. While we do not claim that monotonicity cannot be lost, we do see that it does not seem to be a significant problem.

We close this discussion of nonmonotonicity by pointing out that it could occur whenever we transfer results to a rectangular grid. So, for example, even if we solved the diffusion step on the stream-tube mesh, we would see this nonmonotonicity if we wished to plot the concentration on a rectangular grid.

TABLE 6.1
CPU timing results (in seconds) for the stream-tube and VCCMM algorithms.

Method	Δt used	Stream-cell or trace-back mesh generation	Advection steps	Diffusion steps	Grid transfer
10 stream-tubes					
$N = 5$	30	9.52	0.063	8.200	7.894
$N = 10$	30	21.62	0.090	8.182	8.119
$N = 50$	30	109.96	0.164	8.282	8.550
40 stream-tubes					
$N = 10$	30	87.80	0.732	8.218	66.585
VCCMM					
	3	86.29	24.767	81.038	——
	6	96.38	12.712	40.964	——

6. Performance of the basic algorithm. The advection step in section 4 and the diffusion step in section 5 give the basic algorithm for the fully conservative Eulerian-Lagrangian stream-tube method. The next two sections will discuss extensions of the algorithm to more complex situations. We record now that the method is both mass conservative and preserves positivity of the concentration, meaning that undershoots cannot occur.

LEMMA 6.1. *Assume that a positivity preserving method is used to solve the diffusion step on either the stream-tube mesh or a rectangular mesh. In the latter case, if the linear transfer operator is used, assume also that* $\tilde{\tilde{c}}_S^n(x)$ *defined by* (5.6) *is slope limited so that* $\tilde{\tilde{c}}_S^n(x) \geq 0$. *If each stream-tube concentration* $c_S^n \geq 0$, *then also* $c_S^{n+1} \geq 0$. *Assuming* $c^0 \geq 0$, *the fully conservative Eulerian-Lagrangian stream-tube method both conserves mass and produces nonnegative concentrations.*

Proof. By construction, the advection step is mass conservative. We have noted that the diffusion step is mass conservative in Lemmas 5.1 and 5.2, and so the overall method is also mass conservative.

For the positivity, we first note that $\tilde{c}_{\bar{R}}^n$ defined by (4.9) on the reference stream-cell $\bar{R}$ is nonnegative. This is due to the MUSCL slope limiter (4.10)–(4.11). The advection step (4.14) then clearly defines c_R^{n+1} as nonnegative. (See especially the second equality in (4.14).) The diffusion step preserves positivity by Lemmas 5.1 and 5.2, so every stage of the method preserves nonnegativity, and thus the method itself does the same. ∎

6.1. A comparison to the VCCMM. We now compare the efficiency of the stream-tube method versus the VCCMM [3], which works completely on the 50×50 rectangular grid. The CPU timing results are reported in Table 6.1 for our simulation up to the final time of 1200. We include the cases of a stream-tube mesh with both 10 and 40 stream-tubes, using different numbers of maximal points per rectangular grid cell (which we denote by N). In subsection 4.2, we noted that there is no significant difference in the solutions between the $N = 10$ and $N = 50$ cases; however, the $N = 5$ case gives a noticeably poorer solution. The stream-tube mesh generation and advection step times scale with this number, since there are more intersections of the rectangular and stream-tube meshes when using more points to define the streamlines. This number also slightly increases the time to compute all the advection steps.

The VCCMM is run with two time steps, having CFL numbers 4.2 ($\Delta t = 3$) and 8.4 ($\Delta t = 6$). The trace-back mesh generation increases only a little for the longer time step, and the advection steps take roughly half the time.

Comparing the stream-tube method and VCCMM, we see that the mesh generation may be more efficient for the stream-tube mesh, since we get good results for coarser meshes. If we compare VCCMM to the 40 stream-tube case, we have comparable spatial resolution and comparable mesh generation times.

The advection stepping time shows that the stream-tube method is more efficient. First, we can get good results using much larger Δt, so we need only 5 to 10 times fewer steps. But also the time per step is much improved, since the stream-tubes are essentially one-dimensional but the VCCMM is two-dimensional. For example, the highly refined 40 stream-tube case uses only 0.018 seconds per advection step, versus 0.064 for VCCMM with the larger time step.

Finally, the diffusion step is comparable for the two methods. Our reported times simply reflect the use of a diffusion step for every advection step, which is not necessary. So the stream-tube method uses only 40 diffusion steps, but the VCCMM uses 400 or 200. The main difference here is the use of the grid transfer operators, which are unnecessary in the VCCMM.

6.2. A comparison to Godunov's method. Finally, we consider the error resulting from a simple radial test example. The velocity is given by

$$\mathbf{u} = \frac{(x, y)}{x^2 + y^2}.$$

Initially $c^0 = 0$, and the tracer enters the domain through a well in the corner of the domain $(0, 20)^2$ at $(0, 0)$. Since there is no diffusion ($D = 0$), the tracer concentration is the characteristic function of a circle, the radius of which is determined by mass conservation.

The numerical velocity used is the Raviart–Thomas projection of $\mathbf{u}$; that is, we only use the average normal velocity on each edge of a fixed rectangular grid (in our case, 50×50). Therefore, the stream-tube mesh is *not* given by a simple radial mesh in polar coordinates (although it is close to this). Instead, the stream-tube mesh is constructed as described in subsections 4.1–4.2. Stream-tubes are generated from points spaced equally distant on the well boundary (taken as the corner element) and so are not equally spaced in arc-length. The boundaries of the stream-tubes are not perfectly straight due to tracing using the approximate Raviart–Thomas velocity. The stream-cells have length determined by equal spacing along the center streamlines, which vary from stream-tube to stream-tube. Moreover, the transverse boundaries of the stream-cells are given by our transverse tracing algorithm, and so they are not quite circular arcs.

The stream-tube method is used to solve the problem twice. In the first case, 20 tubes with between 17 and 21 cells per tube are used, for a total of about 400 stream-cells, and the time step is $\Delta t = 1.0$. We also use $N = 6$ maximal points per rectangular grid cell (and a tracing time step of 0.6). In the second case, 40 tubes and 32 to 42 cells per tube are used, for a total of about 1600 stream-cells, and the time step is $\Delta t = 0.5$. Here, $N = 12$ (and the tracing time step is 0.3). These results are contrasted with the second order Godunov method [10, 1] using a 50×50 rectangular grid of 2500 grid cells. The time step used is $\Delta t = 0.1$, which is about the maximum CFL step allowed.

The L^1 error is given in Table 6.2 at four times. We see that the stream-tube method can represent the solution more accurately for a given number of grid cells and a much larger time step in this simple problem. We also report the CPU time used for the simulation to the final time $t = 100$. The stream-tube method is significantly

TABLE 6.2

Accuracy results (in L^1) and CPU timings for the stream-tube and second order Godunov algorithms. The setup time refers to the stream-tube mesh generation, and the advection time is the time for completing the simulation to time $t = 100$.

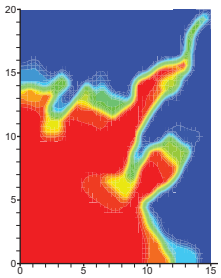
Method	Δt	Number of cells	L^1 error at time 25	50	75	100	CPU time (s) Setup	Advection
Stream-tube								
20 tubes	1.0	400	6.47	7.62	10.10	11.36	4.72	0.120
40 tubes	0.5	1600	2.80	4.81	5.91	6.65	99.86	0.736
Godunov	0.1	2500	4.33	6.88	9.13	11.10	——	2.732

faster per step, since the mesh is so much coarser (and the problems are essentially one-dimensional). Of course, the stream-tube method has a significant setup time for generating the mesh, so it is more efficient only for long-time simulations to at least $t = 200$ in this example.

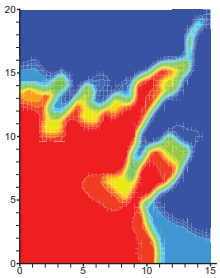
7. Local stream-tube refinement. Since the stream-tubes are independent of each other in the advection step, it is relatively easy to apply local mesh refinement. We explore two types of refinement: refinement within a stream-tube by adding more stream-cells, and stream-tube refinement, i.e., adding more stream-tubes.

We already noted the very large numerical diffusion in the bottom right corner of the domain in Figure 5.2(a), due to the extremely large stream-cells in the stream-tube mesh Figure 4.1(a). We computed this example on a refined mesh with only the bottom right stream-tube being refined to 46 stream-cells. In Figure 7.1(a), we see the result becomes much less numerically diffusive.

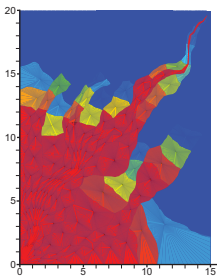
In Figures 7.1(b)–(c), we show the result of adding three stream-tubes to the mesh, refining the two right-most stream-tubes on the top injection well boundary and the top-most stream-tube on the right injection well boundary, i.e., stream-tubes 4 to 6 counting from the left in Figure 4.1. It is obvious that we improve resolution at around $(5, 12)$ and at the tip of the tracer finger near the production well. In fact, comparing Figures 7.1(c) and 5.2(b) with 5.1(b), it is remarkable that this finger has been resolved to so thin a stream-tube on a mesh of only about 260 total elements.



(a) *Stream-cells added, rectangular grid*



(b) *Stream-tubes added, rectangular grid*



(c) *Same as* (b)*, but shown on the stream-tube mesh*

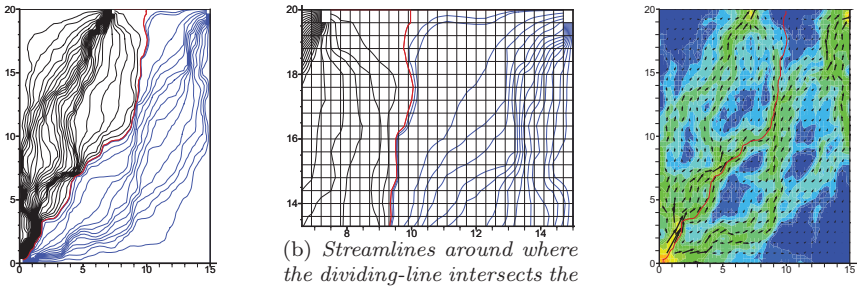
FIG. 7.1. *The concentration (from* 0 *to* 1*) using the same* 10 *stream-tubes, each with about* 20 *stream-cells, as used in our test example and depicted in Figure* 4.1*, except that in* (a) *we refine the bottom right stream-tube to* 46 *stream-cells, and in* (b)–(c) *we refine stream-tubes* 4 *to* 6 *(from the left) into two stream-tubes each, for a total of* 13 *stream-tubes. Results compare to Figures* 5.2(a)–(b).

8. Multiple pressure basins. Streamlines travel between an injection and a production well, from high to low pressure. When there is more than one pressure basin, as when the domain contains three or more wells, special care must be employed in the stream-tube method. In that case, the domain is divided into multiple pressure basins, separated by dividing streamlines. It is unwise to define a stream-cell that contains the dividing-line, since there is no well-defined center-line. In fact, most likely either fluid from two injection wells will merge within the tube, or fluid from one injection well will split toward two production wells.

8.1. A simple example of two pressure basins. To assess the stream-tube method, we need to alter our test example. We keep a zero initial condition on the concentration and the injection and production wells in their original positions, but we halve the strength of the producer. We then add a second production well at the top center of the domain of the same strength.

Similar to the original two-well setting, we equally space sample points around the injection well and trace the streamlines through those points via (4.1)–(4.2). We end up with two groups of streamlines, as depicted in Figure 8.1(a) from the velocity depicted in Figure 8.1(c). The black set on the left side of the domain shows the streamlines that enter the top center production well, and the blue set shows those streamlines entering the top corner production well. In between is the pressure basin dividing-line, shown in red. Any point located to the left of the dividing-line must pass through a streamline which enters the left-most production well; similarly, those points to the right of the dividing-line must pass through a streamline which enters the right-most well.

There is a point, call it x_0, where the dividing-line intersects the top portion of the domain boundary (as in Figure 8.1(b)). At this point, the velocity must vanish. We have already noted that it is critical to approximate the streamlines accurately in subsection 4.2, and the same is even more true for the dividing streamline. Again using the lowest order Raviart–Thomas mixed finite element space, it actually takes an infinite amount of time to trace to the point x_0. However, this is needless. We can simply determine which rectangular grid cell separates left from right flow, i.e., the cell for which $u_1 < 0$ to the left and $u_1 > 0$ to the right. On that cell, $\mathbf{u} = (a+bx, c+dy)$, so we determine $x_{0,1}$ satisfying $a + bx_{0,1} = 0$. The point $x_0 = (x_{0,1}, 20)$ traces vertically down to the bottom of the cell at $(x_{0,1}, 20 - \ell_2)$. We can then easily continue tracing analytically backward to the injection well. This streamline is the dividing-line.

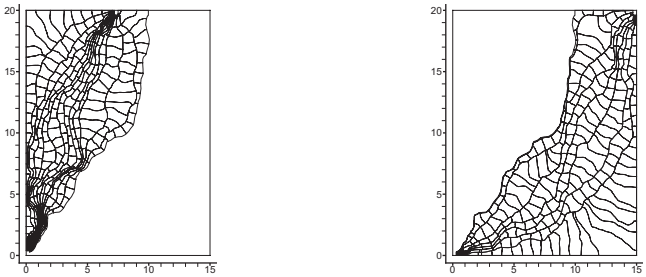


(a) *The streamlines*

(b) *Streamlines around where the dividing-line intersects the domain boundary*

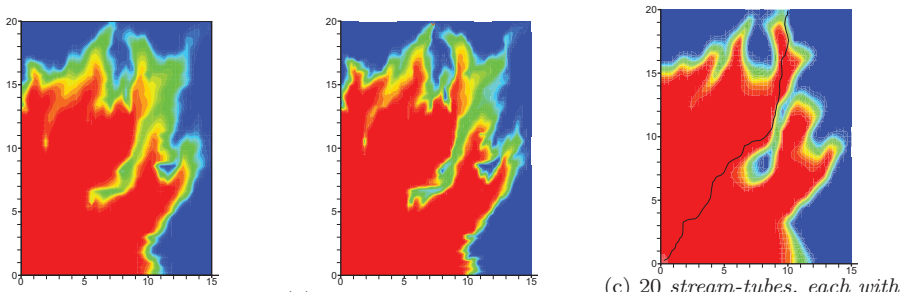
(c) *The velocity*

FIG. 8.1. (a)–(b) *The streamlines used for construction of the stream-tubes in the three-well problem.* (c) *The speed in contour on a log scale with the velocity vectors as arrows. Note the red dividing-line between the two pressure basins.*



(a) *Left domain* (b) *Right domain*

FIG. 8.2. *The stream-tube mesh for the three-well problem, as separated into pressure basins.*



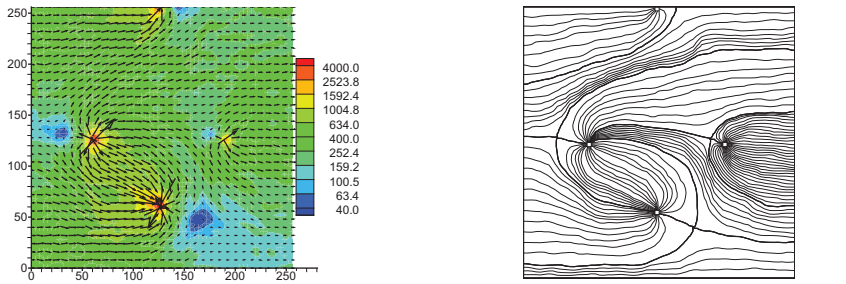
(a) *Godunov,* 50×50 *grid,* $\Delta t = 0.97$ (b) *Godunov,* 100×100 *grid,* $\Delta t = 0.24$ (c) 20 *stream-tubes, each with about* 40 *stream-cells,* $\Delta t = 30$

FIG. 8.3. *The three-well problem concentration (from* 0 *to* 1*) as solved by* (a) *a formally second order Godunov method on a* 50×50 *rectangular grid using the CFL* $\Delta t = 0.968$, (b) *the same method on a* 100×100 *rectangular grid using the CFL* $\Delta t = 0.242$, *and* (c) *the stream-tube method with* 20 *stream-tubes, each with about* 40 *stream-cells, using* $\Delta t = 30$, *with the dividing-line shown. The results are at the final time* $T = 1500$.

Once we have determined the dividing streamline, we use it as one boundary of a stream-tube. In our case, we simply refined the stream-tube within which it lay, and so we have two narrow stream-tubes adjacent to the dividing-line. We show the two pressure basins and our stream-tube mesh in Figure 8.2.

For a single time step using first order Strang splitting, we apply the transport step in each tube, as discussed above. Working over the entire domain Ω, we then transfer the stream-cell concentrations to the rectangular grid and compute the diffusion step as was done previously. Finally, we transfer the grid concentration values back to the stream-cells of the two pressure basins. The results are reported in Figure 8.3(c). For comparison, we report the result of using a formally second order Godunov method [10, 1] on both 50×50 and 100×100 grids using much smaller CFL limited time steps in Figure 8.3(a)–(b). The stream-tube method shows much less numerical dispersion. Moreover, this problem is difficult to approximate well, since the flow is strong along the dividing-line, as seen in Figure 8.1. The streamline method captures the flow on each side of the dividing-line very well.

8.2. An example of multiple pressure basins and the tracing of dividing-lines. In our final numerical example, we consider a case with four wells. One production and two injection wells appear within the domain, and a second production well appears on the boundary of the domain. Moreover, a left to right pressure dif-



(a) *Flow velocity* (b) *Pressure basins and stream-tubes*

FIG. 8.4. *The four-well problem, showing* (a) *the flow speed on a log scale and velocity arrows, and* (b) *the dividing streamlines, pressure basins, and stream-tubes.*

ference is imposed on the boundary of the domain. In all, the domain is divided into eight distinct pressure basins, as shown in Figure 8.4(b). Streamlines can enter the domain either through the boundary of an injection well or through the inflow portion of the domain boundary. Similarly, streamlines can leave the domain either through the boundary of a production well or through the outflow portion of the domain boundary. This numerical example exhibits all four types of pressure basins with streamlines that travel from injection to production well, injection well to outflow boundary, inflow boundary to production well, and inflow to outflow boundary.

To be more specific, the domain is the square $(0,\ell)^2$, where $\ell = 256$. The porosity $\phi = 1$ and the domain is originally free of tracer. One injection well is at $(\ell/4, \ell/2)$ of strength 6.667e4 and the other injection well is at $(3\ell/4, \ell/2)$ of strength 2.667e4. Both inject pure tracer (concentration 1). The first production well is at $(\ell/2, \ell/4)$ of strength -8e4 and the other is at $(\ell/2, \ell)$ of strength -2.667e4. The bottom and top sides of the domain have homogeneous Neumann (no flow) boundary conditions. Dirichlet conditions are imposed on the left and right sides of the domain, with pressure 5e4 and in-flow concentration 1 on the left and -5e4 on the right.

The flow velocity $\mathbf{u}$ is solved using a fine 64×64 uniform rectangular grid. The velocity is shown in Figure 8.4(a). The diffusion/dispersion is applied on the same 64×64 grid using model (5.1) with $d_{\text{mol}} = 1\text{e}{-11}$, $d_{\text{long}} = 1\text{e}{-5}$, and $d_{\text{trans}} = 1\text{e}{-6}$.

We show the evolution of the solution in Figure 8.5, using a time step $\Delta t = 0.045$, which is around 58 times the CFL limited step size. The figures show the solution after 2, 5, and 10 steps. The first set of plots show the concentration at the end of the time step. The second set shows the concentration after the diffusion step on the rectangular grid, before transfer back to the stream-tube mesh. The results show amazing detail, considering that a very coarse mesh and large time steps were used for this very complex flow field.

The stream-tube mesh was constructed as follows. First, the stagnation points of the flow field (i.e., the points where $\mathbf{u} = 0$) need to be determined. Recall that we use the lowest order Raviart–Thomas mixed method on the 64×64 rectangular grid. As noted in subsection 4.2, we can trace analytically within the resulting velocity field. It is also easy to determine the stagnation points within the grid elements, since $\mathbf{u} = (a+bx, c+dy)$. One must also account for stagnation points along the boundaries of the elements.

Once the stagnation points are found, one can trace the streamlines that emanate from these points. We discussed tracing from stagnation points that lie on the bound-

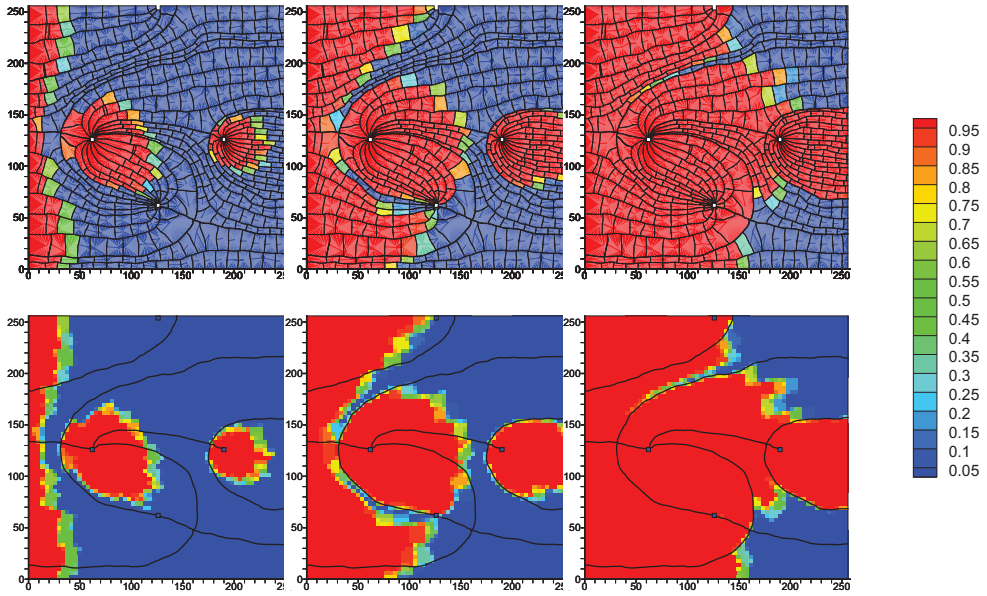


FIG. 8.5. *The four-well problem, showing the concentration at three times, steps* 2, 5, *and* 10, *using* $\Delta t = 0.045$. *The first set of plots show the concentration at the end of the time step. The second set shows the concentration after the diffusion step on the rectangular grid, before transfer back to the stream-tube mesh.*

ary of the domain in the previous subsection. Consider a stagnation point (x_0, y_0) lying within a grid cell $(x_1, x_2) \times (y_1, y_2)$ inside the domain. The Raviart–Thomas velocity is $\mathbf{u} = \big(a(x-x_0), b(y-y_0)\big)$, and so the four streamlines that trace back from the stagnation point pass through (x_1, y_0), (x_2, y_0), (x_0, y_1), and (x_0, y_2). We can continue tracing from these four points, and these curves will define the dividing-lines, partitioning the domain into pressure basins (see Figure 8.4(b)). In our example, there are three stagnation points within the domain and one on the top boundary of the domain. These three trace to a well, to the boundary of the domain, or possibly to another stagnation point (which does not occur in this example).

Finally, within each pressure basin, the stream-tube mesh is constructed. When there is an injection well in the basin, the stream-tube mesh is constructed as described earlier in subsection 4.1. That is, we choose points on the boundary of the injection well, which are traced to either a production well or the outflow boundary. These streamlines form the stream-tubes, and these are further divided into stream-cells as discussed. This same procedure is used in general, tracing from select points on any inflow region, defined either as a boundary of an injection well or as a portion of the inflow boundary of the domain.

9. Conclusions. We presented a new method, the fully conservative Eulerian-Lagrangian stream-tube method, that is the VCCMM combined with the use of a stream-tube mesh. The combined method achieves the main advantages of using both techniques. The VCCMM gives us low numerical diffusion, nonnegative concentrations, and the ability to use very large time steps compared to the CFL limited step. Moreover, it is fully conservative, so both the tracer and ambient fluid mass are locally conserved. This is important in its own right, but also when reactions are considered, since the VCCMM conserves mass and volume, i.e., density.

The use of a stream-tube mesh for the advection computation provides a wide array of advantages. It gives us even lower numerical diffusion than VCCMM alone, since numerical cross-diffusion between stream-tubes is removed entirely. It also allows us to use a coarser mesh, since it is tailored to the flow pattern. The combined method gives an essentially one-dimensional advection problem within each stream-tube. This makes it relatively easy to implement and computationally efficient. In fact, the complex and problem-specific geometric layer-by-layer volume adjustment technique of VCCMM is no longer needed. The new method is easily amenable to local refinement, either in the sense of adding more stream-cells within the stream-tubes or by adding more stream-tubes themselves. Although the new method was presented in two dimensions, in principle it should generalize to three dimensions with the same advantages.

We saw that it was important, especially in the case of multiple pressure basins, to trace the streamlines extremely accurately. This we could do analytically using a Raviart–Thomas mixed method for solving the pressure equation.

The new method uses a Strang splitting between advection and diffusion. The latter can be solved using an efficient diffusion approximation on a rectangular grid. Appropriate grid transfer operators were developed that reduce numerical diffusion due to grid transfer. A slight nonmonotonicity can be seen on the rectangular grid, due to the transfer between meshes. It often does not manifest itself on the stream-tube mesh, and it seems not to be a serious problem.

The combined approach, the fully conservative Eulerian-Lagrangian stream-tube method, has a certain simplicity that gives it the potential to allow generalizations to more complex flows, such as two-phase flows and miscible displacement.

REFERENCES

[1] T. Arbogast, *User's Guide to Parssim*1: *The Parallel Subsurface Simulator, Single Phase*, TICAM Report 98–13, Center for Subsurface Modeling, Texas Institute for Computational and Applied Mathematics, University of Texas at Austin, 1998.

[2] T. Arbogast, A. Chilakapati, and M. F. Wheeler, *A characteristic-mixed method for contaminant transport and miscible displacement*, in Computational Methods in Water Resources IX, Vol. 1: Numerical Methods in Water Resources, T. F. Russell et al., eds., Computational Mechanics Publications, Southampton, UK, 1992, pp. 77–84.

[3] T. Arbogast and C. Huang, *A fully mass and volume conserving implementation of a characteristic method for transport problems*, SIAM J. Sci. Comput., 28 (2006), pp. 2001–2022.

[4] T. Arbogast and C. Huang, *A fully conservative Eulerian-Lagrangian method for a convection-diffusion problem in a solenoidal field*, J. Comput. Phys., 229 (2010), pp. 3415–3427.

[5] T. Arbogast, C. Huang, and T. F. Russell, *A locally conservative streamline method for a model two-phase flow problem in a one-dimensional porous medium*, SIAM J. Sci. Comput., to appear.

[6] T. Arbogast and W.-H. Wang, *Stability, monotonicity, maximum and minimum principles, and implementation of the volume corrected characteristic method*, SIAM J. Sci. Comput., 33 (2011), pp. 1549–1573.

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

[8] T. Arbogast and M. F. Wheeler, *A family of rectangular mixed elements with a continuous flux for second order elliptic problems*, SIAM J. Numer. Anal., 42 (2005), pp. 1914–1931.

[9] T. Arbogast, M. F. Wheeler, and I. Yotov, *Mixed finite elements for elliptic problems with tensor coefficients as cell-centered finite differences*, SIAM J. Numer. Anal., 34 (1997), pp. 828–852.

[10] J. B. Bell, C. N. Dawson, and G. R. Shubin, *An unsplit higher-order Godunov scheme for scalar conservation laws in two dimensions*, J. Comput. Phys., 74 (1988), pp. 1–24.

[11] Y. Béreaux and J. R. Clermont, *Numerical simulation of two- and three-dimensional flows of viscoelastic fluids using the stream-tube method*, Math. Comput. Simulation, 44 (1997), pp. 387–400.

[12] R. Bermejo and J. Carpio, *A semi-Lagrangian-Galerkin projection scheme for convection equations*, IMA J. Numer. Anal., 30 (2010), pp. 799–831.
[13] A. Bermúdez, M. Nogueiras, and C. Vázquez, *Numerical analysis of convection-diffusion-reaction problems with higher order characteristics/finite elements.* I. *Time discretization*, SIAM J. Numer. Anal., 44 (2006), pp. 1829–1853.
[14] A. Bermúdez, M. Nogueiras, and C. Vázquez, *Numerical analysis of convection-diffusion-reaction problems with higher order characteristics/finite elements.* II. *Fully discretized scheme and quadrature formulas*, SIAM J. Numer. Anal., 44 (2006), pp. 1854–1876.
[15] M. A. Celia, T. F. Russell, I. Herrera, and R. E. Ewing, *An Eulerian-Lagrangian localized adjoint method for the advection-diffusion equation*, Adv. Water Resources, 13 (1990), pp. 187–206.
[16] S. Chippada, C. N. Dawson, M. L. Martinez, and M. F. Wheeler, *A projection method for constructing a mass conservative velocity field*, Comput. Methods Appl. Mech. Engrg., 157 (1998), pp. 1–10.
[17] J. R. Clermont and M. E. de la Lande, *Numerical simulation of three-dimensional duct flows of incompressible fluids by using the stream-tube method. Part* I: *Newtonian equation*, Theory Comput. Fluid Dynamics, 4 (1993), pp. 129–149.
[18] H. K. Dahle, R. E. Ewing, and T. F. Russell, *Eulerian-Lagrangian localized adjoint methods for a nonlinear advection-diffusion equation*, Comput. Methods Appl. Mech. Engrg., 122 (1995), pp. 223–250.
[19] J. Douglas, Jr., R. E. Ewing, and M. F. Wheeler, *Approximation of the pressure by a mixed method in the simulation of miscible displacement*, RAIRO Modél. Math. Anal. Numér., 17 (1983), pp. 17–33.
[20] J. Douglas, Jr., and C.-S. Huang, *The convergence of a locally conservative Eulerian-Lagrangian finite difference method for a semilinear parabolic equation*, BIT, 41 (2001), pp. 480–489.
[21] J. Douglas, Jr., F. Pereira, and L.-M. Yeh, *A locally conservative Eulerian-Lagrangian numerical method and its application to nonlinear transport in porous media*, Comput. Geosci., 4 (2000), pp. 1–40.
[22] J. Douglas, Jr., and T. F. Russell, *Numerical methods for convection-dominated diffusion problems based on combining the method of characteristics with finite element or finite difference procedures*, SIAM J. Numer. Anal., 19 (1982), pp. 871–885.
[23] M. Falcone and R. Ferretti, *Convergence analysis for a class of high-order semi-Lagrangian advection schemes*, SIAM J. Numer. Anal., 35 (1998), pp. 909–940.
[24] J. D. Foley, A. V. Dam, S. K. Feiner, and J. F. Hughes, *Computer Graphics: Principles and Practice*, Addison-Wesley, Reading, MA, 1995.
[25] Y. Hasbani, E. Livne, and M. Bercovier, *Finite elements and characteristics applied to advection-diffusion equations*, Comput. & Fluids, 11 (1983), pp. 71–83.
[26] R. W. Healy and T. F. Russell, *Analytical tracking along streamlines in temporally linear Raviart-Thomas velocity fields*, in Computational Methods in Water Resources XIII, vol. 2, Bentley et al., eds., A. A. Balkema, Rotterdam, The Netherlands, 2000, pp. 631–638.
[27] R. W. Healy and T. F. Russell, *Treatment of internal sources in the finite-volume ELLAM*, in Computational Methods in Water Resources XIII, vol. 2, Bentley et al., eds., A. A. Balkema, Rotterdam, The Netherlands, 2000, pp. 619–622.
[28] C.-S. Huang, T. Arbogast, and J. Qiu, *An Eulerian-Lagrangian WENO finite volume scheme for advection problems*, J. Comput. Phys., 231 (2012), pp. 4028–4052.
[29] J. C. Martin and R. E. Wegner, *Numerical solution of multiphase two-dimensional incompressible flow using stream-tube relationships*, Soc. Petrol. Engrg. J., (1979), pp. 313–323.
[30] K. Morton, A. Priestley, and E. Süli, *Stability of the Lagrange-Galerkin method with nonexact integration*, RAIRO Mod. Math. Anal. Numer., 22 (1988), pp. 625–653.
[31] M. Normandin, D. G. Radu, A. Mahmoud, and J. R. Clermont, *Finite element and stream-tube formulations for flow computations—two-dimensional applications*, Math. Comput. Simulation, 60 (2002), pp. 129–134.
[32] O. Pironneau, *On the transport-diffusion algorithm and its applications to the Navier-Stokes equations*, Numer. Math., 38 (1981/82), pp. 309–332.
[33] R. A. Raviart and J. M. Thomas, *A mixed finite element method for* 2*nd order elliptic problems*, in Mathematical Aspects of Finite Element Methods, I. Galligani and E. Magenes, eds., Lecture Notes in Math. 606, Springer-Verlag, New York, 1977, pp. 292–315.
[34] J. E. Roberts and J.-M. Thomas, *Mixed and hybrid methods*, in Handbook of Numerical Analysis, vol. 2, P. G. Ciarlet and J. L. Lions, eds., Elsevier Science, North-Holland, Amsterdam, 1991, pp. 523–639.
[35] G. Strang, *On the construction and comparison of difference schemes*, SIAM. J. Numer. Anal., 5 (1968), pp. 506–517.

[36] B. van Leer, *Towards the ultimate conservative difference scheme.* V. *A second-order sequel to Godonov's methods*, J. Comput. Phys., 32 (1979), pp. 101–136.
[37] H. Wang and M. Al-Lawatia, *A locally conservative Eulerian-Lagrangian control-volume method for transient advection-diffusion equations*, Numer. Methods Partial Differential Equations, 22 (2006), pp. 577–599.
[38] H. Wang, D. Liang, R. E. Ewing, S. L. Lyons, and G. Qin, *An ELLAM approximation for highly compressible multicomponent flows in porous media. Locally conservative numerical methods for flow in porous media*, Comput. Geosci., 6 (2002), pp. 227–251.
[39] Z. X. Chen, Y. R. Yuan, and L. S. Jiang, *Fixed stream-tube method for solving two-phase plane flow problems and its theoretical analysis*, Appl. Math. Mech., 7 (1986), pp. 663–675.