A characteristic-based finite analytic method for solving the two-dimensional steady state advection-diffusion equation

Thomas Lowry
Lincoln Environmental, Lincoln, Canterbury, New Zealand

Shu-Guang Li
A123 Engineering Research Complex, Department of Civil and Environmental Engineering, Michigan State University
East Lansing, Michigan, USA

Received 16 March 2001; revised 25 January 2002; accepted 25 January 2002; published 30 July 2002.

[1] This paper develops an improved finite analytic (FA) solution method to the advection-diffusion equation (ADE) for solving advection-dominated steady state transport problems. The FA method solves the ADE analytically in localized, discrete elements, with each element linked through local boundary conditions. Previous FA methods have suffered from complex solution formulations as well as from numerical dispersion stemming from inaccuracies in the local boundary estimations. Here we use finite difference approximations of the dispersion terms to reduce solution complexity, implement an improved particle-tracking scheme to account for velocity variations within each element, and use a Hermite interpolation scheme to estimate the local boundary conditions. The new FA method, previous FA methods, and a finite difference upwinding method are compared in both homogeneous and heterogeneous velocity fields at different Peclet numbers. Results show the new FA method exhibits little numerical dispersion without undue complexity or computational effort across all tested flow conditions.


1. Introduction

[2] Increasingly, transport predictions in many fields are being relied upon as a basis for practical decision-making. Issues ranging from turbulence modeling, to design criteria for thermal reactors, to well-head protection in groundwater all rely, in one form or another, on the accurate solution of the advection-diffusion equation. In addition, accurate predictions are also important in helping to further our understanding, through scientific research, of many transport phenomena. However, general solution methods to the advection-diffusion equation are typically hindered by one or more of three general drawbacks that limit the usefulness of transport predictions: numerical dispersion, spurious oscillations, and undue computational effort. In this paper we concentrate our attention on reducing these numerical difficulties by focusing on a special variant of the finite difference method called the finite analytic method, for solving partial differential equations [Chen and Li, 1979; Chen and Chen, 1984; Chen, 1988; Li et al., 1992; Li and Wei, 1998] and address some of the limitations to improving the method’s accuracy and applicability.

[3] The finite analytic method is an Eulerian method that solves the governing equation analytically in discrete elements. What separates the finite analytic method from other numerical methods is that it does not tamper with the differentials of the governing equation, as with the finite difference method, nor does it need basis functions to approximate a solution of the integral form of the governing equation, as with finite element methods [Chen and Chen, 1984; Hwang et al., 1985]. The idea of the finite analytic method is to represent the modeling domain as a series of homogeneous, constant parameter elements. The governing equation is then solved analytically in each computational element to obtain algebraic representations of the concentration at each node that are then overlapped to cover the entire region of the problem. Because of its built-in analytical nature, the finite analytic method naturally and systematically takes into account the character of the differential equation, exhibiting a gradual and analytically based upwind shift depending on the direction of flow [Chen and Chen, 1984; Li and Wei, 1998].

[4] Until the work of Li and Wei [1998], the finite analytic method suffered from undue complexity, rendering a formulation [e.g., Chen et al., 1981; Chen and Chen, 1984] that was computationally heavy to implement while still showing some numerical dispersion. Li and Wei removed the complexity of the solution for two-dimensional transport by estimating the second-order dispersion terms using a central difference approximation. Substitution of these approximations back into the original governing equation results in a hyperbolic partial differential equation that is then solved analytically using the method of characteristics. In addition, Li and Wei added an
improved boundary estimation that utilizes information
in the upstream direction that reduces the numerical
dispersion that is persistent with traditional boundary
formulations.

This paper improves the Li and Wei [1998] method by
introducing an improved particle-tracking scheme that
accounts for velocity variations across the local element,
as well as a higher-order boundary interpolation for deter-
miming the local boundary condition. The result is a steady
state solution method that is computationally efficient as
well as numerically accurate across a wide range of flow
conditions and Peclet numbers. Future publications (T.
Lowry and S.-G. Li, A Laplace transform finite analytic
method for solving the two-dimensional time-dependent
advection-diffusion equation, submitted to Water Resources
Research, 2001; hereinafter referred to as submitted manu-
script, 2001) will outline the extension of this new finite
analytic method in conjunction with the space-time accurate
technique of Li et al. [1992] to unsteady solute transport
problems in heterogeneous, two- and three-dimensional
velocity fields.

2. The Finite Analytic Method

This section presents the derivation of the finite
analytic method developed in this research. It is an exten-
sion of the work of Wei [1995], Li and Wei [1998], and
Lowry [2000], and thus the reader is encouraged to study
those papers for more information.

Consider the two-dimensional, steady state, advec-
tion-diffusion equation for a contaminant undergoing first-
order decay with internal sources and sinks. The governing
equation for this problem is

\[
\begin{align*}
\frac{\partial C}{\partial x} + v \frac{\partial C}{\partial y} &= \frac{\partial}{\partial x} \left[ D_{xx}(x,y) \frac{\partial C}{\partial x} \right] + \frac{\partial}{\partial y} \left[ D_{yy}(x,y) \frac{\partial C}{\partial y} \right] \\
&+ \frac{\partial}{\partial x} \left[ D_{xy}(x,y) \frac{\partial C}{\partial y} \right] + \frac{\partial}{\partial y} \left[ D_{yx}(x,y) \frac{\partial C}{\partial x} \right] \\
&+ \frac{\partial}{\partial x} \left[ D_{ys}(x,y) \frac{\partial C}{\partial y} \right] + \frac{\partial}{\partial y} \left[ D_{ys}(x,y) \frac{\partial C}{\partial x} \right] - \lambda C + S(x,y),
\end{align*}
\]

where \( C \) is the solute concentration as a function of \( x \) and \( y \);
\( x \) and \( y \) are the spatial coordinates (\( 0 \leq x < \infty, 0 \leq y < \infty \));
\( u \) is the velocity in the \( x \) direction; \( v \) is the velocity in the \( y \) direction;
\( D_{xx}, D_{yy}, D_{xy}, \) and \( D_{yx} \) are the dispersion coefficients; \( \lambda \) is the first-order decay constant; and \( S \) is a
source-sink term.

The premise of the finite analytic method is to solve
the governing equation analytically in a local, regularly
shaped element (Figure 1). The first step, as in the finite
difference method, is to discretize the solution domain into
discrete, rectangular elements. For illustration purposes we
will consider \( \Delta x \) constant throughout the grid and equal to
\( \Delta y \), although this is not a requirement. Thus the length of
each side of the element in the \( x \) and \( y \) directions is \( 2 \Delta x \) and
\( 2 \Delta y \), respectively.

Within an element centered at \( i,j \), equation (1) is
written as

\[
\begin{align*}
u_{i,j}(x,y) \frac{\partial C}{\partial x} + v_{i,j}(x,y) \frac{\partial C}{\partial y} &= \frac{\partial}{\partial x} \left[ D_{xx}(x,y) \frac{\partial C}{\partial x} \right] + \frac{\partial}{\partial y} \left[ D_{yy}(x,y) \frac{\partial C}{\partial y} \right] + \frac{\partial}{\partial x} \left[ D_{xy}(x,y) \frac{\partial C}{\partial y} \right] \\
&+ \frac{\partial}{\partial y} \left[ D_{yx}(x,y) \frac{\partial C}{\partial x} \right] - \lambda C \\
&+ S_{i,j},
\end{align*}
\]
where all coefficients are defined locally as above within the area $x_{j-1} \leq x \leq x_{j+1}$ and $y_{i-1} \leq y \leq y_{i+1}$. Equation (2) is a partial differential equation defined on a rectangular grid describing the variation of concentration within each local element. If all coefficients are assumed constant within the local element, as well as proper local boundary conditions, equation (2) can be solved using the technique of separation of variables [Chen and Chen, 1984]. The resulting analytical solution represents the concentration in the local element as a function of $x$ and $y$ and is evaluated and rearranged such that the concentration at the central node $P$ is a weighted average of the surrounding nodal concentrations, each time an analytically based coefficient. Depending on the formulation of the local element boundary conditions, however, the analytic coefficients can be quite messy. In their traditional form [Chen and Chen, 1984] the coefficients include at least one infinite series of exponential functions. The evaluation of these series is expensive to compute, especially for problems in variable velocity fields, since each summation must be evaluated at each node. To gain accuracy, different boundary approximations may be imposed, but each step in accuracy adds to the computational effort.

(10) To circumvent this problem, we employ an alternative approximate technique for solving equation (2), as presented by Li and Wei [1998]. Examination of the analytical solution reveals that the elliptic diffusion terms are responsible for the complications in the analytical coefficients [Carrier, 1976], and it is widely known that numerical solutions are complicated by the hyperbolic first-order advection terms [Li et al., 1992]. To take advantage of this, the analytically difficult dispersion terms are first approximated numerically prior to solving the resulting partial differential equation analytically. Finite difference approximations of the dispersion terms are highly accurate [Chang and Chen, 1987], and thus little is lost with this approach. Starting with the discretized domain and equation (2), the dispersion terms are evaluated using a central difference scheme and are represented as follows:

$$\frac{\partial}{\partial x} D_{xx} \frac{{\partial}C}{{\partial}x} = \frac{1}{\Delta x^2} \left( D_{xx(i+1/2)} C_{i+1,j} - (D_{xx(i+1/2)} + D_{xx(i-1/2)}) C_{i,j} + D_{xx(i-1/2)} C_{i-1,j} \right) = f_{xx,i}$$

(3)

$$\frac{\partial}{\partial y} D_{yy} \frac{{\partial}C}{{\partial}y} = \frac{1}{\Delta y^2} \left( D_{yy(j+1/2)} C_{i,j+1} - (D_{yy(j+1/2)} + D_{yy(j-1/2)}) C_{i,j} + D_{yy(j-1/2)} C_{i,j-1} \right) = f_{yy,j}$$

(4)

$$\frac{\partial}{\partial x} D_{xy} \frac{{\partial}C}{{\partial}y} = \frac{1}{\Delta x \Delta y} \left[ D_{xy(i+1/2,j+1/2)} (C_{i+1,j+1} + C_{i,j+1} - C_{i+1,j} - C_{i,j}) + D_{xy(i+1/2,j-1/2)} (C_{i+1,j-1} + C_{i,j-1} - C_{i+1,j} - C_{i,j-1}) \right] = f_{xy}$$

(5)

$$\frac{\partial}{\partial y} D_{xy} \frac{{\partial}C}{{\partial}x} = \frac{1}{\Delta x \Delta y} \left[ D_{xy(i+1/2,j+1/2)} (C_{i+1,j+1} + C_{i,j+1} - C_{i+1,j} - C_{i,j}) + D_{xy(i-1/2,j+1/2)} (C_{i-1,j+1} + C_{i,j+1} - C_{i-1,j} - C_{i,j}) \right] = f_{yx}$$

(6)

The diffusion coefficients are evaluated at the cell interfaces using the arithmetic average of the two closest nodes. Inserting $f_{xx,i}, f_{yy,j}, f_{xy,j}$ and $f_{yx,i}$ into equation (2) gives

$$u_{ij}(x,y) \frac{\partial C}{\partial x} + v_{ij}(x,y) \frac{\partial C}{\partial y} + \lambda C = f_{xx,i} + f_{yy,j} + f_{xy} + f_{yx} + S_{ij}.$$  

(7)

This is a hyperbolic partial differential equation that can be solved analytically using the method of characteristics. After evaluation at node $P$ we get the following solution:

$$C_{ij} = e^{-\lambda h} \left[ C_{i-1,j} - f_{xx,i} + f_{yy,j} + f_{xy} + f_{yx} + S_{ij} \right]$$

(8)

where $P$ is the point where the flow line that passes through node $P$ crosses the local element boundary in the upstream direction (Figure 1); $C_{p'}$ is the concentration at $p'$, and $b_t$ is equal to travel time along the length of the solution characteristic from node $P$ to the element boundary.

[11] Generally, the spatial coordinates of $p'$ will not coincide with a nodal point, and thus $C_{p'}$ will need to be interpolated from surrounding nodal values. The simplest method is to use linear interpolation between the two closest nodal points on the boundary segment where $p'$ lies. When one examines the grid layout, however, it can be seen that better information on the boundary can be obtained if concentration information from nodes in the upstream direction is used (Figure 1). $C_{p'}$ can then be interpolated between a point $p''$, which is the location on the boundary where the flow line that crosses the next upstream node crosses the boundary, and the next closest nodal value ($i-1,j$ for the example in Figure 1). This is the improved boundary function as described by Li and Wei [1998].

[12] The concentration at $p''$ is derived by using the improved finite analytic (IFA) solution given in equation (8), except the solution is evaluated at the upstream source node rather than at node $P$. Thus the equation for the concentration at $p''$ is written as

$$C_{p''} = e^{-\lambda h} \left[ C_{mn} - f_{xmn} + f_{ymn} + f_{xmn} + f_{ymn} + S_{mn} \right]$$

(9)

where the subscripts $m$ and $n$ refer to the $x$ and $y$ cell indices of the upstream source node, $f_t$ is the travel time from node $m,n$ to the element boundary, and all like terms are defined around node $m,n$.

[13] When one examines equations (8) and (9), two possible areas of inaccuracy are apparent; first is the determination of $b_t$ and $f_t$, which have the units of time and are the advective travel times between each node and the element boundary, and second is the determination of $C_{p'}$, which is the concentration on the element boundary where the solution characteristic crosses the boundary. Li and Wei [1998] use a straight-line estimation based on a constant velocity assumption at the nodal points to determine $b_t$, $f_t$, and the positions of $p'$ and $p''$. In addition, they utilize a linear interpolation between $p''$ and the nodal end point (e.g., $i-1,j$) to evaluate $C_{p'}$. Here we improve these boundary
estimations by employing an improved particle-tracking algorithm that accounts for the variability in velocities, to numerically integrate the solution characteristic and determine \( b_i \) and \( f_i \). Second, we utilize a "two-point higher-order" interpolation scheme, sometimes called Hermite interpolation [Holly and Preissmann, 1977], to evaluate the concentration \( C_{p'} \). The Hermite scheme interpolates the concentration at \( p' \) by forming a quadratic equation using the concentrations and the concentration derivatives at each end point. As will be shown, these improvements drastically reduce numerical dispersion and improve plume trajectory, especially for cases with complicated velocity fields.

3. Determination of the Finite Analytic Coefficients

[14] In this section we present the calculation of the finite analytic coefficients incorporating the use of the particle-tracking algorithm as well as Hermite interpolation. It should be noted that the position of \( p' \), \( p'' \), the upstream source node \((m,n)\), and the interpolation nodal end point \((e.g., i-1,j)\) all change depending on the direction of flow. For that reason we will present the calculations based on a single flow direction as shown in Figure 1 of \( u \geq 0, v \geq 0, \) and \( u \geq v \). Extension to other flow directions is a straightforward exercise and is described completely by Lowry [2000]. In addition, all derivations are performed assuming dispersion coefficients are constant within each local element \((e.g., D_{x(i+1)/2,j}, D_{xi-1/2,j})\), which simplifies equations (3)–(6). For advection-dominated cases, comparisons of results with and without this assumption show negligible impact on model predictions.

[15] The entire derivation consists of five steps: (1) Track a particle backward from node \( P \) to find the position of \( p' \); (2) on the basis of flow direction at \( p' \), choose the upstream source node and track a particle forward to find the position of \( p'' \); (3) evaluate the concentrations and concentration derivatives at \( p'' \), using equation (9); (4) interpolate the respective values at \( p' \) and (5) use equation (8) to calculate the concentration and derivatives at node \( P \). All these steps are nested such that an algebraic representation of the concentration (and the derivatives) can be formed at each node in the grid that consists of the concentration and the concentration derivatives at the surrounding nodes, each times an analytically based coefficient.

3.1. Particle Tracking Algorithm

[16] To accomplish steps 1 and 2, particles must be tracked backward from node \( P \) to the element boundary as well as forward from the upstream source node \((m,n)\) to the element boundary. This determines not only the travel times \( b_i \) and \( f_i \) but also the positions on the boundary of \( p' \) and \( p'' \). Generally, most particle-tracking codes use linear or bilinear interpolation to determine the velocity field across the modeling domain, although other schemes such as bicubic interpolation are sometimes used [Anderson and Woesner, 1992; Ruan and McLaughlin, 1999]. Here bilinear interpolation is used, which assumes that for any given \( x \) coordinate, the velocity varies linearly with \( y \), and for any given \( y \) coordinate, the velocity varies linearly with \( x \).

[17] The complete tracking process starts with a particle at node \( i,j \) and tracks it backward until it reaches the element boundary. This is position \( p' \). On the basis of flow direction at \( p' \), an upstream source node is then chosen \((i-2,j-1)\), from which a particle is tracked forward to the boundary. This is the location of \( p'' \). The determination of these positions only needs to occur once at the beginning of the simulation, as the positions remain unchanged for steady state velocity fields. During particle tracking, the total tracking times at each node are saved for use as the \( b_i \) and \( f_i \) terms in equations (8) and (9). With \( f_i \) known, \( C_{p'} \) is then calculated directly using equation (9). The algebraic representation for the concentration at \( p'' \) becomes

\[
C_{p'} = R_f C_{i-2,j-1} + R_{yf} (C_{i-3,j-1} + C_{i-1,j-1}) \\
+ R_{yf} (C_{i-2,j-2} + C_{i-2,j-1}) + R_{yf} (C_{i-1,j} - C_{i-3,j} + C_{i-3,j-2} \\
- C_{i-1,j-2}) + R_{yf} S_{i-2,j-1}
\]

with all coefficients defined in Table 1.

3.2. Hermite Interpolation

[18] Once the locations of \( p' \) and \( p'' \) are known, and the concentration at \( C_{p'} \) is calculated, a "two-point higher-order" interpolation scheme [Holly and Preissmann, 1977], commonly called Hermite interpolation, is used to determine the concentration at \( C_{p''} \). Even though the modeling domain is two-dimensional, the interpolation is performed on a local element boundary segment, and thus a one-dimensional interpolation scheme is all that is needed. Previous efforts of Li and Wei [1998] used a linear interpolation to determine \( C_{p''} \).

[19] For the example case shown in Figure 1, the concentrations at each interpolation end point are \( C_{i-l,j} \) and \( C_{p''} \), and the concentration derivatives in the \( y \) direction at each end point are \( C_{Yi-l,j} \) and \( C_{Yp''} \). We can construct a third-degree interpolating polynomial between these two end points using the concentration values and the derivatives such that

\[
C(r_d) = A r_d^3 + B r_d^2 + D r_d + E,
\]

where \( r_d \) is analogous to the Courant number and represents the relative distance along the interpolation segment that \( p' \) lies from the far end point (Figure 1). For this example, \( r_d \) is defined as

\[
r_d = \frac{y_p - y_{p'}}{y_j - y_{p'}}
\]

where \( y \) is the \( y \) coordinate of the respective subscript.

[20] For the example case, the four coefficients \( A, B, D \), and \( E \) can be evaluated using the following four conditions:

\[
C(1) = C_{i-l,j}, \quad C(0) = C_{p''}, \quad CY(1) = CY_{i-l,j}, \quad CY(0) = CY_{p''},
\]

where \( CY \) indicates the first spatial \( y \) derivative. Performing the necessary algebra to evaluate \( A, B, D, \) and \( E \), and substituting them into (11) and simplifying, one obtains

\[
C(r_d) = \alpha_1 C_{i-l,j} + \alpha_2 C_{p''} + \alpha_3 CY_{i-l,j} + \alpha_4 CY_{p''},
\]
where

\[ \alpha_1 = r_d^2 (3 - 2r_d); \quad \alpha_2 = 1 - \alpha; \]

\[ \alpha_3 = r_d^2 (1 - r_d) \Delta ep; \quad \alpha_4 = -r_d (1 - r_d)^2 \Delta ep \]

and \( \Delta ep \) is the interval length between \( i-j \) and \( p'' \) (defined as \( y_j - y_{p''} \) for this example).

[21] It should be noted that only the derivative (\( x \) or \( y \)) that is aligned in the same direction as the boundary segment where the interpolation occurs is used in equations (13) and (14), which is the \( y \) derivative for the example case. Equations (13), (14), and (15) change depending on the flow direction and must be adjusted as needed [see Lowry, 2000]. To complete the concentration interpolation, we set \( C'_{p'} = C(r_d) \) for use in equation (8).

### 3.3. Calculation of Concentration Derivatives

[22] The use of the concentration derivatives in the Hermite interpolation means that the derivatives must also be directly calculated at each nodal point and at \( p'' \), and interpolated at \( p' \). Taking both the \( x \) and \( y \) derivatives of equation (2), we see that the transport of the concentration derivatives is accomplished in exactly the same manner as the transport of the concentration, with the addition of an extra “decay” term and another term involving the opposite derivative, both of which account for spatially nonconstant velocity fields. For the \( y \) derivative in the example case, the governing partial differential equation within each finite analytic element is

\[
u(x, y) \frac{\partial (CY)}{\partial x} + v(x, y) \frac{\partial (CY)}{\partial y} = D_{x,y} \frac{\partial^2 (CY)}{\partial x^2} + D_{x,y} \frac{\partial^2 (CY)}{\partial y^2} + D_{x,y} \frac{\partial^2 (CY)}{\partial y \partial x} + D_{x,y} \frac{\partial^2 (CY)}{\partial x \partial y}
\]

\[ - \left( \lambda + \frac{\partial^2 (CY)}{\partial y^2} \right) CY + \frac{\partial CY}{\partial y} \]

\[ + SY_{i,j}, \quad (16) \]

where \( CX \) and \( CY \) are the \( x \) and \( y \) partial first spatial derivatives of \( C \), and \( SY \) is the \( y \) derivative of the source/sink function. If we represent the dispersion terms, the acceleration terms (\( \partial^2 \), \( \partial y \), and \( \partial x \)), and the \( CX \) term by finite difference approximations, equation (16) is of the same form of equation (7) and is solved using the finite analytic method described here. The solutions for the derivative at both \( p'' \) and \( i,j \) have the same form as equation (10) but with the added acceleration and \( CX \) terms. The \( x \) derivative equation is derived in exactly the same manner as equation (16). To find the \( y \) derivative of \( C_{p'} \), designated as \( CY_{p'} \), we must solve for the derivative of \( C(r_d) \). Taking the derivative of \( C(r_d) \) gives

\[ CY(r_d) = 3Ar_d^2 + 2Br_d + D. \]

Substituting the known coefficients A, B, and D into (17) and simplifying, we get for the example case

\[ CY(r_d) = \beta_1 C_{i,j-1} + \beta_2 C_{p'} + \beta_3 CY_{i,j-1} + \beta_4 CY_{p'}, \]

where

\[ \beta_1 = 6r_d (r_d - 1) \Delta ep^{-1}; \quad \beta_2 = -\beta; \]

\[ \beta_3 = r_d (3r_d - 2); \quad \beta_4 = (r_d - 1) (3r_d - 1). \]

To complete this derivation, we set \( CY_{p'} = CY(r_d) \).

[23] To help minimize computational effort, linear interpolation, rather than Hermite interpolation, is used to propagate the “off-derivative,” which is the derivative that is not aligned with the boundary segment (the \( x \) derivative for vertical boundary segments, as in this case, and the \( y \) derivative for horizontal boundary segments). This eliminates the need to calculate the cross derivative that would otherwise be needed to compute the off-derivative. Since the derivative values are somewhat self-correcting [Holly and Preissmann, 1977] the overall loss in accuracy with this technique is minimal. Thus the off-derivative at \( p' \) for the example case is

\[ CX_{p'} = r_d CX_{i,j-1} + (1 - r_d) CX_{p'} \]

### 3.4. Final Form

[24] To obtain the concentration at node \( P \) for our example case, we substitute the expression for \( C_{p'} \) and \( CY_{p'} \)
(equation 10) and the \( y \) derivative form of (10) into equation (14) and in turn substitute that result into equation (8). After collecting terms and rearranging, we get
\[
\begin{align*}
C_{i,j} &= A_1 C_{i,j-1} + A_2 C_{i,j+1} + A_3 C_{j-1, j} + A_4 C_{j, j+1} + A_5 C_{i-1, j} + A_6 C_{i-1, j+1} + A_7 C_{i+1, j-1} + A_8 C_{i+1, j+1} + A_9 C_{j+1, j+1} + A_{10} C_{i+1, j+1} + A_{11} C_{i, j+1} + A_{12} C_{i, j+1} + A_{13} C_{i, j} + A_{14} C_{i, j} + A_{15} C_{i, j}
\end{align*}
\]
where \( A_1, A_2, \ldots, A_{15} \) are given in Table 2 and Table 1.

Likewise for the \( y \) derivative, we substitute the expression for \( C_{y'} \) and \( C_{y''} \) into equation (18) and in turn substitute that result into the \( y \) derivative form of equation (8). After collecting terms and rearranging, we get
\[
\begin{align*}
C_{Y_i,j} &= B_1 C_{i-1, j} + B_2 C_{i, j} + B_3 C_{i+1, j} + B_4 C_{i, j+1} + B_5 C_{i, j-1} + B_6 C_{j, j} + B_7 C_{j, j+1} + B_8 C_{j, j-1} + B_9 C_{j+1, j} + B_{10} C_{j, j+1} + B_{11} C_{j, j-1} + B_{12} C_{j+1, j}
\end{align*}
\]
where \( B_1, B_2, \ldots, B_{12} \) are given in Table 3 and Table 1.

### Table 3. Coefficients for Use in Equation (22)

<table>
<thead>
<tr>
<th>Coefficient</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>( B_1 )</td>
<td>((1/R_{e_x})e^{-(y_{x})}y_{z}R_{y_{x}}/R_{y_{z}})</td>
</tr>
<tr>
<td>( B_2 )</td>
<td>((1/R_{e_x})e^{-(y_{x})}y_{z}R_{y_{x}}/R_{y_{z}})</td>
</tr>
<tr>
<td>( B_3 )</td>
<td>((1/R_{e_x})e^{-(y_{x})}y_{z}R_{y_{x}}/R_{y_{z}})</td>
</tr>
<tr>
<td>( B_4 )</td>
<td>((1/R_{e_x})e^{-(y_{x})}y_{z}R_{y_{x}}/R_{y_{z}})</td>
</tr>
<tr>
<td>( B_5 )</td>
<td>((1/R_{e_x})e^{-(y_{x})}y_{z}R_{y_{x}}/R_{y_{z}})</td>
</tr>
<tr>
<td>( B_6 )</td>
<td>((1/R_{e_x})e^{-(y_{x})}y_{z}R_{y_{x}}/R_{y_{z}})</td>
</tr>
<tr>
<td>( B_7 )</td>
<td>((1/R_{e_x})e^{-(y_{x})}y_{z}R_{y_{x}}/R_{y_{z}})</td>
</tr>
<tr>
<td>( B_8 )</td>
<td>((1/R_{e_x})e^{-(y_{x})}y_{z}R_{y_{x}}/R_{y_{z}})</td>
</tr>
<tr>
<td>( B_9 )</td>
<td>((1/R_{e_x})e^{-(y_{x})}y_{z}R_{y_{x}}/R_{y_{z}})</td>
</tr>
<tr>
<td>( B_{10} )</td>
<td>((1/R_{e_x})e^{-(y_{x})}y_{z}R_{y_{x}}/R_{y_{z}})</td>
</tr>
<tr>
<td>( B_{11} )</td>
<td>((1/R_{e_x})e^{-(y_{x})}y_{z}R_{y_{x}}/R_{y_{z}})</td>
</tr>
<tr>
<td>( B_{12} )</td>
<td>((1/R_{e_x})e^{-(y_{x})}y_{z}R_{y_{x}}/R_{y_{z}})</td>
</tr>
</tbody>
</table>

### 3.5. A Word About Notation

All terms defined around the upstream source node are designated with an \( f \) subscript, to denote the forward tracking necessary from that node. Similarly, the \( b \) subscript pertains to terms defined around node \( P \), to denote the backward tracking that occurs from that node. The \( x \) and \( y \) subscripts are used to distinguish those terms that are part of the \( x \) or \( y \) derivative equations, respectively. Since both the concentration values and the derivatives are used to determine the algebraic coefficients, a “mix” of these subscripts are found in the definition of the coefficients. While adding to the notational complexity, this format allows identification of the “source” of each term in the coefficients.

Equations (21), (22), and (23) are specific to the flow direction shown in Figure 1 with \( A_{1-17} \), \( B_{1-17} \), and \( B_{1-12} \) being the analytically based algebraic coefficients. For nonreactive transport where \( \lambda = 0 \), we apply L’Hospital’s rule [Wei, 1995] to the terms with \( \lambda \) in the denominator, \( (R_{eb}, R_{eby}, R_{ebx}, R_{efy}, R_{efy}, R_{eby}) \), which are then reduced to the particle travel times, \( b, o, f \), depending on the respective subscript. For the \( R \) terms that include the acceleration terms, this substitution applies only when the acceleration terms are zero.

When one compares this solution to that of most grid-based solution methods, it can be seen that 15 nodes (not including node \( P \)) rather than the usual eight nodes are used and that the 15 nodes are distributed in the upstream direction (Figure 2). For advection-dominated cases, the nodes closest to the streamline that passes through node \( i, j \) are the most heavily weighted. For pure advection, equation (21) reduces to two nodes and their derivatives, the
4. Examples and Discussion

In this section we compare the method developed here, which we call the improved finite analytic with Hermite and particle tracking (IFAHPT) method, the Li and Wei method (improved finite analytic, or IFA), the IFA method without the improved boundary condition (FA), and the upwinding finite difference method (FD-UW) in four different cases at varying Peclet numbers. The FA method used here is similar to the other two finite analytic methods (IFAHPT and IFA) but determines the concentration at \( p' \) by linear interpolation between the two closest nodal points, rather than using upstream information. Comparisons by Li and Wei [1998] show that this form of the FA method has accuracy and performance similar to traditional FA methods [e.g., Chen and Chen, 1984].

Each case will model a groundwater solute transport problem consisting of a constant source concentration in the lower left-hand corner of a rectangular modeling domain. The boundaries are selected sufficiently far away from the source so that the concentration gradients on the boundaries are small enough that a no-diffusive/dispersive solute flux assumption can be used. This type of boundary condition is sometimes called an “open” boundary condition. The first case compares the different solution methods in a uniform flow field at different flow angles, to illustrate the improvements of utilizing upstream information. The second case is a sinusoidal velocity field angled at a one-third cross flow \((u = 3v)\) in homogeneous media. The one-third cross flow is the flow angle that produces the highest interpolation error for the IFAHPT and IFA methods and thus presents a worst case scenario. The third case uses the same velocity field as case two but models it on two differently sized grids that are intended to test the benefits of the particle-tracking algorithm in the IFAHPT method. The fourth case comprises a mean one-third cross flow in a randomly heterogeneous media. Parameter values for each case are listed in Table 5. The setup and results of each case are presented separately below.

4.1. Case 1: Transport in a Uniform Cross Flow

Depending on the solution method, solution accuracy can be highly dependent on the flow direction in relation to the grid alignment. Finite difference methods, for example, provide an almost exact solution for flow that is perfectly aligned with the grid. Unfortunately, this is usually not the case, and thus the versatility of a solution method stems from its ability to handle flow from varying directions. One of the advantages of the IFAHPT (and the IFA) method is that it provides an extra point, besides those that occur when the grid is perfectly aligned with the flow, where an exact solution exists. This is due to the use of the upstream concentration information and occurs when the position of \( p'' \) coincides with that of \( p' \) on the local element boundary (Figure 1).

To illustrate this, a 200-m \( \times \) 325-m modeling domain with a constant source concentration in the lower left-hand corner is used to simulate transport at varying angles from 0°, which is perfectly aligned with the x axis, to 45° (Table 5), which is the least aligned possibility (assuming a uniform grid). All simulations assume pure advective flow. Results are presented by plotting the 95% dimensionless concentration \((C/C_o)\) contour for each solution method at each flow angle on a single plot. In addition, the relative error of each method at each flow angle is plotted to provide a more quantitative method of comparison. The relative error in this case is defined as the sum of the square of the differences between the simulated plumes and the analytical...
Table 5. Parameter Values for Method Comparisons

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Case 1</th>
<th>Case 2</th>
<th>Case 3</th>
<th>Case 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Grid size, m</td>
<td>( \Delta x = \Delta y = 1 )</td>
<td>( \Delta x = \Delta y = 1 )</td>
<td>( \Delta x = \Delta y = 1, 4 )</td>
<td>( \Delta x = \Delta y = 1 )</td>
</tr>
<tr>
<td>Flow velocity, m/d</td>
<td>( u = 1.0 ) ( v = 0–1.0 )</td>
<td>( u = 1.0 ) ( v = \text{equation (25)} )</td>
<td>( u = 1.0 ) ( v = \text{equation (25)} )</td>
<td>( u = 1.0 ) ( v = 0.333 )</td>
</tr>
<tr>
<td>Grid nodes, number</td>
<td>200 ( \times ) 325</td>
<td>500 ( \times ) 200</td>
<td>500 ( \times ) 200 ( \times ) 125 ( \times ) 50</td>
<td>500 ( \times ) 200</td>
</tr>
<tr>
<td>Longitudinal dispersivity ( \alpha ), cm</td>
<td>0.0</td>
<td>0.0 and 5.0</td>
<td>5.0</td>
<td>0.0 and 5.0</td>
</tr>
<tr>
<td>Ratio of ( \alpha ) and ( \alpha_1 )</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
</tr>
<tr>
<td>Source location</td>
<td>10 m ( \leq x \leq 20 m )</td>
<td>5 m ( \leq x \leq 8 m ) Gaussian</td>
<td>16 m ( \leq x \leq 32 m )</td>
<td>16 m ( \leq x \leq 32 m )</td>
</tr>
<tr>
<td>Random field</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>( \lambda_x = 10 m ) ( \lambda_y = 5 m )</td>
</tr>
</tbody>
</table>

where \( u \) is the velocity in the \( x \) direction; \( v \) is the velocity in the \( y \) direction; \( V_{mag} \) is the scaling factor for the wave function; \( x \) is the position in the grid in the \( x \) direction; \( \text{Per} \) is the length of the period of the velocity fluctuation; and \( V_o \) is the velocity in the \( y \) direction.

4.2. Case 2: Transport in a Deterministic, Variable Cross Flow

[36] Figure 3 shows the 95% contour level plots. Each “plume” is shaded differently only to provide contrast from one to another, and thus the shading is meaningless in terms of concentration. The degree to which numerical dispersion reduces the solution accuracy at different flow angles can be clearly seen by the narrowing or vanishing of the contour. The IFAHPT method has “exact” interpolation points at flow angles of 0°, 26.6°, and 45°, when \( \text{Per} \) is equal to 0 or 1 (when \( p' \) coincides with \( p'' \) or a nodal point). The greatest error occurs at flow angles of 18.4° and 33.7°, when \( p'' \) is exactly halfway between \( p' \) and the other interpolation end point (\( \text{Per} = 0.5 \)). It should be noted that these angles are valid only when \( \Delta x = \Delta y \) and will change depending on the relative sizing of the grid. In contrast, the FA method shows the most error at flow angles close to 26.6°, as evidenced by the pinching-out of the contours. The FD-UW method, on the other hand, has the greatest interpolation error when the flow angle is 45°.

[37] Figure 4 shows the relative error of the various solution methods as the square of the differences between the analytical solution and each solution method along a cross section of the plume 175 cell widths from the source. Plotting the error as a function of the flow angle shows quantitatively the improvements made by the addition of both the improved boundary estimation and the Hermite interpolation routine. The additional exact interpolation point for the IFAHPT and the IFA methods is well demonstrated here, showing an error of 0 when \( p' \) and \( p'' \) coincide at an angle of 26.6°. Throughout the entire range of flow angles, the IFAHPT method is the most accurate.

Figure 3. Ninety-five percent contour of dimensionless concentration (\( C/C_o \)) for flow angles of 0°–45° for the IFAHPT method, the FA method, and the FD-UW method (case 1). The shading of each plume is only to provide contrast from one to another and does not indicate relative concentration.
the mean \( y \) velocity. This velocity field is divergence-free in that it does not violate continuity. Parameter values for Case 2 are \( V_{\text{mag}} = 1 \text{ m/d} \), \( \text{Per} = 20 \text{ m} \), and \( V_0 = 0.333 \text{ m/d} \).

A uniform grid spacing of 1 m in both directions is used, resulting in a computational grid with 500 nodes in the \( x \) direction and 200 nodes in the \( y \) direction. The Peclet numbers, velocities, and grid spacing correspond to dispersivities of 0 and 5 cm, respectively, the latter value being a typical value for capturing pore-scale dispersion in groundwater transport problems [Gelhar and Axness, 1983]. All parameters are shown in Table 5. Results are presented as two-dimensional contour plots as well as plume cross sections along the \( y \) axis, 430 cell widths from the source.

[39] Figure 5 shows the predicted two-dimensional plumes in pure advective flow. The IFAHPT method shows very little numerical dispersion, maintaining a concentration along the centerline that is 99.5% of the source concentration (0.995 dimensionless concentration, \( C/C_0 \)). The IFA method shows some numerical dispersion with a centerline dimensionless concentration of 0.664 at the cross-section point. The improvement of the IFAHPT method over the IFA method is mainly due to the Hermite interpolation in the IFAHPT method versus the linear interpolation of the IFA method. In addition, for the IFA method, the centerline is offset by one node. As will be discussed below, this is an artifact of the straight-line velocity estimation of the IFA method. The FA method shows substantial numerical dispersion, with a centerline dimensionless concentration of 0.499. Not surprisingly, the FD-UW method shows the most numerical dispersion, with a centerline dimensionless concentration of just over 0.228.

[40] Figure 6 shows the concentration profile of each solution as well as the analytical solution along column 438 (see Figure 5 for cross-section location). The IFAHPT scheme is able to capture the shape of the analytical solution almost exactly, showing virtually no numerical dispersion. The numerical dispersion of the other schemes is clearly evident here.

[41] Figure 7 shows plume contours with a Peclet number of 20, and Figure 8 shows the concentration profiles along column 438 (see Figure 5 for cross-section location). This is for comparative purposes between each solution method since no analytical solution is readily available to judge the absolute accuracy. Comparing the profiles of the IFA, FA, and FD-UW methods in Figure 8, it is evident that they are...
virtually unchanged from those in Figure 6, indicating numerical dispersion is dominating the solution.

### 4.3. Case 3: Deterministic, Variable Cross Flow at Varying Grid Size

As discussed above, the use of the finite difference approximation of the derivative terms reduces the advection-diffusion equation to a hyperbolic equation, the solution of which requires knowledge of the concentration at the element boundary where the solution characteristic crosses the boundary. Li and Wei [1998] use a straight-line estimation, based on the velocity at the central node, for estimating the shape of this characteristic. As the flow field becomes more complicated, straight-line estimation of the characteristic becomes less accurate. The addition of particle tracking in this work allows for a better estimation of the shape of the solution characteristics by accounting for velocity variations across each local element. The improvement due to particle tracking is illustrated here by testing the IFAHPT method, with and without particle tracking, in the sinusoidal velocity field described in case 2, on two 500-m × 200-m grids, one with a uniform grid spacing of 1 m and the other with a uniform grid spacing of 4 m. As the grid size is increased, the straight-line characteristic assumption becomes invalid and the effect of this inaccuracy can be seen.

Figure 9 shows the 95% dimensionless concentration contours from each solution method (with and without particle tracking) on each grid size. The fine-grid case shows very little difference between the two approaches since the straight-line estimation is a valid assumption at this scale. For the larger grid, however, the straight-line assumption can no longer capture the sinusoidal velocity field closely enough to reproduce the correct result (which we designate as the results from the fine-grid simulation), and the prediction degrades in two ways: increased numerical dispersion and loss of plume trajectory. This is the reason the case 2 IFA example was offset by one node (Figure 6). The large grid prediction with particle tracking does show some numerical dispersion over the fine-grid simulations, but this error is minor compared with the straight-line large-grid prediction. These effects can easily be seen in the plume vertical cross sections (Figure 10). The net improvement of particle tracking is the ability to model more complicated flow fields on a larger grid.

### 4.4. Case 4: Transport in Heterogeneous Media

Case 4 applies the IFAHPT method to a heterogeneous flow field to test the ability of the method to handle...
complicated flow fields efficiently, robustly, and accurately. The hydraulic conductivity is represented as a spatially correlated random field (Figure 11) characterized by the mean (mean LnK), variance (\(\sigma^2_{\text{LnK}}\)), and correlation scales (\(\lambda_x\) and \(\lambda_y\)) of the log conductivity. Boundary conditions for the flow model are set as constant head boundaries to provide a mean \(x\) velocity of 1 m/d and a mean \(y\) velocity of 0.333 m/d. A very fine grid simulation using \(\Delta x = \Delta y = 0.25\) m (one fourth of the base example) is used for comparisons with the exact solution. Simulations are performed with Peclet equal to infinite and 20. Results are presented as two-dimensional concentration contour plots and plume cross sections, 430 m in the \(x\) direction from the center of the plume source. The parameters defining this test example are given in Table 5.

Figure 12 shows the two-dimensional plumes for the pure advective case. Both the IFA and FA methods suffer from numerical dispersion, as does the FD-UW method. The IFAHPT method shows very little numerical dispersion when compared with the exact solution; however, it does show dimensionless concentrations greater than 1 and less than 0. This is due to “overshoot/undershoot,” which is a

Figure 8. Cross sections along column 438 for plumes shown in Figure 7.

Figure 9. The 95% dimensionless concentration (\(C/C_0\)) contour for the predicted plume distribution in a one-third, sinusoidal cross flow at different grid sizes (case 3).
well-documented artifact of the Hermite interpolation scheme [Holly and Preissmann, 1977] and is unavoidable without the inclusion of additional algorithms [e.g., Leonard, 1988]. Note how overshoot/undershoot is also evident in the small-grid simulation, indicating it is a function of the derivative values at the concentration fronts and cannot be controlled by the grid size. This overshoot/undershoot is not much of a practical restriction since even a small amount of modeled dispersion sufficiently softens the concentration front enough that the Hermite interpolation scheme becomes quite accurate. This is demonstrated below. Even with the overshoot/undershoot anomaly, the shape and trajectory of the plume are much better than the other three methods. If the IFAHPT plume is “clipped” such that no values of $C/C_o$ exceed 1 or are less than 0 (plume not shown), the solution is almost an exact fit to the small-grid “exact” solution. Our experience across a wide range of velocity fields shows clipping of the plume to be accurate in relation to analytical or fine-grid solutions. Generally, the amount of overshoot is close to the amount of undershoot, and thus mass is largely conserved with this technique.

While clipping the concentration values is admittedly not rigorous in its mathematical approach, it does provide a good fit and gives an indication of the accuracy of the general finite analytic approach described here. The cross sections of each plume are shown in Figure 13 and reveal the degree of numerical dispersion present in each method. The IFAHPT method shows very good agreement with the “exact” solution. For comparison, the clipped solution cross section is also shown in Figure 13.

Once a small amount of dispersion is added to the simulation, the overshoot/undershoot problem disappears (Figure 14 and Figure 15). The IFAHPT method shows a centerline dimensionless concentration that is higher than the fine-grid simulation. This is due to overshoot that occurs early on in the plume path where the concentration front is very sharp and the effect is propagated downstream. The plume for the FD-UW method is virtually unchanged from the pure advective case, while the IFA and FA methods show only minor changes from the added dispersion. This indicates that for Peclet numbers greater than 20, the IFA, FA, and FD-UW methods are unsuitable since the degree of

![Figure 10](image10.png)

**Figure 10.** Cross sections along column 470 for plumes shown in Figure 9.

![Figure 11](image11.png)

**Figure 11.** Random field for use in case 4 simulations.
numerical dispersion is greater than or equal to the modeled dispersion. This can easily be seen by comparing the plume cross sections of the pure advective case (Figure 13) to the Peclet = 20 case (Figure 15).

5. Drawbacks and Limitations

[47] The method presented here assumes transport is two-dimensional and steady state and as such is limited in its applicability. That being said, there are still many applications for which these assumptions are plausible and valid. Mathematically, it is possible to solve transient cases using finite analytic methods; however, the complexity of the solution makes the direct approach impractical. There are other ways, however, of addressing this issue, as is discussed below. The extension to three dimensions is relatively straightforward and is also discussed below.

[48] Regarding implementation, the main drawback of this method is the overshoot/undershoot that occurs in the presence of sharp concentration fronts. This is due to the inclusion of the concentration derivatives in the Hermite interpolation scheme that have very large values at these points. For conditions with even a slight amount of dispersion, this drawback quickly disappears since the derivative values become more manageable. Our experience here shows that for sharp-edged sources with Peclet numbers less than about 120, the overshoot/undershoot conditions is of minimal concern. For dispersed initial sources, excellent results are obtained without overshoot/undershoot, even for infinite Peclet numbers.

6. Computational Efficiency

[49] For a modeling domain with an equivalent number of nodes, the IFAHPT method is computationally more expensive than more traditional methods such as finite difference or finite element methods. This stems from the need to solve three equations (the concentration equation and the two derivative equations) rather than one. However, when one compares the results on the basis of equivalent accuracy, the
IFAHPT method is much less expensive. To achieve the same accuracy, traditional methods would require a grid refinement whose computational cost would far outweigh the cost of evaluating the derivative terms in the IFAHPT method. However, even considering the additional computational overhead, the simulation run times for the examples in this paper are relatively small, of the order of 15–20 s for simulations converging to an accuracy of $10^{-5}$ on a Pentium III 650-MHz PC with 128 Mb of RAM. This includes the time to run the particle-tracking algorithm, which in reality adds very little to the total run time since it needs to be run only once per simulation rather than per iteration.

7. Conclusions and Future Considerations

In this paper we have developed an improved finite analytic method for solving transport problems in advection-dominated, variable flow fields. The work builds on that of Li and Wei [1998] and traditional FA methods [Chen and Chen, 1984] by adding an improved particle-tracking scheme to account for velocity variations and to better determine the shape of the solution characteristic in each local element, as well as a higher-order interpolation scheme (Hermite interpolation) to better refine the local boundary conditions. We have shown that these additions greatly reduce numerical dispersion and perform well at a wide range of Peclet numbers and varying flow fields. These additions, coupled with the efficiency of the Li and Wei method, provide a solution technique that is easy to implement and accurate in its application.

While we feel the work here is significant in producing a steady state method that is both robust and accurate, as well as applicable to many different flow and transport problems, we see the need for extension of this method to transient cases. Many problems involving the solution of the advection-diffusion equation are by nature time-dependent, and thus a future paper [T. Lowry and S.-G. Li, submitted manuscript, 2001] presents a transient version of the IFAHPT method by utilizing the space-time accurate...
technique of Li et al. [1992]. By applying this technique, we are able to use finite analytic methods in transient problems. To accomplish this, the transient advection-diffusion equation is transformed from “real” space using a Laplace transform, effectively removing the time derivative and producing a steady state equation (in form) in complex Laplace space. The resulting equation is then solved using the IFAHPT method with the complex valued solution being returned to real space with an efficient Laplace inverse routine [DeHoog et al., 1982]. This approach is beneficial for two reasons. First, since the method is steady state in Laplace space, it is extremely efficient. The solution need only be calculated and inverted at a single, future time, with no time-stepping involved. This means a 20-day simulation takes approximately the same computational time as a 2000-day simulation. Second, a sharp concentration gradient in real space is represented as a diminishing wave function in Laplace space, meaning the gradients at these points are smooth. This removes the overshoot undershoot problem associated with the Hermite interpolation scheme and is the justification for applying Hermite interpolation to this work.

[52] Finally, the method outlined in this work lends itself to three-dimensional transport as well (both steady state and transient). While the application to three dimensions is similar to the two-dimensional case and relatively straightforward, the logic in the programing and implementation is much more difficult, stemming mainly from the high number of flow combinations that must be accounted for to include the improved boundary condition of Li and Wei [1998]. Particle tracking must be applied in three dimensions, while the interpolation used to determine the concentration of the boundary point position must be performed in two dimensions. Utilizing Hermite interpolation requires simultaneously solving seven equations (the concentration, the \( x \), \( y \), and \( z \) derivatives, and three cross derivatives) instead of just three equations for the two-dimensional case. This may prove to be computationally expensive for the steady state case, but combined with the Laplace transform approach described above for the transient case, it is expected the three-dimensional transient method will have a significant computational advantage. Work on this problem is already underway, with significant progress to date.

[53] Acknowledgments. The research described in this paper is partially sponsored by the National Science Foundation under grants BES-9811895 and EEC-0088137 and by the New Zealand Foundation for Research, Science, and Technology under contract LVLX00006. We would like to thank the manuscript reviewers and editors of WRR, whose comments added greatly to the final version of this paper.

References