An investigation of Eulerian-Lagrangian methods for solving heterogeneous advection-dominated transport problems

Feng Ruan

Applications Research, GeoQuest, Schlumberger, Houston, Texas

Dennis McLaughlin

Ralph M. Parsons Laboratory, Department of Civil and Environmental Engineering, Massachusetts Institute of Technology, Cambridge

Abstract. Numerical simulation of solute transport in heterogeneous porous media is greatly complicated by the large velocity and concentration gradients induced by spatial variations in hydraulic conductivity. Eulerian-Lagrangian methods for solving the transport equation can give accurate solutions to heterogeneous problems if their interpolation algorithms are properly selected. This paper compares the performance of four Eulerian-Lagrangian solvers that rely on linear, quadratic, cubic spline, and taut spline interpolators. In each case a tensor product decomposition is used to reduce the general n-dimensional interpolation problem to a sequence of n one-dimensional problems. Comparisons of a set of test problems indicate that the linear and taut spline interpolators are dispersive while the quadratic and cubic spline interpolators are oscillatory. The cubic and taut spline interpolators give consistently better accuracy than the more conventional linear and quadratic alternatives. Simulation experiments in two- and three-dimensional heterogeneous media indicate that the taut spline interpolator, which is applied here for the first time to a solute transport problem, is able to yield accurate essentially nonoscillatory solutions for high grid Peclet numbers. The cubic spline interpolator requires significantly less computational effort to achieve performance comparable to the other methods.

1. Introduction

High-resolution numerical simulations are being used with increasing frequency to investigate the behavior of solutes moving through heterogeneous porous media. In such media, small-scale velocity fluctuations contribute to larger-scale dispersion and mixing while also inducing small-scale fluctuations in solute concentration. Aspects of this process are still not well understood, especially at field sites where geologic variability cannot be adequately described with simple models. Numerical simulations that explicitly resolve small-scale velocity variations can provide useful information about macrodispersion mechanisms. In such simulations the Fickian term of the solute transport equation typically accounts only for pore-scale dispersion, while the advection term accounts for dispersion induced by velocity fluctuations at scales larger than the pore scale. Advection-dominated transport problems of this type are quite challenging and may require a different solution approach than more conventional groundwater transport problems, which make no attempt to resolve velocity fluctuations at such small scales [*Sudicky*, 1989]. Our focus in this paper is on a particular class of numerical solution techniques that are well suited for the advection-dominated simulations needed to test macrodispersion theories.

We start with a brief review of the most widely used options, which divide naturally into Lagrangian, Eulerian, and hybrid Eulerian-Lagrangian solution techniques. We then discuss in more detail some Eulerian-Lagrangian algorithms that are particularly well suited for investigating macrodispersion in heterogeneous porous media. We devote particular attention to the role of spatial interpolation, which has an important effect on the accuracy and efficiency of advection-dominated solutions.

Paper number 1999WR900049.
0043-1397/99/1999WR900049$09.00

The Lagrangian (or particle-tracking) approach for simulating solute transport represents a solute plume by a finite number of moving particles. The coordinates of each particle are described by a multidimensional Lagrangian transport (Langevin) equation [*Tompson and Gelhar*, 1990; *LaBolle et al*., 1996]. The average solute concentration in a given control volume is inferred by counting the number of particles contained within this volume at any given time. The accuracy of the method improves as the number of particles within each control volume increases while its resolution improves as the volume becomes smaller. The only way to meet these competing demands is to increase the total number of particles. Naturally, this increases the computational demands of the algorithm [*Yeung and Pope*, 1988; *Dimou*, 1992; *Rovelstad*, 1991, 1994; *Tompson and Gelhar*, 1990; *LaBolle et al*., 1996].

The Eulerian approach obtains a solution to the transport problem by integrating the advection-dispersion equation over a fixed computational grid. This approach can be computationally demanding for advection-dominated problems when stringent stability and accuracy constraints can necessitate the use of small grid cells and time steps [*Ames*, 1992; *Celia and Gray*, 1992]. Nevertheless, the Eulerian approach is widely used, probably because of the flexibility it offers in handling boundary conditions and source terms.

The Eulerian-Lagrangian approach is a hybrid method that shares features with both the Lagrangian and Eulerian approaches. Most Eulerian-Lagrangian algorithms divide the solution of the transport problem into two steps [*Cheng et al.*, 1984; *Neuman*, 1984; *Wheeler and Dawson*, 1988]. The first step uses an approach similar to particle tracking to solve the advective portion of the transport equation. The second step uses a Eulerian approach to solve the remaining dispersive portion. The algorithm repeatedly alternates between these two steps to obtain solute concentrations at discrete grid points and times. In some cases, hybrid Eulerian-Lagrangian methods can provide better accuracy and/or resolution than either Lagrangian or Eulerian solution algorithms of comparable cost.

Eulerian-Lagrangian methods have been widely investigated in the hydrologic and numerical analysis literature [*Holly and Preissmann*, 1977; *Holly and Jean-Marc*, 1984; *Ewing and Russell*, 1981; *Neuman*, 1981, 1984; *Douglas and Russell*, 1982; *Glass and Rodi*, 1982; *Cheng et al.*, 1984; *Baptista*, 1987; *Chiang et al.*, 1989; *Ho et al.*, 1990; *Celia and Gray*, 1992; *Bentley and Pinder*, 1992; *Yeh and Chang*, 1992; *Healy and Russell*, 1993, 1998; *Oliveira and Baptista*, 1994]. Several of these studies suggest that the accuracy of the spatial interpolation algorithm used during the advective step of the two-step solution procedure has a significant effect on the accuracy of the Eulerian-Lagrangian approach [*Dawson et al.*, 1989; *Xiang and Zhen*, 1992]. This is particularly true for advection-dominated problems.

In this paper we investigate four Eulerian-Lagrangian solution methods which are based on different spatial interpolation techniques. These are the (1) linear, (2) quadratic, (3) cubic spline, and (4) taut spline interpolators. The taut spline interpolator is especially convenient for transport problems with highly variable spatial gradients. We evaluate the performance of each interpolation alternative with a standard one-dimensional constant velocity test problem as well as with more challenging two- and three-dimensional variable velocity problems. In order to make the spline interpolators feasible for use in multidimensional problems we introduce a simplified, but mathematically equivalent, numerical implementation based on a tensor product decomposition. Our test problems suggest that spline interpolators can significantly improve solution accuracy by reducing numerical dispersion and spurious oscillations typically associated with linear and quadratic interpolators, respectively. This is particularly true of the taut spline interpolator, which practically eliminates the numerical oscillations typically associated with higher-order interpolators.

2. The Eulerian-Lagrangian Approach

Consider the following version of the solute transport equation written in nonconservative form and defined over a spatial domain Ω and time interval $J = (0, T]$:

$$\begin{gathered}\frac{\partial c}{\partial t} + \mathbf{v} \cdot \nabla c = \nabla \cdot (D \nabla c) + s \qquad (x, t) \in \Omega \times J \\ c(\mathbf{x}, 0) = c_0(\mathbf{x}) \qquad x \in \Omega \\ c = \bar{c}(\mathbf{x}, t) \qquad (x, t) \in \Omega_1 \times J\end{gathered} \tag{1}$$

where t is time, $c(\mathbf{x}, t)$ is solute concentration, $\mathbf{v}(\mathbf{x})$ is a spatially variable steady state velocity field, $s(\mathbf{x}, t)$ is a source-sink term, and D is a constant dispersion tensor [*Peaceman*, 1966]. We assume that the dispersion tensor is aligned with a constant mean velocity $\bar{\mathbf{v}}$:

$$D_{11} = \bar{\mathbf{v}}\alpha_L \qquad D_{22} = D_{33} = \bar{\mathbf{v}}\alpha_T \qquad D_{ij} = 0 \qquad i \neq j \tag{2}$$

where α_L and α_T are the longitudinal and transverse dispersivities, respectively.

The boundary of Ω, $\partial\Omega = \partial\Omega_1 + \partial\Omega_2$, is divided into two parts: (1) an inflow or no flow boundary $\partial\Omega_1$, where $\mathbf{v} \cdot \mathbf{n} \leq 0$ and (2) an outflow boundary $\partial\Omega_2$, where $\mathbf{v} \cdot \mathbf{n} > 0$. In both cases, $\mathbf{n}$ is an outward pointing unit normal vector on $\partial\Omega$. The locations of these boundaries can be identified prior to solution of (1) since $\mathbf{v}(\mathbf{x})$ is given. Most Eulerian-Lagrangian algorithms only enforce specified boundary conditions on inflow boundaries. As indicated below, this is possible because of the way the solution procedure is partitioned into advective and dispersive steps.

The transport equation of (1) can be written in Lagrangian form with the same initial and boundary conditions [*Cheng et al.*, 1984]:

$$\frac{Dc}{Dt} = \nabla \cdot (D \nabla c) + s \tag{3}$$

where the material derivative Dc/Dt represents the time rate of change calculated along a streamline (a particle path line). This derivative can be approximated by a backward finite difference evaluated along path lines:

$$\frac{Dc_i}{Dt} \simeq \frac{c[\mathbf{x}_i(t_{n+1}), t_{n+1}] - c[\mathbf{x}_i(t_n), t_n]}{\Delta t} \tag{4}$$

where $\mathbf{x}_i(t_{n+1})$ and $\mathbf{x}_i(t_n)$ are the Eulerian coordinates of a hypothetical particle traveling along path line i, evaluated at times t_{n+1} and t_n, respectively (see Figure 1), and $\Delta t = t_{n+1} - t_n$ is the time interval (assumed to be constant). On any given time step we call the coordinate $\mathbf{x}_i(t_n)$ the origin (or foot) of path line i.

The first (advective) step of the Eulerian-Lagrangian procedure derives the concentrations at the origins of the path lines ending at the nodes of a fixed Eulerian computational grid. The origin of path line i can be found by backtracking along the path line from time t_{n+1} to time t_n using the following kinematic equation [*Neuman*, 1981; *Chiang et al.*, 1989; *Wheeler and Dawson*, 1988]:

$$\mathbf{x}_i(t_n) = \mathbf{x}_i(t_{n+1}) - \int_{t_n}^{t_{n+1}} \mathbf{v}[\mathbf{x}_i(\xi)]\, d\xi \qquad i = 1, \cdots, N \tag{5}$$

where $\mathbf{x}_i(t_{n+1})$ is the vector location of node i. This is the widely used modified method of characteristics (MMOC) approach. Since (5) is an implicit equation for $\mathbf{x}_i(t)$ it must generally be solved numerically. The time interval $[t_n, t_{n+1}]$ may be divided into M fractional time steps of length $\Delta t' = \Delta t/M$ (see Figure 1), and the integral over each fractional time step may be computed with an appropriate integration algorithm. In our examples we use a fourth-order Runge-Kutta technique [*Press et al.*, 1986]. The selection of the number of fractional time steps depends on both Δt and the magnitude of the local velocity vector. It should be large enough to provide a few intermediate integration points along the portion of the path line crossing a single Eulerian grid cell. If this fractional time step criterion is followed and the velocity field is time-invariant, the accuracy of the advective step solution does not depend on Δt.

The discrete numerical algorithm used to integrate (5) requires velocity values at various points along the path line between $\mathbf{x}_i(t_{n+1})$ and $\mathbf{x}_i(t_n)$. Since these intermediate points typically do not coincide with predefined nodal positions the

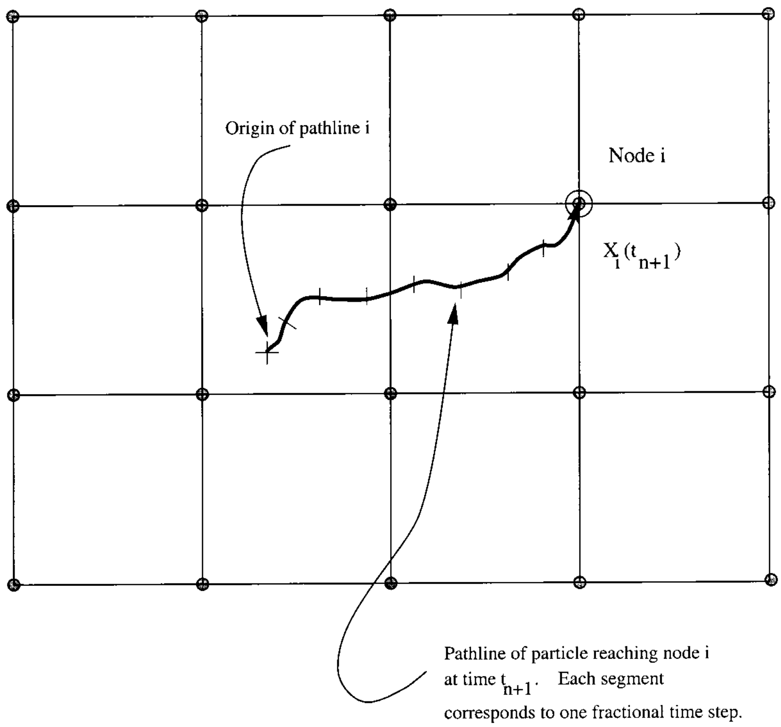
Origin of pathline i
Node i
$X_i(t_{n+1})$
Pathline of particle reaching node i at time t_{n+1}. Each segment corresponds to one fractional time step.

Figure 1. Interpolation in the Eulerian-Lagrangian solution procedure.

required velocities must be interpolated from nearby nodal values. Accurate velocity interpolation can be difficult in locations where velocity gradients are high. However, interpolation need only be done once when the velocity field is time-invariant (as we assume here). When the origin of path line i has been found from (5), the concentration $c[\mathbf{x}_i(t_n), t_n]$ may be obtained by interpolating known concentrations at nearby grid nodes. This concentration is saved for use in the second step of the Eulerian-Lagrangian procedure.

The second (dispersive) step of the solution procedure derives the unknown nodal concentrations at time t_{n+1} from the following equation, which is obtained by substituting (4) into (3):

$$c[\mathbf{x}_i(t_{n+1}), t_{n+1}] - \Delta t[\nabla \cdot (D\nabla c) + s]_{\mathbf{x}_i(t_{n+1}), t_{n+1}} = c[\mathbf{x}_i(t_n), t_n] \quad (6)$$

where $c[\mathbf{x}_i(t_n), t_n]$, the known concentration at the origin of path line i, is obtained from the first step of the Eulerian-Lagrangian procedure and the derivatives in the bracketed term are evaluated at $\mathbf{x}_i(t_{n+1})$. We solve this spatial differential equation with an implicit centered-in-space finite difference procedure that provides values of $c[\mathbf{x}_i(t_{n+1}), t_{n+1}]$ at the N nodes of the computational grid at time t_{n+1}. Although finite element or higher-order finite difference methods could be used to discretize (6), there is no real need for this in the advection-dominated problems of interest here since the errors arising from spatial discretization of this expression are of secondary importance when compared to the spatial interpolation errors arising in the advective step (see error discussion below). The solution of the finite difference system can be used to initialize the next computational step from t_{n+1} to t_{n+2}.

Boundary conditions in an Eulerian-Lagrangian solution procedure are applied in each of the two computational steps. In the advective step it is sufficient to specify the concentration on inflow boundaries since outflow boundaries should not be intersected by path lines derived from a backtracking algorithm. When a path line intersects an inflow boundary during the backtrack from t_{n+1} to t_n, it is convenient to choose $\mathbf{x}_i(t_n)$ to be the intersection point and to then set $c[\mathbf{x}_i(t_n), t_n]$ equal to $\bar{c}[\mathbf{x}_i(t_n), t_n]$, the boundary concentration specified on $\partial\Omega_1$. In such a case, Δt in (6) should be replaced by $\Delta t'' = t_{n+1} - t'$, where t' is the intersection time [*Bentley and Pinder*, 1992]. In the dispersive step, boundary conditions are required everywhere on $\partial\Omega$ in order to obtain a unique solution to (6). At inflow boundary nodes the dispersive step concentration is set equal to $\bar{c}[\mathbf{x}_i(t_n), t_n]$. At outflow boundary nodes the most convenient approach for advection-dominated problems is to set the dispersive step concentration equal to the concentration value obtained from the preceding advective step. This approach allows the outflow boundary to remain "open" while providing the constraints needed to solve (6). It works well if the outflow boundaries are placed far enough from pumping wells and other localized sources and sinks.

There are several possible sources of error in the Eulerian-Lagrangian solution procedure outlined above. First, there are errors associated with velocity interpolation and backtracking during the advective (Lagrangian) step. If the velocity field is time-invariant, these errors decrease as the spatial grid is refined and the size of the fractional time step used in backtracking is reduced. Second, there are errors associated with concentration interpolation and forward propagation along the particle path lines. These errors generally decrease as the spatial grid is refined. They have the distinctive feature of accumulating from step to step so their aggregate effect may grow as the time step is decreased. Velocity interpolation errors can also accumulate in this way if the velocity field is time-dependent. Third, there are errors resulting from the splitting of the advective and dispersive steps over discrete time intervals. These errors generally decrease as the time step is reduced since the splitting operation introduces no error in the limit as the time step goes to zero. Fourth, there are errors due to the spatial discretization used in the dispersive (Eulerian) computational step. These errors depend on the spatial discretization technique and grid resolution used in the dispersive step.

The relative magnitudes of these various error sources depend on many factors in addition to the time step and spatial grid resolution. These factors include the source size, the degree of heterogeneity in the velocity field, the travel distance, and the values of the local dispersivities. Our experience indicates that for advection-dominated problems, errors due to temporal discretization of the splitting operation (type 3) and errors due to spatial discretization of the dispersive step (type 4) are typically small relative to the other errors mentioned above (types 1 and 2) provided that the time step and spatial resolution are sufficiently small. If the time step is made too large (for example, in an effort to reduce the cumulative effects of type 2 concentration interpolation errors), type 3 splitting errors can eventually dominate [*Valocchi and Malmstead*, 1992].

Our approach in this paper is to use a "moderate" time step that is large enough to keep computational demands and concentration interpolation errors in check but small enough to make splitting and temporal discretization errors negligible. This generally optimizes the algorithm's accuracy. When the time step is at or near the optimal value, spatial interpolation is the most important factor governing the accuracy of the Eulerian-Lagrangian approach, at least for the advection-dominated problems of interest here. Unfortunately, we have no general guideline for determining a priori just what this "optimal" time step should be in any given application. Our approach has been to identify the optimal time step by carrying out sensitivity analyses with respect to both time and space steps. Further details are presented by *Ruan* [1997].

3. Spatial Interpolation Alternatives

The discussion presented above suggests that if the velocity field is time-invariant and the time step is of moderate size, spatial discretization/interpolation is the primary source of error in advection-dominated Eulerian-Lagrangian solutions. We shall see in our examples that the errors associated with a given level of spatial resolution depend significantly on the method used to interpolate velocity and concentration in the advective step of the solution procedure. Consequently, it is useful to consider in more detail how the four spatial interpolation alternatives examined in this paper are actually implemented.

Among these alternatives the linear and quadratic interpolators have been the most widely used in multidimensional transport applications [*Baptista et al*., 1984; *Cheng et al*., 1984; *Chiang et al*., 1989; *Goode*, 1990]. Higher-order interpolators have generally been regarded as undesirable either because they are difficult to implement or they require too much computational effort [*Lapidus and Pinder*, 1982]. In this section we formulate a simple and efficient approach to multidimensional polynomial interpolation that relies on a tensor product decomposition. This decomposition makes cubic spline interpolation an attractive option for solute transport problems.

3.1. Global One-Dimensional Interpolation

It is easiest to introduce the relevant notation if we start with the one-dimensional case. The interpolators of interest here approximate a one-dimensional function $z(x)$ (e.g., velocity or concentration) over the interval $[x_1, x_L]$ from a set of discrete values $z_1 = z(x_1)$, $z_2 = z(x_2)$, $\ldots$, $z_L = z(x_L)$ defined at locations $x_1 < x_2 < \cdots < x_L$ that are not necessarily equally spaced. The approximating function $\hat{z}(x)$ can be constructed from a specified set of N linearly independent scalar interpolation polynomials (or basis functions) $\phi_j(x)$ as follows:

$$\hat{z}(x) = \sum_{j=1}^{N} a_j \phi_j(x) \tag{7}$$

where the a_j are unknown coefficients that depend on the data $z_1, \cdots, z_L$. The number of basis functions N depends on L and the particular interpolation method selected. This global expression for the general interpolator is convenient for our analysis of the tensor product decomposition.

The four interpolators considered here differ both in the form used for the basis functions in (7) and in the region over which these functions are defined (their support). Each of the linear and quadratic basis functions is a simple polynomial defined over the interval $[x_l, x_{l+1}]$ between two adjacent data points (two polynomials per interval for the linear case and three for the quadratic). Each cubic spline basis function is constructed from a fourth-order B spline, which is defined over the interval spanned by five data points. The total number of basis functions (N) is equal to $2(L - 1)$, $3(L - 1)$, and L for the linear, quadratic, and cubic spline, respectively. Detailed discussions are provided by *de Boor* [1978, 1992] and *Farin* [1993].

The unknown basis function coefficients appearing in (7) may be identified by constraining the approximating function at each data location to be equal to the corresponding data value. The number of constraints required at each internal data location is two, three, and one for the linear, quadratic, and cubic spline interpolators, respectively. When end effects are accounted for, the total number of constraints generated in each case is equal to N, the total number of unknown coefficients. These constraints can be written in a concise form if we define an N vector u_i containing the locations of all constraints. Then the approximating function at each data location will be equal to the corresponding data value if

$$\hat{z}(u_l) = \phi_i(u_l)a_i = \Phi_{il}a_i = z(u_l) \tag{8}$$

where $\Phi_{il} = \phi_i(u_l)$ is an N by N matrix of the basis functions evaluated at the constraint locations. Indicial notation is used (i.e., repeated indices are understood to be summed) from now on whenever the range of summation is apparent from context. Since the basis functions are linearly independent the set of coefficients that satisfies (8) is

$$a_i = [\Phi^{-1}]_{il} z(u_l) \tag{9}$$

where Φ^{-1} is the inverse of Φ. The corresponding global interpolation equation can be obtained by substituting (9) into (7) and adopting indicial notation:

$$\hat{z}(x) = \phi_i(x)[\Phi^{-1}]_{il} z(u_l) \tag{10}$$

It is apparent from this expression that the linear, quadratic, and cubic spline interpolators yield approximations to $z(x)$ that depend linearly on the data in the vector $z(u_l)$. In practice, the procedures used to solve the one-dimensional interpolation problem take advantage of the block diagonal (for the linear and quadratic cases) or banded (for the cubic spline case) structure of the Φ matrix.

3.2. Global Multidimensional Interpolation

The concepts outlined above for one-dimensional interpolation extend naturally to multiple dimensions. For example, the two-dimensional counterpart to (7) is

$$\hat{z}(x, y) = \sum_{i=1}^{N_x} \sum_{j=1}^{N_y} a_{ij}\psi_{ij}(x, y) \tag{11}$$

where $\psi_{ij}(x, y)$ is a specified two-dimensional basis function, a_{ij} is the corresponding unknown interpolation coefficient, and N_x and N_y are the number of coefficients associated with the x and y directions, respectively. We presume here that the data $z(x_i, y_j) = z_{ij}$ to be interpolated in the two-dimensional procedure are located on a rectangular grid of points with coordinates x_i, $i = 1, \cdots, L_x$ and y_j, $j = 1, \cdots, L_y$. Although this is a specialized spatial configuration, it is sufficiently general to apply to the finite difference computational grids used in our transport simulations.

For the interpolators considered here the two-dimensional basis functions may be written as products of the one-dimensional basis functions defined above, evaluated at x and y. In this case the two-dimensional interpolation equation has the following "tensor product" form [*de Boor*, 1978; *Wait and Mitchell*, 1985]:

$$\hat{z}(x, y) = \sum_{i=1}^{N_x} \sum_{j=1}^{N_y} a_{ij}\phi_i(x)\phi_j(y) \tag{12}$$

As in the one-dimensional case, the unknown basis function coefficients may be identified by constraining the approximating function at each data location to be equal to the corresponding data value. The numbers of constraints N_x or N_y and the constraint coordinate vectors u_{xm} or u_{yn} in each direction are the same as for the comparable one-dimensional interpolators with either L_x or L_y data points.

The multidimensional counterpart to (10) can be expressed in indicial notation as [*Ruan*, 1997]

$$\hat{z}(x, y) = \phi_i(x)[\Phi_x^{-1}]_{im}z(u_{xm}, u_{yn})[\Phi_y^{-1}]_{jn}\phi_j(y) \tag{13}$$

In practice, this expression can be evaluated by carrying out two one-dimensional interpolations. To investigate this further, suppose that we desire an interpolated estimate of $z(x, y)$ at the point (x, y). The first one-dimensional problem is to construct L_y univariate interpolators, each along the x direction, and then to use each of these interpolators to obtain $\hat{z}$ values at (x, y_1), (x, y_2), $\cdots$, (x, y_{L_y}). The second one-dimensional problem is then to estimate $z(x, y)$ by interpolating along the y direction the L_y values obtained from the first problem. *Ruan* [1997] shows that the estimate $\hat{z}(x, y)$ obtained from these two sequential one-dimensional interpolations is identical to the direct two-dimensional estimate given in (13). The same result holds if the order of the one-dimensional interpolations is reversed (i.e., interpolation along y followed by interpolation along x) [*de Boor*, 1992; *Press et al.*, 1986]. Finally, the equivalence of the sequential and direct approaches extends to three-dimensional problems, where three one-dimensional interpolations are required (e.g., first along x, next along y, and last along z).

3.3. Local Multidimensional Interpolation

We have seen above that the n-dimensional interpolation problem can be expressed as a sequence of n one-dimensional problems. This is particularly attractive for cubic spline algorithms since each one-dimensional subproblem can be solved with a local form of the one-dimensional cubic spline function, which is easier to use than the global basis function expansion of (10). The global forms for the one-dimensional interpolators considered here are computationally demanding and can lead to poorly conditioned Φ matrices. The local forms are mathematically equivalent to the global forms but are more efficient and better conditioned. For this reason we use the global interpolators above to present and derive the multidimensional tensor product decomposition but rely on local one-dimensional interpolators with a tensor product decomposition for actual computations.

The local linear and quadratic interpolators are the well-known Lagrangian interpolation functions [*Press et al.*, 1986]. The local spline interpolator can be written in terms of the normalized coordinate α as follows:

$$\hat{z}(x) = \begin{cases} \alpha z_l + (1-\alpha)z_{l+1} + Cz''_l + Dz''_{l+1} & x \in (x_l, x_{l+1}] \\ 0 & \text{otherwise} \end{cases} \tag{14}$$

where

$$\alpha = \frac{x_{l+1} - x}{x_{l+1} - x_l}$$

$$C = \tfrac{1}{6}(x_{l+1} - x_l)^2(\alpha^3 - \alpha)$$

$$D = \tfrac{1}{6}(x_{l+1} - x_l)^2[(1-\alpha)^3 - (1-\alpha)]$$

The unknown coefficients z''_l and z''_{l+1} are the second derivatives of $\hat{z}(x)$ evaluated at x_l and x_{l+1}, respectively. They are determined by requiring the first derivatives of $\hat{z}(x)$ to be continuous at each x_l for $l = 2, \cdots, L-1$. This constraint yields a tridiagonal system of L equations in the L unknown z''_ls for each one-dimensional interpolation subproblem. Consequently, the total number of floating point operations required to perform a complete multidimensional interpolation with the local cubic spline formulation is of order L. *De Boor* [1992] demonstrates the equivalence of the global and local versions of the cubic spline interpolator, and *Press et al.* [1986] discusses implementation details.

The local version of the cubic spline provides a convenient way to introduce the taut spline interpolator, which is a variant of (14). The basic idea is to add tension (or stretch) to the local cubic polynomial to eliminate inflection points. This typically reduces spurious oscillations [*de Boor*, 1978]. The local form of the taut spline may be written in a form similar to (14):

$$\hat{z}(x) = \begin{cases} \alpha z_l + (1-\alpha)z_{l+1} + C'(\Psi)z''_l + D'(\Psi)z''_{l+1} & x \in (x_l, x_{l+1}] \\ 0 & \text{otherwise} \end{cases} \tag{15}$$

where $C'(\Psi)$ and $D'(\Psi)$ are modified spline coefficients defined in the appendix. The definition of the "tautness parameter" Ψ for data interval l depends on the existence of an inflection point in this interval. When such a point exists, the standard cubic spline polynomial is "stretched" to provide a more nearly linear interpolation between z_l and z_{l+1}. This has the effect of lowering the order of the interpolation polynomial in regions where concentration gradients are high. Since the tautness parameter is computed in every data interval at every time step the taut spline continually adapts to changing conditions. Furthermore, the one-dimensional taut spline may be incorporated into a decomposed multidimensional interpolation in the same way as the standard cubic spline function. Appendix A describes the taut spline algorithm in more detail. The simplifications introduced by decomposition make the cubic and taut spline interpolators computationally competitive alternatives to more traditional linear and quadratic interpolators.

The various interpolators discussed in this section behave differently when used in Eulerian-Lagrangian transport solvers. Generally speaking, interpolation accuracy increases as the order of the interpolator increases. However, care must be taken in assessing accuracy since some interpolators (e.g., linear and taut spline) tend to damp or smooth the true solution while others (e.g., quadratic and cubic spline) tend to introduce spurious oscillations. In order to appreciate better the strengths and weaknesses of each alternative it is helpful to consider some examples.

4. Examples

4.1. One-Dimensional Example

In this section we compare the performance of the linear, quadratic, cubic spline, and cubic taut spline interpolators for a well-known one-dimensional benchmark problem. This problem propagates a single sharp concentration front from a continuous source of concentration $\bar{c}$ through a domain of length L at a constant pore velocity v. The velocity is time-invariant, and the space and time steps are fixed. The time step is sufficiently small to have a negligible effect on solution accuracy (the Courant number in our one-dimensional tests is always <0.5). An exact solution is available in closed form [*Ogata*, 1961].

It is convenient for sensitivity analysis purposes to relate the coefficients of the transport equation to a few dimensionless parameters. In particular, the velocity and dispersion coefficients for the one-dimensional case can be written

$$v = \frac{Cr\Delta x}{\Delta t} = \frac{CrLN_t}{N_xT} \tag{16}$$

$$D = \frac{v\Delta x}{Pe} = \frac{CrN_tL^2}{PeTN_x^2} \tag{17}$$

where $Cr = (v\Delta t)/\Delta x$ is the Courant number, $Pe = (v\Delta x)/D$ is the grid Peclet number, T is the duration of the simulation, N_t is the number of equal time steps in $[0, T]$, and N_x is the number of grid (or data) intervals in $[0, L]$. For our example we set $L = 1$ and $T = 1$ and examine the sensitivity of the results to Pe, with all other dimensionless inputs held constant, for two different values of N_x. Since the velocity is constant in this one-dimensional example, only concentration interpolation error is an issue.

Figure 2 summarizes the simulation results at time 0.45 for various grid Peclet numbers. In all cases the number of time steps N_t is 500. Figures 2a, 2b, and 2c show normalized concentration $c' = c/\bar{c}$ versus dimensionless distance for a fine grid with $N_x = 550$ and $Cr = 0.5$, giving a ratio $Cr/N_x = 0.00091$. Figures 2d, 2e, and 2f are obtained from a coarser grid with $N_x = 100$, $Cr = 0.091$, and the same ratio $Cr/N_x = 0.00091$. Note that the dispersion coefficients for the fine and coarse grids for any given grid Peclet number are different (this is why the front of the fine-grid analytic solution is slightly sharper than the coarse-grid front). Generally speaking, numerical transport problems are more difficult to solve when the Courant and Peclet numbers are high and easier to solve when these numbers are small. In this example we concentrate on high grid Peclet number performance since we are especially interested in problems where advection dominates.

As the grid Peclet number increases (from top to bottom) or the grid resolution decreases (from left to right) the linear interpolator solution becomes more damped, and the quadratic interpolator solution becomes more oscillatory. The cubic spline solution displays some parasitic wiggles, and the cubic taut spline solution exhibits minor numerical dispersion with the highest Peclet number and coarsest grid (Figure 2f). However, it is apparent that the two spline interpolators can achieve more accurate solutions than the linear and quadratic interpolators, especially on a coarse grid. Such a feature is important in multidimensional applications where the numerical grid can have millions of nodes.

The cubic and the cubic taut spline interpolators have different error properties as simulation time increases. This is shown in Figure 3, which provides four snapshots of concentration fronts from a simulation that uses the same parameters as the one shown in Figure 2f. The degree of oscillation in the cubic spline solution stays approximately the same, but the degree of numerical dispersion in the cubic taut spline solution accumulates over time. Overall, the two spline interpolators are able to provide reasonably accurate solutions to this one-dimensional problem for grid Peclet numbers as high as 500.

4.2. Two-Dimensional Example

In this section we consider two-dimensional solute transport through a synthetically generated heterogeneous steady state velocity field in a rectangular domain. Such simulations have been used to investigate large-scale solute dispersion and to test stochastic macrodispersion theories [see, e.g., *Graham and McLaughlin*, 1989]. The time-invariant velocity used in our example is obtained from the random field generator described by *Ruan and McLaughlin* [1998]. This generator relies on approximate velocity spectral and cross-spectral densities that are derived from a linearized analysis of the groundwater flow equation and Darcy's law [*Gelhar*, 1993]. The longitudinal and transverse components of the generated velocity field are spatially cross correlated by virtue of their mutual dependence on the log hydraulic conductivity. The mean velocity is aligned with the longitudinal direction, and the log hydraulic conductivity is assumed to have a Gaussian spectral density. The velocity field conserves mass and is practically divergence-free at each grid node.

The velocity and dispersion coefficients for the two-dimensional case can be related to dimensionless parameters in the following way:

$$\begin{aligned} v_x &= v_x'\bar{v} = v_x'\frac{CrL_xN_t}{N_xT} \\ v_y &= v_y'\bar{v} = v_y'\frac{CrL_yN_t}{N_yT} \\ D_x &= \frac{\bar{v}\Delta x}{Pe_x} = \frac{CrN_tL_x^2}{Pe_xTN_x^2} \\ D_y &= \frac{\bar{v}\Delta y}{Pe_y} = \frac{CrN_tL_xL_y}{Pe_yTN_xN_y} \end{aligned} \tag{18}$$

where v_x' and v_y' are dimensionless random velocities in the longitudinal x and transverse y directions, $\bar{v}$ is the mean ve-

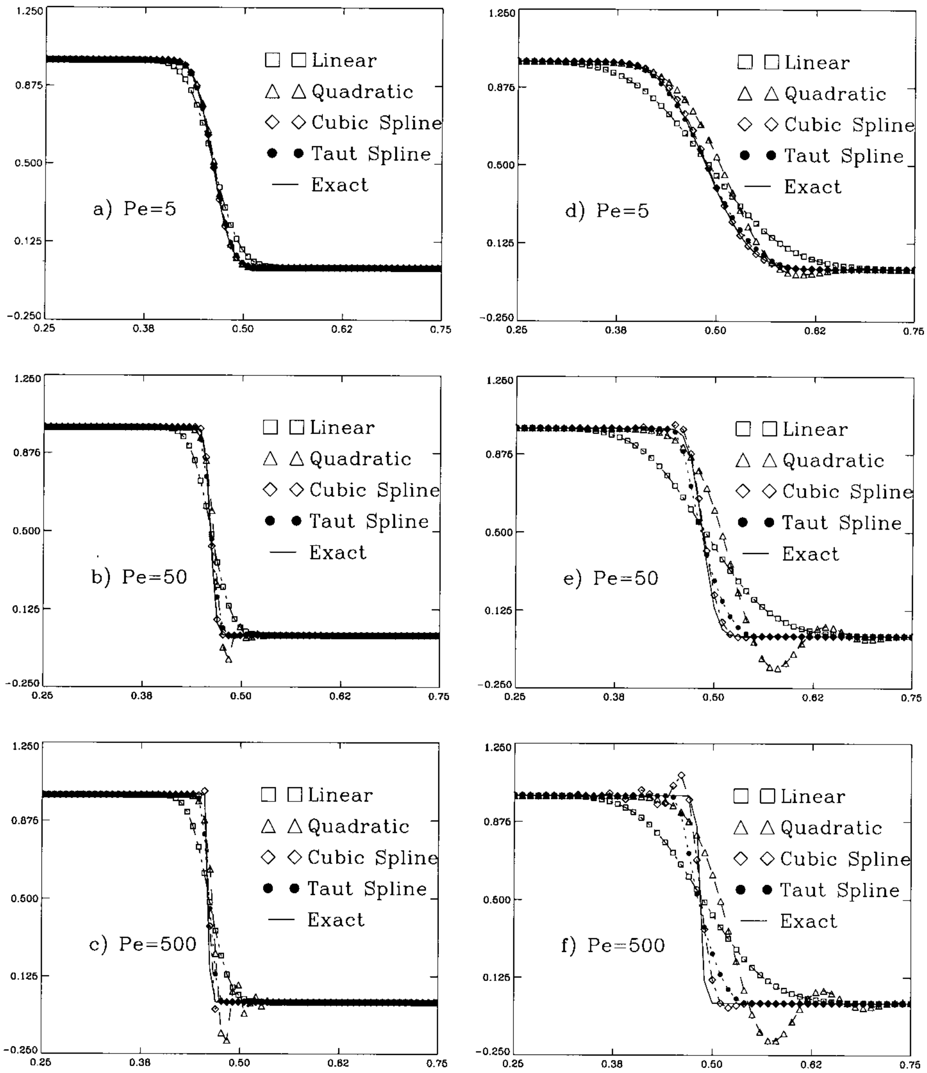

Figure 2. Comparison of Eulerian-Lagrangian normalized concentration solutions versus normalized distance for various Peclet numbers. Results in Figures 2a, 2b, and 2c are computed on a fine grid ($N_x = 550$), while results in Figures 2d, 2e, and 2f are computed on a coarse grid ($N_x = 101$).

locity (aligned with the x coordinate), $Cr = (\bar{v}\Delta t)/\Delta x$ is the Courant number associated with the mean velocity, $Pe_x = (\bar{v}\Delta x)/D_x$ and $Pe_y = (\bar{v}\Delta y)/D_y$ are the longitudinal x and transverse y grid Peclet numbers, and N_x and N_y are the numbers of grid (or data) intervals in the x and y directions. For our nominal simulation we set $L_x = 2$, $L_y = 1$, $N_x = 120$, $N_y = 60$, $T = 1$, and $N_t = 160$. The Courant number is 0.42, and both grid Peclet numbers are 40 in all two-dimensional simulations.

The instantaneous source for our two-dimensional example follows a Gaussian spatial distribution and is applied at time $t = 0$. The source function is centered at location (0.4, 0.5) and has spatial standard deviations of 0.025 and 0.10 in the longitudinal and transverse directions, respectively. The dimensionless velocities v'_x and v'_y are characterized by their mean, which is $\bar{\mathbf{v}} = (\bar{v}_x, \bar{v}_y) = (\bar{v}, 0)$ by construction, the log hydraulic conductivity variance $\sigma_f^2 = 1.0$, and the dimensionless log conductivity correlation scale $\lambda/L = 0.067$. The simulation inputs are summarized in Table 1. Without any loss of generality we adopt constant time and space steps in this example.

Figure 4 shows typical streamlines derived from the dimensionless velocity field used for our two-dimensional example. The colored background indicates the magnitude of the velocity vector. In our two-dimensional example we again consider the four interpolators investigated in the one-dimensional case. The same interpolation method is used for path line tracking and concentration interpolation. Plate 1 shows concentration contours obtained at time 0.75. The linear interpolator solution in Plate 1a is very smooth with a relatively low peak. The quadratic interpolator solution in Plate 1b exhibits more variability in regions of high concentration. It also produces large patches of negative concentrations at the upper and front portions of the plume. The taut spline (Plate 1c) and cubic spline (Plate 1d) solutions are as variable as the quadratic solution but are much less prone to give negative concentrations.

Our two-dimensional example confirms the qualitative conclusions drawn from the one-dimensional example, indicating that the linear interpolator is dispersive and the quadratic interpolator is oscillatory. The slightly oscillatory cubic spline solution produces some negative concentrations in the plume

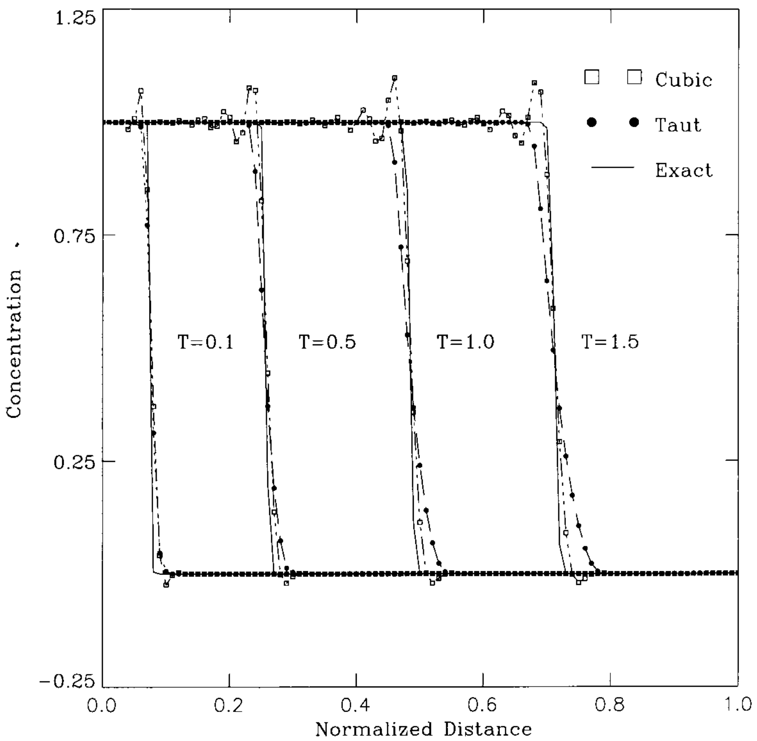

Figure 3. Comparison of cubic and taut spline solutions at various times, where $Pe = 500$.

tails where there are multiple local peaks of low intensity. Negative concentrations are almost absent in the slightly dispersive taut spline solution.

In our one-dimensional example we were able to obtain a quantitative evaluation of solution accuracy by comparing numerical and analytical solutions. Such comparisons are not possible in multidimensional heterogeneous velocity problems where closed form analytical solutions are unavailable. In our two-dimensional example we must use less direct measures, such as the convergence and mass balance properties of the solution, to assess accuracy. These are discussed in the following sections.

4.2.1. Sensitivity to grid resolution. A numerical algorithm for solving the solute transport equation should converge to a stable result (ideally the unknown exact solution) as its time step and spatial grid are refined. We can say that convergence is achieved when a further increase in spatial and temporal resolution does not make a significant difference in the solution, as measured by an appropriate norm. Here we measure convergence relative to a "reference" cubic spline solution obtained with a refined node spacing equal to $0.2\Delta x$ in both directions, where Δx is the nominal (coarsest) x direction spacing given in Table 1. The time step for the reference solution is 0.00625, corresponding to a Courant number of 2.1.

We compare the reference solution to a set of solutions computed on grids coarser than the reference grid, with node spacings of decreasing size defined by

$$\Delta x_i = \Delta x\left(1 - \frac{i}{8}\right) \qquad i = 0, \cdots, 6 \qquad (19)$$

where the index i refers to a particular grid. The spatially discretized random velocity for the reference simulation is generated on the coarsest (Δx) grid and then interpolated onto each of the six finer grids with a third-order spline. Our simulation results are relatively insensitive to discretization of the velocity field since even the coarsest-grid spacing is less than one third of the continuous velocity correlation scale. Consequently, all grids see essentially the same velocity field. Their solutions differ primarily because of differences in errors introduced during the advective step of the Eulerian-Lagrangian procedure.

For each spatial resolution considered ($i = 0, \cdots, 6$) we project the reference simulation velocity and concentration fields onto grid i. We then evaluate the root-mean-square difference between the reference solution and grid i solutions as follows:

Table 1. Input Parameters for the Nominal Two-Dimensional Simulation

Parameters	Values
Log k variance σ_f^2	1.0
Porosity ρ	0.3
Mean hydraulic gradient $\|J\|$	0.002
Domain length $L_x = 2L$	2
Domain width $L_y = L$	1
Grid size $N_x \times N_y$	121 × 61
Normalized grid spacing $\Delta x/L = \Delta y/L$	0.0167
Total simulation time T	1
Total time steps N_t	160
Normalized hydraulic conductivity geometric mean K_gT/L	168
Normalized log hydraulic condition correlation length λ/L	0.067
Grid Peclet number $Pe_x = Pe_y$	40
Longitudinal Courant number Cr	0.42
Normalized standard deviation of initial source $\sigma_x/L \times \sigma_y/L$	0.025 × 0.10
Normalized initial source location $(x_0/L, y_0/L)$	(0.4, 0.5)

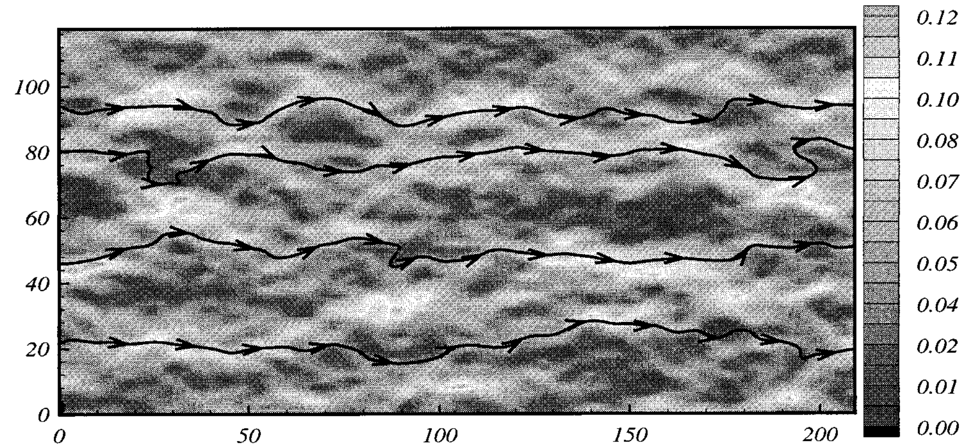

Figure 4. Random velocity field used in the transport simulations. The axis units refer to node numbers. The color scale indicates the magnitude of local two-dimensional velocity. Note that the mean velocity gradient in the x direction induces anisotropy in the velocity field.

$$J_{Gi} = \frac{1}{N_i M_i(t)} \left[\sum_{k=1}^{N_i} [c_i^*(x_k, t) - \hat{c}_i(x_k, t)]^2 \right]^{1/2} \qquad i = 0, \cdots, 6 \tag{20}$$

where c_i^* is the reference concentration solution projected onto grid i, $\hat{c}_i$ is the concentration solution computed on grid i, $M_i(t)$ is the computed solute mass for grid i, and N_i is the total number of nodal points at resolution i. If the time step is sufficiently small, as it is in the examples considered here, the spatial resolution of the computational grid is the primary factor affecting the value of the convergence norm. Note that the Courant number decreases as the node spacing increases since the time step and mean velocity remain constant.

Figure 5 plots the log of J_{Gi} versus the log of Δx_i at time $t = 1.0$. As expected, the error measure decreases toward zero as

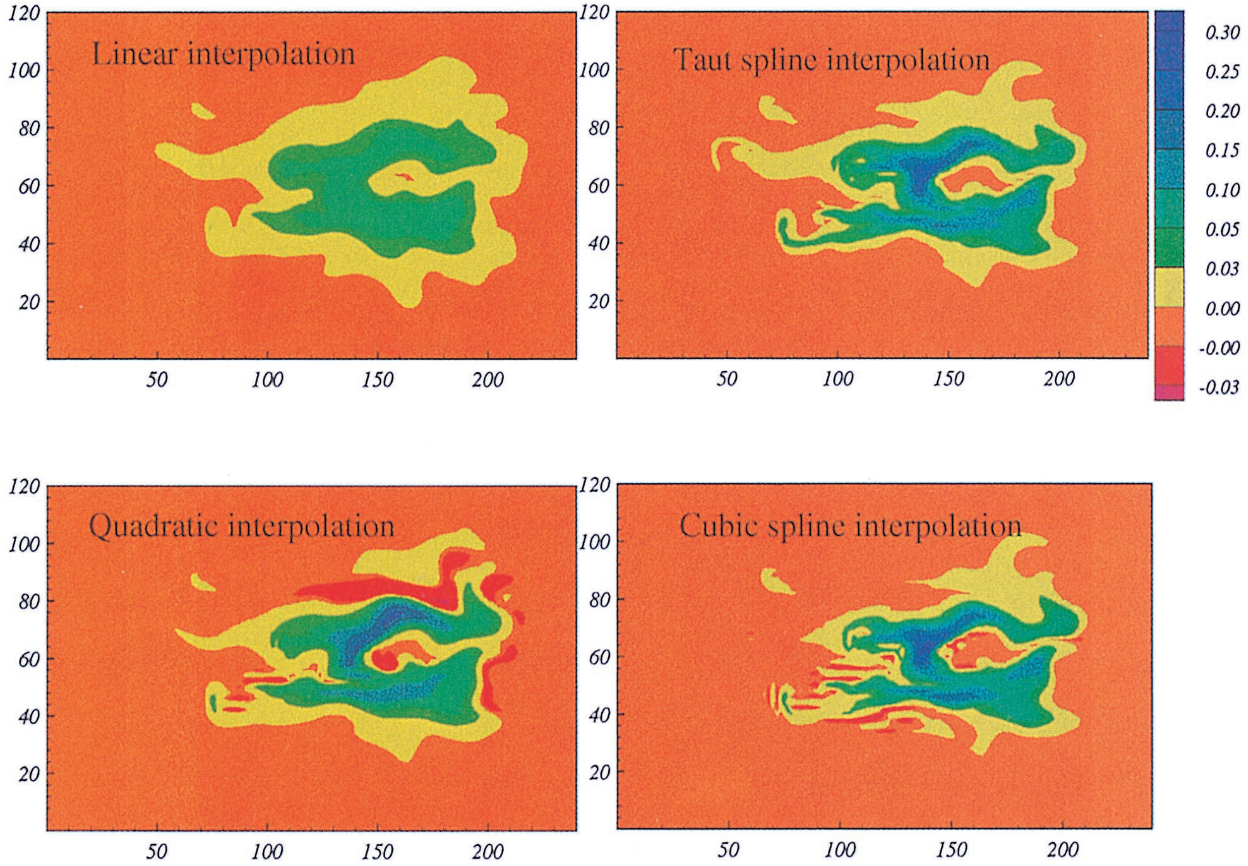

Plate 1. Simulated two-dimensional plumes in a spatially variable velocity field. Axis units refer to node numbers.

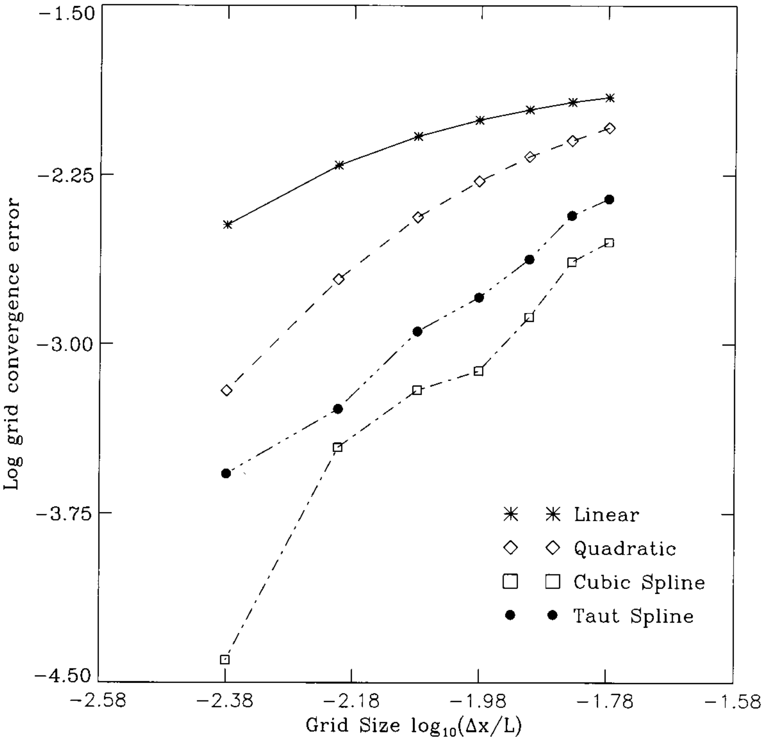

Figure 5. The $\log_{10}$ of the mean square convergence norm (J_{Gi}) plotted versus the $\log_{10}$ of the x dimensionless grid node spacing $\Delta x_i/L$.

Δx_i decreases, indicating convergence to the fine-grid reference solution. The magnitude of the error measure and the rate of convergence of this measure differ significantly for the four interpolators, with the cubic and taut spline giving consistently better results at all resolutions studied. The least squares lines fit to the error norm curves give rough estimates for the order of accuracy of each method (linear = 0.92, quadratic = 1.91, taut spline = 2.1, and cubic spline = 2.85). These convergence slopes correspond well to the interpolator order, with the exception of the taut spline, which converges at nearly the same rate as the quadratic. However, it should be noted that the magnitude of the taut spline error measure is significantly smaller than the magnitude of the quadratic error measure at all resolutions. In any case it is apparent that convergence errors decrease more rapidly for the higher-order interpolators as compared to the more frequently used linear interpolator. The obvious question is whether or not this performance improvement is obtained at the expense of increased computation.

Table 2 compares the computational times the four interpolators require to achieve approximately the same degree of convergence (i.e., approximately the same value of J_{Gi}). Computation times are tabulated for low, medium, and high values of J_{Gi}. These times apply for a Digital Equipment Corporation (DEC) Alpha workstation with a CPU speed of 333 Mhz and a core memory of 128 Mb. Note that the range of grid sizes considered did not produce all possible outcomes (e.g., even the finest grid was still too coarse to yield medium or low J_{Gi} values with the linear interpolator). However, the results are sufficient to demonstrate that the cubic spline is much more efficient than the other techniques. In particular, the cubic spline is able to achieve convergence performance comparable to the other alternatives with a much coarser grid. The efficiency of the spline interpolator is a direct result of the tensor

Table 2. Comparison of Computational Times

	Method			
$\text{Log}_{10} J_G$	Linear	Quadratic	Cubic Spline	Taut Spline
High J_G (−2.42 ± 0.02)				
CPU time, s	121.33	50.36		70.66
Grid	481 × 241	241 × 121		138 × 69
Medium J_G (−2.65 ± 0.05)				
CPU time, s		95.57	10.40	98.94
Grid		321 × 161	138 × 69	161 × 81
Low J_G (−3.15 ± 0.01)				
CPU time, s		242.7	32.99	
Grid		481 × 241	193 × 97	

product decomposition, which makes it possible to divide the multidimensional interpolation problem into a sequence of simple one-dimensional problems that can each be solved with a local interpolation algorithm. The linear interpolator is the least attractive alternative since it requires both a finer grid and more computation time to achieve the same degree of convergence as the quadratic and taut spline interpolators.

It should be noted that J_{Gi} is an aggregate measure of algorithm performance and cannot be expected to adequately capture all factors that may be of interest in evaluating a particular interpolation method. For example, although Table 2 would suggest that the taut spline is not superior to the quadratic interpolator in terms of J_{Gi}, it is much less oscillatory and may therefore be preferred in certain applications (e.g., where it is important to avoid solutions with negative concentrations). A decision to adopt any particular interpolator should consider a number of different performance criteria relevant to the problem of interest.

4.2.2. Mass balance. It is well known that the Eulerian-Lagrangian transport solvers considered above do not generally conserve mass [*Baptista et al.*, 1984]. We can evaluate the mass balance performance of our Eulerian-Lagrangian interpolators by comparing the total simulated solute mass at various times to the total solute mass injected at the source at the initial time. The total simulated mass at any given time is obtained by numerically integrating the simulated solute concentration over the entire computational grid. The result of the mass balance comparison for our two-dimensional example is shown in Table 3, which expresses mass balance error as a percentage of initial injected mass. For consistency the mass balance results for all four interpolation alternatives are computed with the nominal 121 by 61 grids specified in Table 1. Table 3 indicates that the mass balance errors for all interpolators are a few percent or less. This is acceptable for most applications, especially when considered together with the other measures evaluated in this paper.

4.3. Three-Dimensional Example

In our final example we consider solute transport in a three-dimensional heterogeneous two-scale isotropic time-invariant velocity field. This two-scale field is obtained by superimposing two statistically independent random replicates that have isotropic Gaussian spectral densities with the same mean and variance but different correlation scales. The large-scale velocity replicate, which describes the effect of regional variations in hydraulic conductivity, has a dimensionless correlation scale of 0.20 in each direction. The small-scale velocity replicate, which describes the effect of local fluctuations about the regional trend, has a dimensionless correlation scale of 0.04 in each direction. The mean velocity is aligned with the x axis, the domain dimensions are $L_x = L_y = L_z = 1$, and the simulation time is $T = 1$. Both the large- and small-scale velocity replicates are obtained from the spectrally based random field generator previously used in our two-dimensional example [*Ruan and McLaughlin*, 1998]. Table 4 summarizes the various dimensionless inputs for this example, including the log hydraulic conductivity statistics used in the random field generator. Without any loss of generality we use constant space and time steps.

Table 3. Comparison of Mass Balance Errors

Method	Time = 0.67	Time = 1.00
Linear	1.22%	1.87%
Quadratic	0.57%	0.87%
Cubic spline	0.61%	0.96%
Taut spline	1.07%	2.10%

Table 4. Input Parameters for the Three-Dimensional Simulation

Parameters	Values
Log k variance σ_f^2	1.0
Porosity ρ	0.3
Mean hydraulic gradient $\lvert J \rvert$	0.02
Domain length $L_x = L$	1
Domain width $L_y = L$	1
Domain height $L_z = L$	1
Grid size $N_x \times N_y \times N_z$	$51 \times 51 \times 51$
Normalized grid spacing $\Delta x/L = \Delta y/L = \Delta z/L$	0.02
Total simulation time T	1
Total time steps N_t	40
Normalized log K geometric mean $K_g T/L$	5.00
Normalized (small) log K correlation length $\lambda_{\mathrm{Small}}/L$	0.04
Normalized (large) log K correlation length $\lambda_{\mathrm{Large}}/L$	0.20
Grid Peclet number $Pe_x = Pe_y = Pe_z$	100
Longitudinal Courant number Cr	0.42
Normalized initial source dimension $\sigma_x/L \times \sigma_y/L \times \sigma_z/L$	$0.08 \times 0.08 \times 0.08$
Normalized initial source location $(x_0/L, y_0/L, z_0/L)$	(0.15, 0.5, 0.5)

Figure 6 shows simulated two-scale plumes originating at a continuous source located at dimensionless coordinates (0.15, 0.5, 0.5) near the right end of the simulation domain. All plumes are plotted at dimensionless time $t = 1.0$. Figures 6a, 6b, and 6c display the 0.05 concentration isosurface and selected concentration contours for the linear interpolator solution. Figures 6d, 6e, and 6f show the corresponding plots for the taut spline interpolator. A comparison of these solutions clearly shows the dispersive nature of the linear interpolator. The taut spline solution captures more details than the linear interpolator while almost entirely avoiding oscillations and negative concentrations. These 132,651 node transport simulations can be performed in a few hours on a 333 Mhz, 128 Mb DEC Alpha workstation.

5. Conclusions

This paper examines a particular class of numerical solution techniques that is well-suited for the advection-dominated solute transport simulations needed to test macrodispersion theories. These Eulerian-Lagrangian methods are able to deal with the steep gradients and complex plume geometries associated with highly variable (e.g., random) multidimensional velocity fields. Our numerical experiments indicate that the accuracy of a particular Eulerian-Lagrangian solver is highly dependent on the method used to interpolate velocity and concentration values at the nodes of the spatial computational grid. These experiments consider classical linear and quadratic interpolators as well as new cubic spline and taut spline interpolators, which are applied here for the first time to multidimensional variable velocity problems.

Our investigation confirms the common observation that the linear Eulerian-Lagrangian interpolator yields dispersive solutions while the quadratic interpolator yields oscillatory solu-

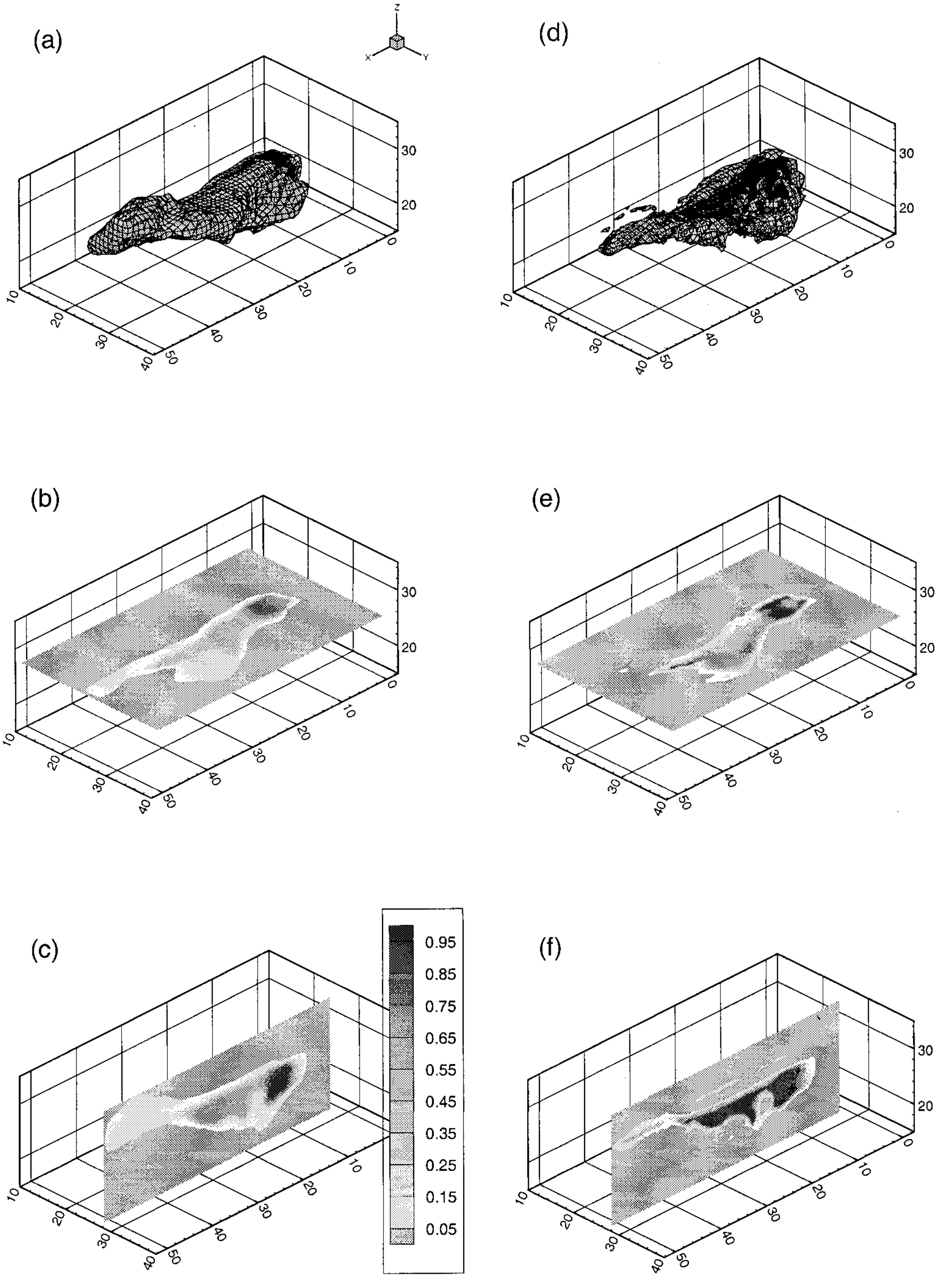

Figure 6. Concentration contours obtained with (a), (b), and (c) the linear interpolator and (d), (e), and (f) the taut spline interpolator in a three-dimensional random velocity field. Axis units refer to node numbers.

tions. The cubic spline interpolator is slightly oscillatory but much less so than the quadratic interpolator. The limited oscillations produced by the cubic spline interpolator appear to be restricted to areas with steep concentration gradients (primarily in the tail of the solute plume). The taut spline interpolator (a variant on the cubic spline) is slightly dispersive but much less so than the linear interpolator. The taut spline adaptively stretches the cubic spline approximating function to make it more nearly linear in the vicinity of sharp gradients. This essentially eliminates spurious oscillations while maintaining accurate solutions even when Peclet numbers are quite high.

The cubic and taut spline interpolators are superior to the linear and quadratic alternatives in several respects. First, visual inspection of simulated two-dimensional solute contour plots (Plate 1) indicates that the two spline interpolators are better able to capture sharp fronts and irregularities than the linear and quadratic interpolators. Second, the two spline interpolator computational times are comparable to or better than the linear and quadratic interpolators for a given level of convergence (as measured by the value convergence norm J_G). The cubic spline is especially efficient, with computation times approximately an order of magnitude less than those achieved by any of the other alternatives. These advantages are achieved without any significant sacrifice in mass balance error, which is a few percent or less for all four interpolators.

It should be noted that an alternative Eulerian-Lagrangian approach, the Eulerian-Lagrangian local adjoint method, has the advantage of conserving mass perfectly while providing a mathematically consistent implementation of boundary conditions [*Celia et al.*, 1990; *Ewing et al.*, 1994; *Healy and Russell*, 1998]. Since there has not yet been a direct comparison of the accuracy or computational performance of the cubic spline and local adjoint Eulerian-Lagrangian methods for a multidimensional variable velocity problem, it is not possible to say which approach is better for a given application. However, it is important to remember that solution accuracy, mass conservation, and computational efficiency are all important aspects of algorithm performance that need to be considered in any comparison.

The computational demands of traditional higher-order interpolators have tended to discourage their use in practical multidimensional applications. The cubic and taut spline interpolators described here rely on a tensor product decomposition of the n-dimensional spatial interpolation problem into a sequence of n one-dimensional problems. Each one-dimensional problem may be cast in a local (rather than a global) form, which yields a simple set of tridiagonal equations for the unknown interpolation coefficients. This enables the cubic and taut spline interpolators to achieve acceptable accuracy with a much coarser grid and smaller computation time than the linear and quadratic interpolators.

Several investigators have suggested that it is possible to improve the accuracy of linear Eulerian-Lagrangian solvers by adaptively refining the computational grid near steep concentration gradients [*Neuman*, 1981, 1984; *Yeh and Chang*, 1992]. Such adaptive schemes are attractive in concept but difficult to implement in practice, especially for multidimensional problems. The locations of steep concentration fronts change over time and are difficult to predict when the velocity field is heterogeneous [*Yeh and Chang*, 1992]. The cubic and taut spline interpolators, especially the latter, provide relatively simple alternatives to adaptive grid refinement. The spline consistently provides accurate approximations to sharp fronts by using higher-order interpolation and by requiring derivative continuity. The adaptive adjustments offered by the taut spline interpolator make it possible to achieve this accuracy without oscillations. Unlike Hermite polynomial interpolators, the cubic and taut spline interpolators do not require explicit input of concentration derivatives [*Holly and Preissmann*, 1977; *Glass and Rodi*, 1982]. This has the advantage of simplifying the solution algorithm while reducing computation time.

The tradeoff between oscillation and numerical dispersion is explicitly addressed in a number of contemporary transport solution methods, including Petrov-Galerkin methods with upstream weighting [*Dick*, 1983] and the total variance diminishing method [*Cox and Nishikawa*, 1991]. The total variance diminishing method is of particular relevance to the present discussion since, like the taut spline, it adjusts its local spatial approximation in response to local gradients. The upstream-weighted Petrov-Galerkin, total variance diminishing, and taut spline methods all provide some flexibility in trading off dispersion and oscillation. However, it is possible that the aggregate solution error, including the effects of both spurious oscillation and dispersion as measured by an appropriate error norm, is consistently lower for some methods than others [*Gresho and Lee*, 1979]. If so, the methods with lower aggregate error may be preferred, regardless of the position adopted regarding the relative merits of dispersion versus oscillation.

There is a need for a careful comparison of the various candidate methods for solving heterogeneous advection-dominated transport problems. Such a comparison should include the methods discussed in this paper, the total variance diminishing method, the Eulerian-Lagrangian local adjoint method, Petrov-Galerkin methods, and other higher-order implicit Eulerian methods, perhaps with multigrid capabilities. This comparison should include multidimensional advection-dominated test problems with heterogeneous velocity fields. It should also consider the effect of time step as well as space step on convergence.

Multidimensional solute transport problems are considerably more difficult when the velocity field is heterogeneous and advection dominates local dispersion. This is precisely the case encountered in theoretical and practical investigations of macrodispersion and small-scale concentration variability [*Tompson and Gelhar*, 1990]. Eulerian-Lagrangian methods are well suited for solutions of advection-dominated transport problems, provided that the required spatial interpolators are chosen carefully. Our study indicates that accurate and computationally efficient results can be obtained with the cubic spline and taut spline interpolators. This conclusion is confirmed by *Ruan* [1997], who uses the cubic spline method to generate an extensive set of high-resolution heterogeneous transport simulations. Both studies demonstrate that spline-based Eulerian-Lagrangian solvers are able to track sharp fronts and capture detailed spatial structure without introducing excessive numerical dispersion or spurious oscillation. For this reason we believe that the Eulerian-Lagrangian approach is an attractive option for numerical investigations of solute transport in heterogeneous media.

Appendix: The One-Dimensional Taut Spline Interpolator

The taut spline interpolator is a variant on the one-dimensional local cubic spline interpolator given in (15) [*de Boor*, 1978]. The taut spline approximation of $z(x)$ takes three forms, depending on the magnitude of a parameter η that measures the change in the first derivative $\hat{z}(x)$:

$$\hat{z}(x) = \begin{cases} \alpha z_l + (1-\alpha)z_{l+1} + C'(\Psi)z''_l + Dz''_{l+1} & \eta \le 1/3 \\ \alpha z_l + (1-\alpha)z_{l+1} + Cz''_l + Dz''_{l+1} & 1/3 < \eta < 2/3 \\ \alpha z_l + (1-\alpha)z_{l+1} + Cz''_l + D'(\Psi)z''_{l+1} & \eta \ge 2/3 \end{cases} \tag{A1}$$

where α, C, and D are defined as in (15) and the remaining quantities are defined as follows:

$$\eta = \frac{|z'_l|}{|z'_l| + |z'_{l+1}|}$$

$$C'(\Psi_l) = \left[\frac{\Psi_l(\alpha;\, 1-\eta) - \alpha}{\Psi''_l(1;\, 1-\eta)} \right] \Delta^2 x_l$$

$$D'(\Psi_l) = \left[\frac{\Psi_l(1-\alpha;\, \eta) - \alpha + 1}{\Psi''_l(1;\, \eta)} \right] \Delta^2 x_l$$

The discrete first derivatives z'_l and z'_{l+1} used to derive η are finite difference approximations computed from the data values z_l and z_{l+1}.

The function $\Psi_l(\alpha, \eta)$ determines the tautness of the approximating cubic polynomial. It is defined as follows:

$$\Psi_l(\alpha;\ \eta) = \beta_l(\eta;\ \gamma)\alpha^3 + [1 - \beta_l(\eta;\ \gamma)]\left[\frac{\alpha - \xi_l(\eta;\ \gamma)}{1 - \xi_l(\eta;\ \gamma)}\right]_+^3 \tag{A2}$$

where

$$\beta_l(\eta,\ \gamma) = \frac{1 - \gamma/3}{\eta} \tag{A3}$$

$$\xi_l(\eta,\ \gamma) = 1 - \gamma \min\left\{1 - \eta,\ \frac{1}{3}\right\}$$

and the operator $[\quad]_+$ is nonzero only when its argument is greater than zero. The parameter $0 \le \gamma \le 3$ is a smoothness or tension parameter that controls the magnitude of the tautness adjustment. The second spatial derivatives of Ψ_l appearing in the definition of $C'(\Psi_l)$ and $D'(\Psi_l)$ are taken with respect to the unnormalized spatial coordinate x.

When a computed η falls within the range (1/3, 2/3] in data interval l, the product $z''_l z''_{l+1}$ is less than or equal to 0, and there is no inflection point in the approximating cubic polynomial. In this case it is unlikely that the interpolated approximation will be oscillatory, and the local taut spline approximation is simply a cubic spline polynomial. When η falls outside this range, an inflection point exists, and the interpolating cubic spline polynomial is stretched (in effect, the order of the approximation is reduced) to help prevent oscillations. The unknown second derivative coefficients z''_l and z''_{l+1} used to construct the taut spline approximation are derived from a tridiagonal system of equations similar to those obtained for the classical cubic spline interpolator.

Acknowledgments. The authors wish to thank the three reviewers for their helpful comments and suggestions, which greatly improved the paper.

References

Ames, W. F., *Numerical Methods for Partial Differential Equations*, 3rd ed., Academic, San Diego, Calif., 1992.

Baptista, A. M., Solution of advection-dominated transport by Eulerian-Lagrangian methods using the backwards method of characteristics, Ph.D. dissertation, Mass. Inst. of Technol., Cambridge, 1987.

Baptista, A. M., E. E. Adams, and K. D. Stolzenbach, Eulerian-Lagrangian analysis of pollutant transport in shallow water, *Tech. Rep. 296*, R. M. Parsons Lab., Mass. Inst. of Technol., Cambridge, 1984.

Bentley, L. R., and G. F. Pinder, Eulerian-Lagrangian solution of the vertically averaged groundwater transport equation, *Water Resour. Res.*, *28*, 3011–3020, 1992.

Celia, M. A., and W. G. Gray, *Numerical Methods for Differential Equations: Fundamental Concepts for Scientific and Engineering Applications*, Prentice-Hall, Englewood Cliffs, N. J., 1992.

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

Cheng, R. T., V. Casulli, and S. N. Milford, Eulerian-Lagrangian solution of the convection-dispersion equation in natural coordinates, *Water Resour. Res.*, *20*, 944–952, 1984.

Chiang, C. Y., M. F. Wheeler, and P. B. Bedient, A modified method of characteristics technique and a mixed finite element method for simulation of groundwater solute transport, *Water Resour. Res.*, *25*, 1541–1549, 1989.

Cox, R. A., and T. Nishikawa, A new total variation diminishing scheme for the solution of advective-dominant solute transport, *Water Resour. Res.*, *27*, 2645–2654, 1991.

Dawson, C. N., T. F. Russell, and M. F. Wheeler, Some improved estimates for the modified method of characteristics, *SIAM J. Numer. Anal.*, *26*, 1487–1512, 1989.

de Boor, C., *A Practical Guide to Splines*, Springer-Verlag, New York, 1978.

de Boor, C., *Spline Toolbox: For Use With MATLAB*, Math Works, Inc., Natick, Mass., 1992.

Dee, D. P., Prescribed solution forcing method for model verification in hydraulic engineering, paper presented at 1991 National Conference on Hydraulic Engineering, Am. Soc. of Civ. Eng., Nasheville, July 1991.

Dick, E., Accurate Petrov-Galerkin methods for transient convection diffusion problems, *Int. J. Numer. Methods Eng.*, *19*, 1425–1433, 1983.

Dimou, K., 3-D hybrid Eulerian-Lagrangian/particle tracking model for simulating mass transport in coastal water bodies, Ph.D. dissertation, Mass Inst. of Technol., Cambridge, 1992.

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

Ewing, R. E., and T. F. Russell, Multistep Galerkin methods along characteristics for convection-diffusion problems, in *Advances in Computer Methods for Partial Differential Equations*, vol. IV, edited by R. Vichnevetsky and R. S. Stepleman, pp. 28–36, Int. Assoc. for Math. and Comput. in Simul., Rutgers Univ., New Brunswick, N. J., 1981.

Ewing, R. E., H. Wang, and R. C. Sharpley, Eulerian-Lagrangian localized adjoint methods for transport of nuclear-waste contamination in porous media, in *Computational Methods in Water Resources*, vol. X, edited by A. Peters et al., pp. 241–248, Kluwer Acad., Norwell, Mass., 1994.

Farin, G., *Curves and Surfaces for Computer Aided Geometric Design: A Practical Guide*, Academic, San Diego, Calif., 1993.

Gelhar, L. W., *Stochastic Subsurface Hydrology*, Prentice-Hall, Englewood Cliffs, N. J., 1993.

Glass, J., and W. Rodi, A higher order numerical scheme for scalar transport, *Comput. Methods Appl. Mech. Eng.*, *31*, 337–358, 1982.

Goode, D. J., Particle velocity interpolation in block-centered finite difference groundwater flow equation, *Water Resource. Res.*, *26*, 925–940, 1990.

Graham, W., and D. B. McLaughlin, Stochastic analysis of nonstationary subsurface solute transport, 1, Unconditional moments, *Water Resour. Res.*, *25*, 215–232, 1989.

Gresho, P. M., and R. L. Lee, Don't suppress the wiggles: They're telling you something!, *Comput. Fluids*, *9*, 223–253, 1979.

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

Healy, R. W., and T. F. Russell, Solution of the advection-dispersion equation in two-dimensions by a finite-volume Eulerian-Lagrangian localized adjoint method, *Adv. Water Resour.*, *27*, 11–26, 1998.

Ho, L. W., Y. Maday, A. Patera, and E. M. Ronquist, A high-order Lagrangian-decoupling method for the incompressible Navier-Stokes equations, *Comput. Methods Appl. Mech. Eng.*, *80*, 65–90, 1990.

Holly, F. M., and U. Jean-Marc, Dispersion simulation in two-dimensional tidal flow, *J. Hydraul. Eng.*, *110*, 905–925, 1984.

Holly, F. M., and A. Preissmann, Accurate calculation of transport in two dimensions, *J. Hydraul. Eng.*, *103*, 1259–1277, 1977.

LaBolle, E. M., G. E. Fogg, and A. F. B. Tompson, Random-walk simulation of transport in heterogeneous porous media: Local mass-conservation problem and implementation methods, *Water Resour. Res.*, *32*, 583–593, 1996.

Lapidus, L., and G. F. Pinder, *Numerical Solution of Partial Differential Equation in Science and Engineering*, John Wiley, New York, 1982.

Neuman, S. P., A Eulerian-Lagrangian numerical scheme for the dispersion-convection equation using conjugate space-time grids, *J. Comput. Phys.*, *41*, 270–294, 1981.

Neuman, S. P., Adaptive Eulerian-Lagrangian finite element method for advection-dispersion, *Int. J. Numer. Methods Eng.*, *20*, 321–337, 1984.

Ogata, A., A solution of the differential equation of longitudinal dispersion in porous media, *U.S. Geol. Surv. Prof. Pap.*, *p411A*, A1–A7, 1961.

Oliveira, A., and A. M. Baptista, Integration or interpolation ELMs?, *Computational Methods in Water Resources*, vol. X, edited by A. Peters et al., pp. 333–340, Kluwer Acad., Norwell, Mass., 1994.

Peaceman, D. W., Improved treatment of dispersion in numerical calculation of multidimensional miscible displacements, *Soc. Pet. Eng. J.*, *6*, 213–216, 1966.

Press, W. H., B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, *Numerical Recipes: The Art of Scientific Computing*, Cambridge Univ. Press., New York, 1986.

Rovelstad, A. L., Lagrangian analysis of vorticity transport in a numerical simulated turbulent channel flow, Ph.D. dissertation, Univ. of Md., College Park, 1991.

Rovelstad, A. L., The effects of interpolation errors on the Lagrangian analysis of simulated turbulent channel flow, *J. Comput. Phys.*, *110*, 190–195, 1994.

Ruan, F., *A Numerical Investigation of Solute Plumes in Nested Multi-Scale Porous Media*, Ph.D. thesis, Mass. Inst. of Technol., Cambridge, 1997.

Ruan, F., and D. B. McLaughlin, An efficient multivariate random field generator using the fast Fourier transform, *Adv. Water Resour.*, *21*, 385–399, 1998.

Sudicky, E. A., The Laplace transform Galerkin technique: A time-continuous finite element theory and application to mass transport in groundwater, *Water Resour. Res.*, *25*, 1833–1846, 1989.

Tompson, A. F. B., and L. W. Gelhar, Numerical simulation of solute transport in three-dimensional, randomly heterogeneous porous media, *Water Resour. Res.*, *26*, 2541–2562, 1990.

Valocchi, A., and M. Malmstead, Accuracy of operator splitting for advection-dispersion-reaction problems, *Water Resour. Res.*, *28*, 1471–1476, 1992.

Wait, R., and A. R. Mitchell, *Finite Element Analysis and Applications*, John Wiley, New York, 1985.

Wheeler, M. F., and C. N. Dawson, An operator splitting method for advection-diffusion-reaction problems, in *Mathematics of Finite Elements and Applications (MAFELAP) Proceedings*, vol. VI, edited by J. A. Whiteman, pp. 463–482, Academic, San Diego, Calif., 1988.

Xiang, W., and S. Zhen, The bivariate spline approximate solution to the hyperbolic equation with variable coefficients, *J. Comput. Math.*, *10*, 112–128, 1992.

Yeh, G. T., and J. R. Chang, An exact peak capturing and oscillation-free scheme to solve advection-dispersion transport equations, *Water Resour. Res.*, *28*, 2937–2951, 1992.

Yeung, P. K., and S. B. Pope, An algorithm for tracking fluid particles in numerical simulation of homogeneous turbulence, *J. Comput. Phys.*, *79*, 373–416, 1988.

D. McLaughlin, Ralph M. Parson's Laboratory, Department of Civil and Environmental Engineering, Massachusetts Institute of Technology, Cambridge, MA 02139. (dennism@mit.edu)

F. Ruan, Applications Research, GeoQuest, Schlumberger, 5599 San Felipe, Suite 1700, Houston, TX 77056. (fruan@houston.geoquest.slb.com)

(Received September 15, 1998; revised February 12, 1999;
accepted February 12, 1999.)