INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids **28**: 461–476 (1998)

AN OPERATOR SPLITTING ALGORITHM FOR THE THREE-DIMENSIONAL ADVECTION–DIFFUSION EQUATION

LIAQAT ALI KHAN* AND PHILIP L.-F. LIU

School of Civil and Environmental Engineering, Cornell University, Ithaca, New York 14853, USA

SUMMARY

Operator splitting algorithms are frequently used for solving the advection–diffusion equation, especially to deal with advection dominated transport problems. In this paper an operator splitting algorithm for the three-dimensional advection–diffusion equation is presented. The algorithm represents a second-order-accurate adaptation of the Holly and Preissmann scheme for three-dimensional problems. The governing equation is split into an advection equation and a diffusion equation, and they are solved by a backward method of characteristics and a finite element method, respectively. The Hermite interpolation function is used for interpolation of concentration in the advection step. The spatial gradients of concentration in the Hermite interpolation are obtained by solving equations for concentration gradients in the advection step. To make the composite algorithm efficient, only three equations for first-order concentration derivatives are solved in the diffusion step of computation. The higher-order spatial concentration gradients, necessary to advance the solution in a computational cycle, are obtained by numerical differentiations based on the available information. The simulation characteristics and accuracy of the proposed algorithm are demonstrated by several advection dominated transport problems.

KEY WORDS: advection–diffusion equation; operator splitting algorithm; Holly and Preissmann scheme; method of characteristics; finite element method

1. INTRODUCTION

The three-dimensional advection–diffusion equation for a conservative concentration field $c(t, \mathbf{x})$ can be written as

$$\partial_t c = Lc, \tag{1}$$

where L is the sum of two differential operators

$$L = L_1 + L_2, \tag{2}$$

such that

$$L_1 = -\mathbf{u}\cdot\nabla, \qquad L_2 = \nabla\cdot(\mathbf{D}\cdot\nabla), \tag{3}$$

where vector $\mathbf{u} = (u, v, w)$ is the flow velocity, $\mathbf{D}$ is a second-order tensor of diffusion coefficients, $\mathbf{x} = (x, y, z)$ is the spatial co-ordinate, and t is time. With proper initial and boundary conditions, Equation (1) constitutes a well-posed problem.

* Correspondence to: HydroQual Inc., 1 Lethbridge Plaza, Mahwah, N.J. 07430, USA.

CCC 0271–2091/98/130461–16$17.50

Received January 1997
Revised September 1997

The advection–diffusion Equation (1) is a parabolic equation. However, for advection-dominated problems it behaves like a hyperbolic equation. A numerical procedure appropriate for parabolic equations is not generally suitable for hyperbolic equations and vice-versa. For such problems, operator splitting allows the decomposition of Equation (1) into an advection equation and a diffusion equation, and the use of appropriate numerical procedures for solving the split equations. The concept of operator splitting has been discussed in literature for more than two decades [1,2], and they have been frequently used for solving engineering problems. However, it is interesting to note that most of the reported operator splitting algorithms for the advection–diffusion equation and the Navier–Stokes equations are based on first-order accurate splitting [3].

The Holly and Preissmann [4] method has been the basis of many operator splitting algorithms for the advection–diffusion equation. The split advection equation is solved by a backward method of characteristics and the diffusion equation by conventional finite difference or finite element methods. During the advection step, a C^1 continuous Hermite interpolation function is used for spatial interpolation of concentration. The spatial derivatives of concentration in the Hermite interpolation function are obtained by solving equations governing their advection and diffusion. The desirable features of solution procedure, small numerical dispersion and dissipation, are obtained at a price, i.e. solving additional equations governing the advection and diffusion of concentration gradients.

The basic concept of the Holly and Preissmann scheme has been adopted by Yang and Hsu [5] and Yang *et al.* [6] for the one-dimensional advection–diffusion equation and by Ding and Liu [7], Glass and Rodi [8], Holly and Usseglio-Polatera [9], and Yang *et al.* [10] for two-dimensional problems. These operator splitting algorithms, including Holly and Preissmann [4], are first-order-accurate in time due to splitting errors. Therefore, Khan and Liu [11] utilized Strang [12] type splitting to make the composite algorithm for a system of one-dimensional advection–diffusion–reaction equations second-order-accurate in time.

A direct application of the Holly and Preissmann scheme to multidimensional advection–diffusion equation results in computationally expensive algorithms. It is necessary to solve $2^d - 1$, where d is the dimension of the problem, additional equations for spatial concentration gradients. To make the algorithm computationally less 'complicated and expensive' for two-dimensional problems, Komatsu *et al.* [13,14] have proposed the use of modified Lagrangian type interpolation functions. The suggested interpolation functions contain 'optimal' weighting coefficients which are selected to reduce numerical dispersion and dissipation. An alternative to Komatsu *et al.* [13,14] is to use the Hermitian interpolation function with estimated derivatives (based on numerical differentiation of concentration) in the advection step. Such an approach, with shape preserving constraints to preserve convexity or concavity of solution, has been analyzed by Rasch and Williamson [15] for the one-dimensional advection equation, and applied to two-dimensional problems by Williamson and Rasch [16].

The solution procedures by Komatsu *et al.* [13,14] and Rasch and Williamson [15] avoid the need to solve concentration gradient equations in both the advection and diffusion steps of computation. However, if the proposed eight-point interpolation function of Komatsu *et al.* [13,14] is used for three-dimensional problems, then it will relate 512 surrounding nodes. The resulting computational molecule is complex, incorporating information from outside the domain of dependence of the solution. Moreover, expanding the computational molecule is counter to the basic motivation of the Holly and Preissmann concept.

The numerical experiments carried out by Rasch and Williamson [15] indicate that the best method of estimating concentration derivatives is problem-dependent. A selected method may not result in simultaneous minimization of numerical dissipation and dispersion. The number

of nodes involved in estimating concentration gradients and enforcing shape preserving constraints is essentially the same as in Komatsu *et al.* [13,14]. Application of shape preserving constraints to multidimensional problems is complicated and requires a series of adjustments of the computed concentration and concentration gradients [16]. Such adjustments are likely to result in mass conservation errors.

Baptista *et al.* [17] have analyzed accuracies of the backward method of characteristics for the one-dimensional advection equation using different interpolation functions. The numerical experiments indicate that the performance of the Hermite interpolation function with estimated derivatives is better than Lagrangian type interpolation functions. In addition, the Holly and Preissmann method is superior to procedures based on Hermite interpolation with estimated derivatives. Consequently, developing operator splitting algorithms for the three-dimensional advection–diffusion equation by retaining the basic features of the Holly and Preissmann concept appears attractive.

In this paper, an operator splitting algorithm for the three-dimensional advection–diffusion equation is presented. It represents a second-order-accurate adaptation of the Holly and Preissmann scheme and an alternative method of making the algorithm computationally efficient for multidimensional problems. Once the characteristic path has been determined for solving the advection equation for concentration, additional computational time for solving equations for concentration gradients is negligible [9,13,14]. The Holly and Preissmann scheme is computationally expensive, especially for multidimensional problems, because it is necessary to solve additional diffusion equations for spatial concentration gradients. Therefore, the solution procedure proposed in this paper solves all the equations advecting concentration gradients. In the diffusion step, only three equations for first-order concentration derivatives are solved. The higher-order gradients are obtained by numerical differentiations based on the available information.

2. OPERATOR SPLITTING ALGORITHM

The basic theory of operator splitting algorithms was developed by Yanenko [1], Marchuk [2] and Strang [12] in the context of locally one-dimensional methods. LeVeque and Oliger [18] have analyzed splitting algorithms for a system of one-dimensional hyperbolic equations, identifying different sources of errors in solving the equations. A similar analysis for a system of coupled one-dimensional advection–diffusion–reaction equations has been presented by Khan and Liu [11]. The analysis is applicable to Equation (1) (with proper definition of the operators L, L_1 and L_2 and setting reaction operator to zero in Khan and Liu). A second-order-accurate Strang type splitting algorithm [3,11,18] for Equation (1) can be obtained by solving the following split equations

$$\partial_t c^1 = L_1 c^1, \qquad c^1(t_n, \mathbf{x}) = c(t_n, \mathbf{x}), \quad t \in [t_n, t_{n+1/2}], \tag{4}$$

$$\partial_t c^2 = L_2 c^2, \qquad c^2(t_n, \mathbf{x}) = c^1(t_{n+1/2}, \mathbf{x}), \quad t \in [t_n, t_{n+1}], \tag{5}$$

$$\partial_t c^1 = L_1 c^1, \qquad c^1(t_{n+1/2}, \mathbf{x}) = c^2(t_{n+1}, \mathbf{x}), \quad t \in [t_{n+1/2}, t_{n+1}], \tag{6}$$

and taking $c(t_{n+1}, \mathbf{x}) = c^1(t_{n+1}, \mathbf{x})$. In the above equations, $c^1(t, \mathbf{x})$ and $c^2(t, \mathbf{x})$ are dependent variables of the split advection and diffusion equations respectively, n is the number of time steps, $t_n = nk$, and k is the time step. Figure 1 shows the sequence of solving split equations in a computational cycle. For a Strang type operator splitting algorithm, the composite algorithm will be second-order-accurate [11,18] provided the numerical procedures for the split equations are at least second-order-accurate.

2.1. Advection step of computation

The advection Equations (4) or (6) can be expressed as

$$\frac{\mathrm{D}c^1}{\mathrm{D}t} = 0, \tag{7}$$

where $\mathrm{D}/\mathrm{D}t = \partial_t + \mathbf{u}\cdot\nabla$. Equation (7) indicates that $c^1(t, \mathbf{x})$ is invariant along characteristic

$$\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t} = \mathbf{u}. \tag{8}$$

The solution of Equation (7) along the characteristic is

$$c^1(t_{n+1}, \mathbf{x}) = c^1(t_n, \mathbf{x} - k\mathbf{u}). \tag{9}$$

In this study, the three-dimensional computational domain is discretized by hexahedral elements, Equation (8) is integrated backward in time by an explicit second-order-accurate Runge–Kutta method, and $c^1(t_n, \mathbf{x} - k\mathbf{u})$ is determined by a C^0 continuous serendipity Hermitian interpolation function.

Let (η, ξ, ζ) be the local co-ordinates corresponding to global co-ordinates (x, y, z), such that $-1 \le (\eta, \xi, \zeta) \le 1$. Then a C^1 continuous Hermitian interpolation function for three-dimensional problems can be obtained by taking the tensor product of one-dimensional Hermitian interpolation functions in the co-ordinate directions [19]. The following serendipity Hermitian interpolation function for $c^1(t_{n+1}, \eta, \xi, \zeta)$ can be obtained

$$c^1(t_{n+1}, \eta, \xi, \zeta) = \sum_{i=1}^{8} (M_i^{000} + M_i^{100}\,\partial_\eta + M_i^{010}\,\partial_\xi + M_i^{001}\,\partial_\zeta + M_i^{110}\,\partial_{\eta\xi} + M_i^{011}\,\partial_{\xi\zeta} + M_i^{101}\,\partial_{\zeta\eta})c_i^1, \tag{10}$$

by neglecting $\partial_{\eta,\xi,\zeta} c^1(t_n, \eta, \xi, \zeta)$ in the C^1 Hermite interpolation function. In Equation (10)

$$\begin{aligned}
M_i^{000} &= H^0(\eta, \eta_i)H^0(\xi, \xi_i)H^0(\zeta, \zeta_i), & M_i^{100} &= H^1(\eta, \eta_i)H^0(\xi, \xi_i)H^0(\zeta, \zeta_i),\\
M_i^{010} &= H^0(\eta, \eta_i)H^1(\xi, \xi_i)H^0(\zeta, \zeta_i), & M_i^{001} &= H^0(\eta, \eta_i)H^0(\xi, \xi_i)H^1(\zeta, \zeta_i),\\
M_i^{110} &= H^1(\eta, \eta_i)H^1(\xi, \xi_i)H^0(\zeta, \zeta_i), & M_i^{011} &= H^0(\eta, \eta_i)H^1(\xi, \xi_i)H^1(\zeta, \zeta_i),\\
M_i^{101} &= H^1(\eta, \eta_i)H^0(\xi, \xi_i)H^1(\zeta, \zeta_i),
\end{aligned} \tag{11}$$

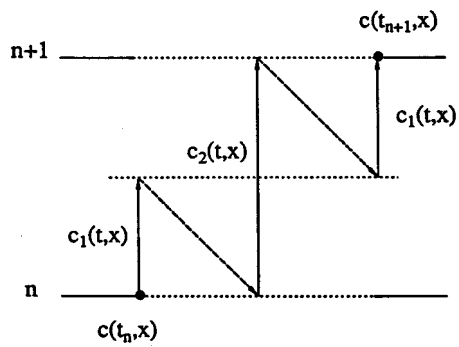

Figure 1. Schematic representation of the second-order-accurate Strang's operator splitting algorithm.

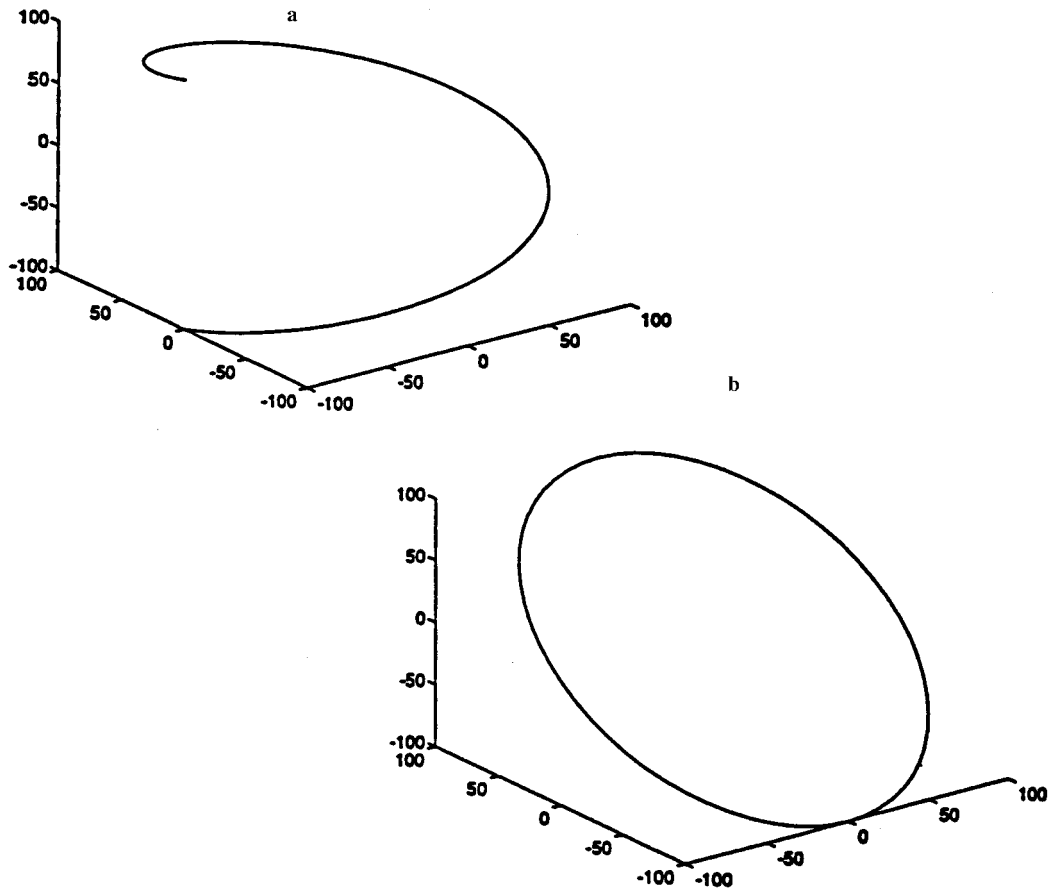

Figure 2. Trajectories of fluid particles in examples 1 and 2.

where

$$H^0(\eta, \eta_i) = -\frac{1}{4}(\eta + \eta_i)^2(\eta + 2\eta_i), \qquad H^1(\eta, \eta_i) = +\frac{1}{4}(\eta + \eta_i)^2(\eta - \eta_i). \tag{12}$$

The expressions for $H^0(\xi, \xi_i)$, $H^1(\xi, \xi_i)$ and $H^0(\zeta, \zeta_i)$, $H^1(\zeta, \zeta_i)$ are obtained by replacing η with ξ and ζ, respectively.

Equation (10) indicates that $c^1(t_{n+1}, \eta, \xi, \zeta)$ is a function of $c^1(t_n, \eta, \xi, \zeta)$, $c^1_\eta(t_n, \eta, \xi, \zeta)$, $c^1_\xi(t_n, \eta, \xi, \zeta)$, $c^1_\zeta(t_n, \eta, \xi, \zeta)$, $c^1_{\eta,\xi}(t_n, \eta, \xi, \zeta)$, $c^1_{\xi,\zeta}(t_n, \eta, \xi, \zeta)$ and $c^1_{\zeta,\eta}(t_n, \eta, \xi, \zeta)$. This information is obtained by solving following first-order

$$\frac{\mathrm{D}c^1_x}{\mathrm{D}t} = -\mathbf{u}_x \cdot \nabla c^1, \qquad \frac{\mathrm{D}c^1_y}{\mathrm{D}t} = -\mathbf{u}_y \cdot \nabla c^1, \qquad \frac{\mathrm{D}c^1_z}{\mathrm{D}t} = -\mathbf{u}_z \cdot \nabla c^1, \tag{13}$$

and second-order

$$\frac{\mathrm{D}c^1_{xy}}{\mathrm{D}t} = -\mathbf{u}_x \cdot \nabla c^1_y - \mathbf{u}_y \cdot \nabla c^1_x - \mathbf{u}_{xy} \cdot \nabla c^1, \qquad \frac{\mathrm{D}c^1_{yz}}{\mathrm{D}t} = -\mathbf{u}_y \cdot \nabla c^1_z - \mathbf{u}_z \cdot \nabla c^1_y - \mathbf{u}_{yz} \cdot \nabla c^1,$$

$$\frac{\mathrm{D}c^1_{zx}}{\mathrm{D}t} = -\mathbf{u}_z \cdot \nabla c^1_x - \mathbf{u}_x \cdot \nabla c^1_z - \mathbf{u}_{zx} \cdot \nabla c^1, \tag{14}$$

concentration gradient equations in global co-ordinates. In the above equations, $c_x^1 = (t, \mathbf{x}) = \partial_x c^1(t, \mathbf{x})$, and $c_y^1(t, \mathbf{x})$, $c_z^1(t, \mathbf{x})$, $c_{xy}^1(t, \mathbf{x})$, $c_{yz}^1(t, \mathbf{x})$, $c_{zx}^1(t, \mathbf{x})$, $\mathbf{u}_x$, $\mathbf{u}_y$, $\mathbf{u}_z$, $\mathbf{u}_{xy}$, $\mathbf{u}_{yz}$, and $\mathbf{u}_{zx}$ are similarly defined. Equations (13) and (14) are obtained by differentiating Equation (7) with respect to spatial variables. The right-hand-sides of the above equations are known from previously solved equations, and they are solved along the characteristic defined by Equation (8). To transform concentration gradients in Equations (14) from local co-ordinates to global co-ordinates and vice-versa, it is necessary to solve equations for $c_{xx}^1(t, \mathbf{x})$, $c_{yy}^1(t, \mathbf{x})$ and $c_{zz}^1(t, \mathbf{x})$. One of the reasons for neglecting $c_{\eta,\xi,\zeta}^1(t_n, \eta, \xi, \zeta)$ in Equation (10) is that the corresponding transformations need 26 additional third-order concentration derivatives.

2.2. Diffusion step of computation

The well-behaved diffusion Equation (5) is discretized in space by a Galerkin finite element method using linear basis function. The resulting semi-discretized system of equations is approximated in time by the Crank–Nicolson finite difference method. The details of the solution procedure can be found in Lapidus and Pinder [19]. To proceed with the computations, it is necessary to determine diffusion of $c_x^2(t, \mathbf{x})$, $c_y^2(t, \mathbf{x})$, $c_z^2(t, \mathbf{x})$, $c_{xy}^2(t, \mathbf{x})$, $c_{yz}^2(t, \mathbf{x})$, and $c_{zx}^2(t, \mathbf{x})$ corresponding to Equations (13) and (14) in the advection step. In the present study, the following first-order concentration gradient equations are solved

$$\partial_t c_x^2 = L_2 c_x^2 + L_{2,x} c^2, \qquad \partial_t c_y^2 = L_2 c_y^2 + L_{2,y} c^2, \qquad \partial_t c_z^2 = L_2 c_z^2 + L_{2,z} c^2, \tag{15}$$

where $L_{2,x} = \nabla \cdot (\partial \mathbf{D}/\partial x \cdot \nabla)$ and $L_{2,y}$, $L_{2,z}$ are similarly defined. These equations are obtained by differentiating Equation (5) with respect to x, y and z respectively. The second term on the right-hand-sides of these equations are known functions of concentration. Therefore, the finite element procedure for diffusion equation, with minor changes, is used for solving Equations (15).

Once $c^2(t_{n+1}, \mathbf{x})$, $c_x^2(t_{n+1}, \mathbf{x})$, $c_y^2(t_{n+1}, \mathbf{x})$ and $c_z^2(t_{n+1}, \mathbf{x})$ are known, the second-order concentration gradients $c_{xy}^2(t_{n+1}, \mathbf{x})$, $c_{yz}^2(t_{n+1}, \mathbf{x})$ and $c_{zx}^2(t_{n+1}, \mathbf{x})$ are determined from the following truncated form of the Hermite interpolation function (10)

$$c^2(t_{n+1}, \eta, \xi, \zeta) = \sum_{i=1}^{8} (M_i^{000} + M_i^{100}\, \partial_\eta + M_i^{010}\, \partial_\xi + M_i^{001}\, \partial_\zeta) c_i^2, \tag{16}$$

where M_i^{000}, M_i^{100}, M_i^{010} and M_i^{001} are defined by Equation (11). The concentration and concentration gradients on the right-hand-side are at time level t_{n+1}, and they are known from the solutions of previous equations.

Application of the finite element method to the three-dimensional diffusion equation results in a large system of sparse simultaneous equations in each time step, requiring significant computer time in solving the diffusion equation. Each of the concentration gradient equations (15) contributes equally in increasing the computational time. A direct application of the Holly and Preissmann [4] method to the three-dimensional advection–diffusion equation, similar to that presented by Ding and Liu [7], Glass and Rodi [8], Holly and Usseglio-Polatera [9] and Yang *et al.* [10] for two-dimensional problems, would involve solving equations governing diffusion of $c_{xy}^2(t_{n+1}, \mathbf{x})$, $c_{yz}^2(t_{n+1}, \mathbf{x})$ and $c_{zx}^2(t_{n+1}, \mathbf{x})$. The proposed method of evaluating these terms results in a saving of $\approx 40\%$ of computational time in the diffusion step of computation.

The motivation for introducing numerical approximations in the diffusion step, rather than in the advection step like Rasch and Williamson [15], is that computations are less sensitive to numerical errors [11,20] in the diffusion step of computations. The solution of the diffusion equation does not have a directional bias. Therefore, numerical schemes for differentiation are not expected to be problem dependent as found by Rasch and Williamson [15]. It should be noted that because hexahedral elements are being used, it is not possible to determine all concentration derivatives by numerical differentiations of $c^2(t_{n+1}, \mathbf{x})$, and completely eliminate the need to solve Equation (15). Numerical differentiations using trilinear interpolation functions result in $c^2_{xx} = c^2_{yy} = c^2_{zz} = 0$, while c^2_{xy}, c^2_{yz} and c^2_{zx} are independent of (x, y), (y, z) and (z, x) respectively.

3. ACCURACY OF THE ALGORITHM

Errors in the operator splitting algorithm result from splitting the governing differential equation and solving the split equations by numerical procedures. These errors are known [11,18] as the splitting error and the truncation error respectively. The results of error analysis in Khan [20] for the three-dimensional advection–diffusion equation are briefly discussed in this section. The analysis is not repeated because it is similar to that presented in Khan and Liu [11], except that equations for the three-dimensional case are algebraically more complex. The previously reported operator splitting algorithms for one-dimensional [4–6] and two-dimensional [7–10] advection–diffusion equations are based on first-order-accurate splitting, i.e. solving split Equations (4) and (5) in a computational cycle. However, by solving the split equations in the sequence indicated by Equations (4)–(6), the splitting error is reduced by an order of magnitude in time. Similar analyses can be found in Demkowicz *et al.* [3], LeVeque and Oliger [18] and Khan and Liu [11,21].

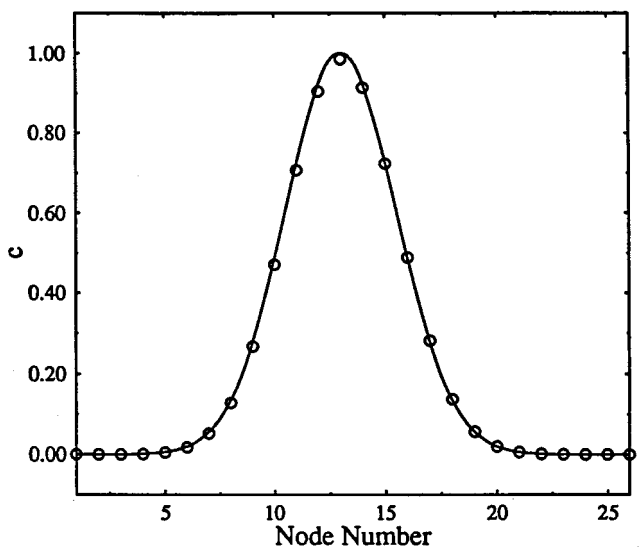

Figure 3. Comparison of numerical and analytical solutions for example 1. (—) is the analytical solution, (○) is the numerical solution.

Once the governing equation has been split as indicated by Equations (4)–(6), it is sufficient to solve the split equations by at least second-order-accurate numerical schemes to maintain second-order-accuracy of the composite algorithm [1,2,11,18]. The numerical procedure for solving the split advection equation is third-order-accurate in space [20] (see Khan and Liu [11] for similar analysis for the one-dimensional case). For constant velocity, no time discretization error is introduced in solving the advection equation [11,20]. The Crank–Nicolson finite element method for the diffusion equation is second-order-accurate in time and space [19,20]. As a result, the composite algorithm presented in this paper is second-order-accurate in time and space

For a given accuracy of the algorithm, the magnitude of error for the advection dominated problem is determined by numerical dispersion and dissipation associated with the numerical procedure for advection equation. Analysis of numerical dispersion and dissipation for the three-dimensional advection equation [20], similar to that for the one-dimensional advection equation [4,5,8,11], indicates that for a damping error of less than 0.0001 per time step, it is necessary to use a spatial resolution of about 20 elements per wave length. The corresponding resolution required for one-dimensional problems is approximately ten elements. The increased resolution necessary for three-dimensional problems is the result of higher dimensionality of the problem, and cross-wind numerical dispersion and dissipation.

4. SIMULATION CHARACTERISTICS

The operator splitting algorithm described in this paper has been tested for a large number of two- and three-dimensional transport problems in Khan [20]. In this section, four three-dimensional examples, mostly adopted from Park and Liggett [22], are presented. The velocity distribution in these examples can be expressed as [22]

$$\mathbf{u}(\mathbf{x}) = \Omega\mathbf{x} + \mathbf{u}_o, \tag{17}$$

where

$$\mathbf{u}_o = (u_o, v_o, w_o)^T \tag{18}$$

is a constant vector representing uniform flow, and

$$\Omega = \begin{bmatrix} 0 & \omega_{12} & 0 \\ \omega_{21} & 0 & \omega_{23} \\ 0 & \omega_{32} & 0 \end{bmatrix} \tag{19}$$

is a constant shear matrix. The components of shear rate of deformation due to the velocity field are

$$\varepsilon_x = (w_y + v_z)/2, \qquad \varepsilon_y = (u_z + w_x)/2, \qquad \varepsilon_z = (v_x + u_y)/2. \tag{20}$$

By appropriately selecting ω_{12}, ω_{21}, ω_{23} and ω_{32}, Equation (17) can be used to represent the velocity field with zero and non-zero shear strains. An analytical solution of the advection–diffusion equation (1) for the velocity field (17) can be found in Park and Liggett [22].

The initial condition for problems considered in this section can be expressed as

$$c(0, x, y, z) = \exp\left(-\frac{(x - x_o)^2}{2\sigma_o^2} - \frac{(y - y_o)^2}{2\sigma_o^2} - \frac{(z - z_o)^2}{2\sigma_o^2} \right), \tag{21}$$

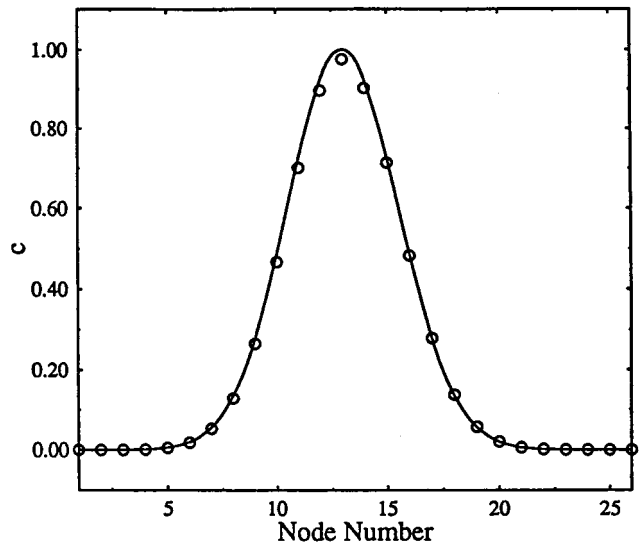

Figure 4. Comparison of numerical and analytical solution for example 2. (—) is the analytical solution, (○) is the numerical solution.

where σ_o is the initial standard deviation of the hump, and (x_o, y_o, z_o) is the position of the centroid at $t = 0$. A value of $\sigma_o = 2.5h$, where h is element size, is used so that numerical dispersion and dissipation are negligible. For the selected value of σ_o, $c(0, \mathbf{x})$ falls from 1.0 to 0.001 over nine elements and the Gaussian hump can be considered to be distributed over approximately 18 elements in each direction.

The examples presented in this section share a common computational domain, $x_i \in [-220, 220]$, $i = 1$, 2 and 3, discretized by $44 \times 44 \times 44$ cubic elements each of size 10^3. The size of domain has been selected so that the Gaussian hump (21) can undergo considerable rotation and translation without being significantly affected by boundary conditions, i.e. $c(t, \mathbf{x}_o) = c^1(t, \mathbf{x}_o) = c^2(t, \mathbf{x}_o) = 0$ for $\mathbf{x}_o \in \Gamma$, where Γ is boundary of the domain. It is necessary to set the concentration to zero on the boundary and using large domains to compare numerical solutions with analytical solutions derived for infinite domain.

4.1. Advection problems

4.1.1. Example 1. The Gaussian hump (21) is placed in the domain with $\mathbf{x}_o = (x_o, y_o, z_o) = (-100, 0, -100)$, and the velocity distribution is given by Equation (17) for

$$-\omega_{12} = \omega_{21} = \frac{3}{2r_0}, \qquad \omega_{23} = \omega_{32} = 0, \qquad (u_o, v_o, w_o) = (0, 0, 0.2170), \tag{22}$$

where $r_o = 220$. As a result, the initial condition undergoes solid body rotation in the x–y plane and a linear translation in the z-direction. A time step $k = 15.3589$ is used so that the Gaussian hump undergoes a complete rotation in 60 time steps. The trajectory of $\mathbf{x}_o$ during the computations is shown in Figure 2(a). As $\varepsilon_x = \varepsilon_y = \varepsilon_z = 0$, the initial condition should remain unchanged in shape over the simulation period. Figure 3 compares numerical and analytical solutions. The maximum Courant number, $v_{\max}$ is 3.2745 in the computations. The computed peak concentration is 0.9850, while the maximum negative concentration is -0.0001. Therefore, the error in the computed peak concentration is only 1.50%.

4.1.2. Example 2. The second example represents a simple change in the velocity field used in the previous example. However, this modification makes the problem considerably more difficult. The new velocity field is given by Equation (17) with

$$-\omega_{12} = \omega_{21} = \omega_{32} = \frac{3}{2r_o}, \qquad \omega_{23} = 0, \qquad \mathbf{u}_o = 0. \tag{23}$$

The Gaussian hump is placed at $(x_o, y_o, z_o) = (-100, 0, 0)$. The trajectory of $\mathbf{x}_o$ is shown in Figure 2(b). For a time step $k = 15.3589$, the maximum Courant number $v_{\max} = 3.9903$.

In this example, $\varepsilon_x = 3/4r_o$ and $\varepsilon_y = \varepsilon_z = 0$. Therefore, the initial condition undergoes rotation and translation as well as deformation. The velocity distribution is such that deformation in

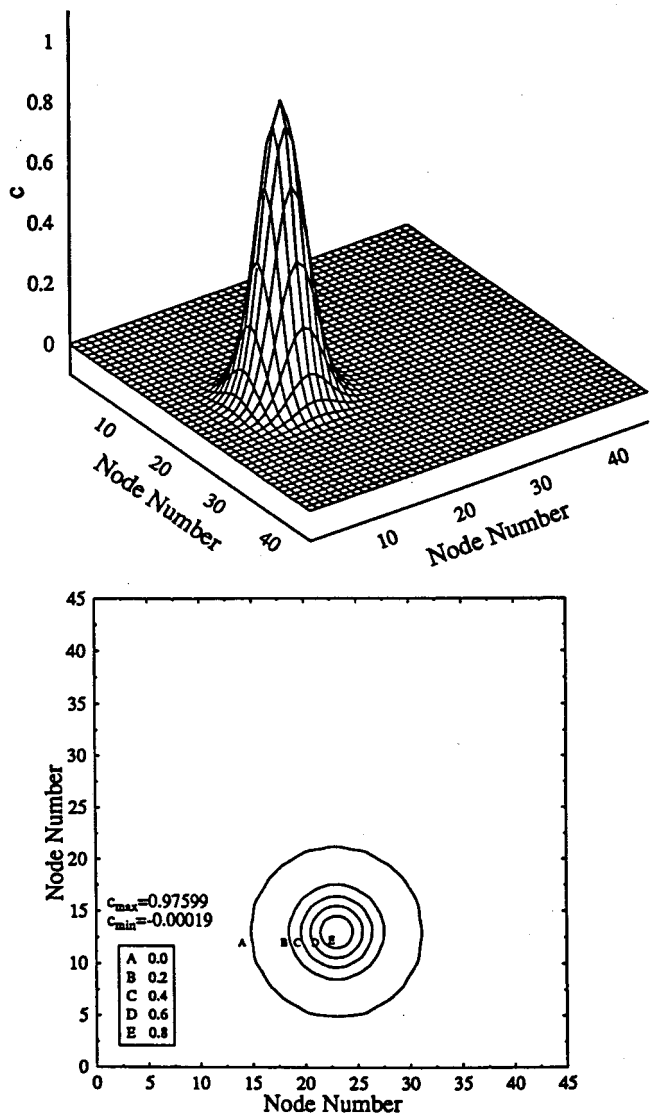

Figure 5. Three-dimensional and contour plots of concentration for advection example 2.

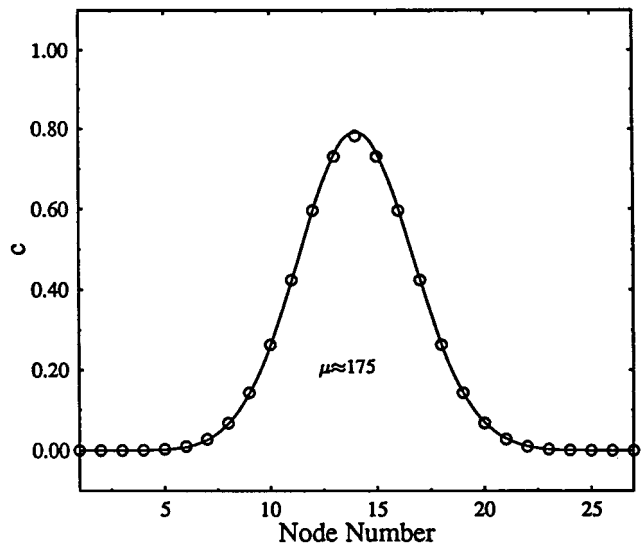

Figure 6. Comparison of numerical and analytical solution for example 3. (—) is the analytical solution, (○) is the numerical solution.

the region $x \leq 0$ is balanced by deformation in the region $x \geq 0$. Consequently, at the end of a complete rotation, the Gaussian hump should return to its initial shape. Figure 4 compares the computed profile after 60 time steps with its initial shape. The corresponding three-dimensional and contour plots of the hump in the $z = 0$ plane are shown in Figure 5. These figures indicate that the computation has maintained the initial profile quite well. Error in the computed peak concentration is 2.40%.

4.2. Advection dominated problems

4.2.1. Example 3. The first advection dominated transport problem considers example 1 with diffusion coefficients $D_{xx} = D_{yy} = D_{zz} = 0.06$. For the computational parameters defined in example l, the Peclet number μ varies from 0 to 250 in the computational domain. Along the trajectory of $\mathbf{x}_0$, $\mu \approx 175$. Figure 6 compares analytical and numerical solutions. Error in the computed peak concentration is only 1.13%. Compared with example l, the smaller error is mainly due to the increasing size of the hump and smaller spatial concentration gradient as computations advance in time.

4.2.2. Example 4. This example considers a two-dimensional flow field with a stagnation point. However, the physical problem is three-dimensional because of the initial condition and non-zero diffusion in all the co-ordinate directions. The flow field is specified by

$$\omega_{12} = \omega_{21} = \rho, \qquad \omega_{23} = \omega_{32} = \gamma, \qquad \mathbf{u}_o = \rho\alpha(2, -2, 0)^T, \tag{24}$$

where $\rho = 0.0006$, $\gamma = 10^{-20}$ and $\alpha = 110$. A non-zero γ is necessary to allow for the analytical solution presented by Park and Liggett [22]. The value of α has been selected to have a stagnation point at $(220, -220)$ in the x–y plane. Figure 7 shows the velocity distribution in the x–y plane.

In this example, ray method described by Park and Liggett [22] is used to obtain an appropriate initial condition by advecting and diffusing a point source until the cloud has

spread sufficiently. The centroid of the resulting hump is located at $(-100, 100, 0)$ in the computational domain. The concentration distribution is normalized by the maximum analytical concentration. For $D_{xx} = D_{yy} = D_{zz} = 0.06$, μ along the trajectory of peak concentration, the line joining $(-220, 220)$ and $(220, -220)$ in Figure 7, varies from 62.25 to 0. The initial condition is then advected and diffused by the numerical model from $(-100, 100, 0)$ to $(0, 0, 0)$ in 30 time steps. For the velocity distribution given by Equation (24), the corresponding time step k comes to 18.3590. The computation is stopped at 30 time steps because further advection of the hump is very slow as velocity approaches zero. Figure 8 shows the concentration distribution at the end of computation. Figure 9 compares results with the analytical solution along the line joining $(-220, 200)$ and $(220, -220)$ and normal passing through centroid of the hump as indicated in Figure 8. The peak value of concentration from the analytical solution is 0.8151, while the numerical solution gives 0.8429. Therefore, the numerical model overshoots the analytical solution by 3.40%. A similar situation has also been reported by Park and Liggett [22].

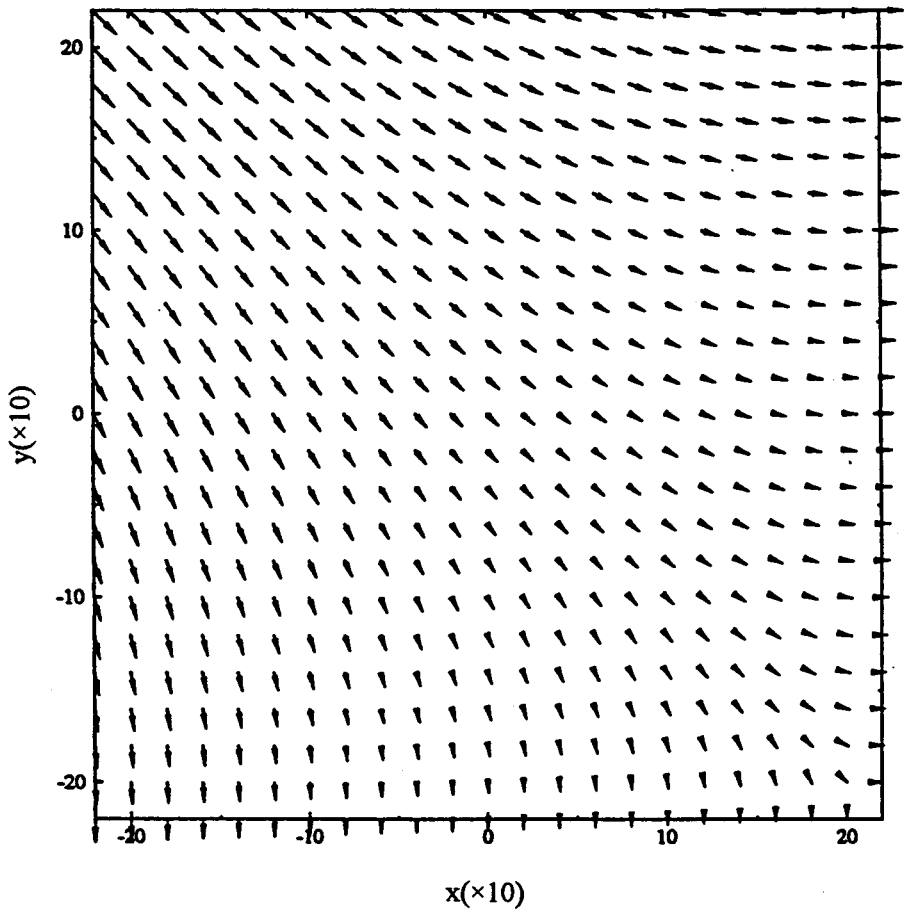

Figure 7. Velocity distribution in the x–y plane in example 4.

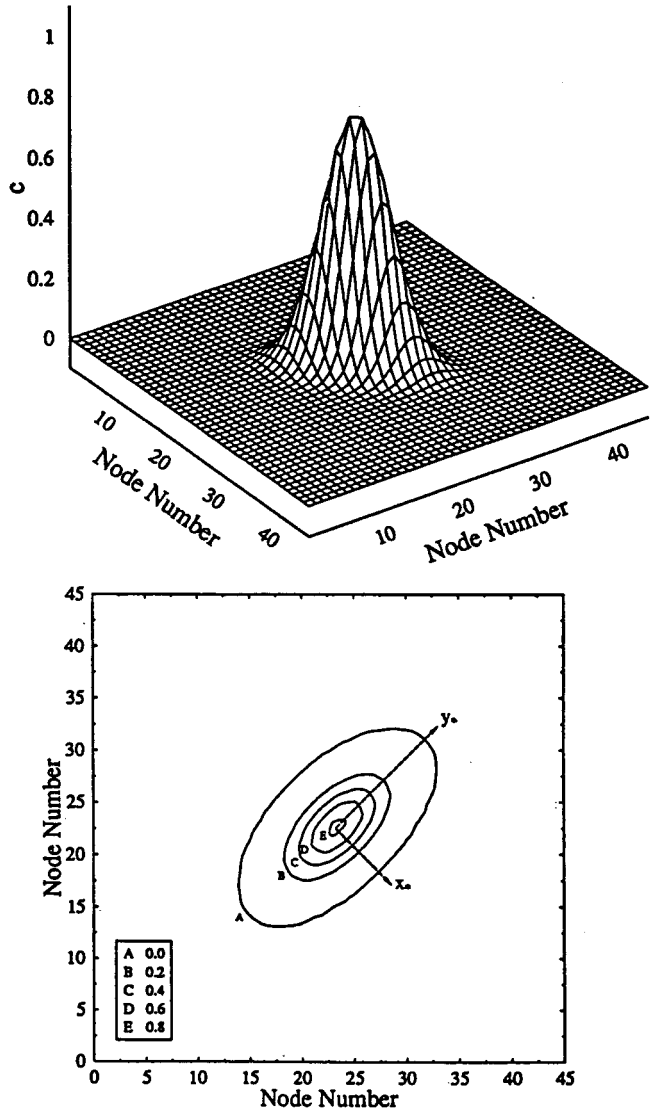

Figure 8. Spatial concentration distribution in example 4.

4.3. *Accuracy of solution*

The computational resources required for verifying accuracy of solution using three-dimensional problems, in general, exceed the computational capabilities which are currently available. Therefore, three two-dimensional problems (a) solid body rotation of Gaussian hump, (b) diffusion of the hump and (c) solid body rotation and diffusion of the hump are used to verify accuracy of the operator splitting algorithm. The computations are performed with (k, h), $(k/2, h/2)$ and $(k/4, h/4)$, where $k = 18.3589$, $h = 10$ and $\sigma_o = 20$, and simulation period $T = 917.943$. Errors in computed peak concentration ε are plotted in Figure 10 as a function of (k, h). The best fit lines and their slopes, indicating accuracy of solution, are also shown in the figure. Slopes of lines corresponding to the method of characteristics for the advection

equation, the finite element method for the diffusion equation and the splitting algorithm for the advection–diffusion equation are 3.12, 1.97 and 2.15 respectively. These results are consistent with the theoretical analysis of accuracies of the operator splitting algorithm and the numerical procedures for solving the split equations.

The numerical examples presented in this paper have been designed such that numerical damping is < 0.0001 per time step. In examples 1 and 2, representing pure advection problems, the expected peak concentration after 60 time steps is $0.9999^{60} = 0.9940$. The computed peak concentrations are within 2.0% of the predicted value. The discrepancy between predicted and computed result is mainly due to non-uniform velocity distribution. If spatial resolution of a problem does not meet the criteria of negligible numerical damping, then magnitude of errors in the computations will be high. For example, if $\sigma_o = 1.5h$ is used in example 1 or 2, then the initial condition will be distributed over approximately 14 elements in each direction. The corresponding numerical damping is 0.00485 per time step. In 60 time steps the peak concentration will be damped to $0.9952^{60} = 0.7470$, resulting in an error of about 25%.

5. SUMMARY AND CONCLUSIONS

An operator splitting algorithm for the three-dimensional advection–diffusion equation has been presented in this paper. The algorithm is a second-order-accurate adaptation of the Holly and Preissmann method, employing Strang type splitting. A backward method of characteristics combined with a C^0 continuous Hermite interpolation function is used to solve the advection equation. Following Holly and Preissmann, the equations for spatial concentration gradients in the Hermite interpolation function are solved in the advection step. For three-dimensional problems, solving the diffusion equation using the conventional finite element method results in a large system of simultaneous equations, requiring high computational time. To reduce the computational time and to make the composite algorithm efficient, diffusion

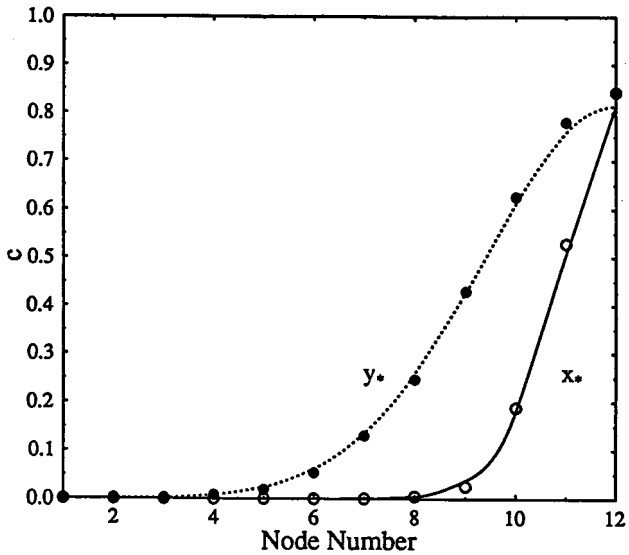

Figure 9. Comparison of numerical and analytical solution for example 4 along x_* and y_* as shown in Figure 8. (—) is the analytical solution, (○) is the numerical solution.

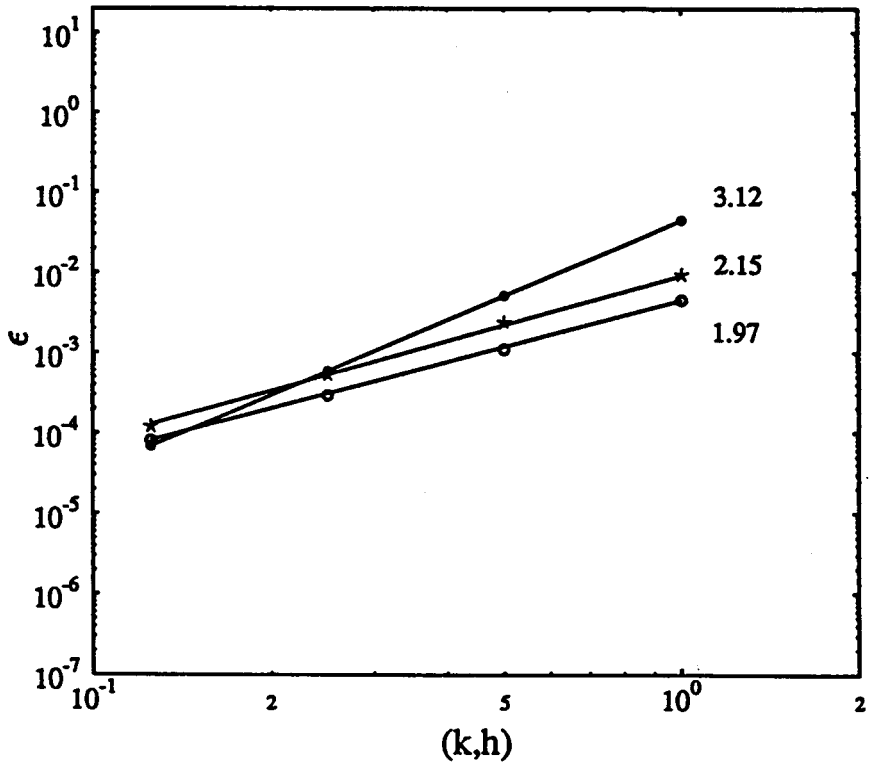

Figure 10. Accuracy of numerical procedures: (●) is the backward method of characteristics for advection equation, (○) is the finite element method for diffusion equation, and (✱) is the composite algorithm for the advection–diffusion equation.

equations for concentration and first-order concentration gradients are solved. The second-order concentration derivatives are obtained by numerical differentiation. As the diffusion step of computation is less sensitive to numerical errors, the proposed method is preferable to Rasch and Williamson [15]. Compared with Komatsu *et al.* [13,14], the present method retains the desirable features of the Holly and Preissmann scheme in developing an efficient operator splitting algorithm for the multidimensional advection–diffusion equation. The numerical examples, involving pure advection and advection dominated transport problems, indicate the good simulation characteristics of the algorithm for three-dimensional problems. The second-order-accuracy of the composite algorithm has been verified by numerical experiments.

ACKNOWLEDGMENTS

The research reported in this paper was in part supported by a research grant from the Xerox Corporation to Cornell University. During the course of study, the first author has also been supported by a Fellowship from the DeFrees Foundation. This research was conducted using the Cornell National Supercomputer Facility, which receives major funding from the National Science Foundation and IBM Corporation, and with additional support from New York State and members of the Corporate Research Institute.

REFERENCES

1. N.N. Yanenko, *The Method of Fractional Steps*, Springer, Berlin, 1971.
2. G.I. Marchuk, 'Splitting and alternating direction methods', in P.G. Ciarlet and J.L. Lions (eds.), *Handbook of Numerical Analysis*, *Vol. I*, *Finite Difference Methods (Part 1)—Solution of Equations in R^n (Part 1)*, North-Holland, Amsterdam, 1990, pp. 197–462.
3. L. Demkowicz, T.J. Oden and W. Rachowicz, 'A new finite element method for solving compressible Navier–Stokes equations based on an operator splitting method and h–p adaptivity', *Comput. Methods Appl. Mech. Eng.*, **84**, 275–236 (1990).
4. F.M. Holly Jr. and A. Preissmann, 'Accurate calculation of transport in two dimensions', *J. Hydr. Eng.*, **103**, 1259–1277 (1977).
5. J.-C. Yang and E.-L. Hsu, 'Time-line interpolation for solution of the dispersion equations', *J. Hydr. Res.*, **28**, 503–520 (1990).
6. G. Yang, P. Belleudy and A. Temperville, 'A higher-order Eulerian scheme for coupled advection diffusion transport', *Int. j numer. methods fluids*, **12**, 43–58 (1991).
7. D. Ding and P.L.-F. Liu, 'An operator splitting algorithm for two-dimensional convection–dispersion–reaction problems', *Int. j. numer. methods eng.*, **28**, 1023–1040 (1989).
8. J. Glass and W. Rodi, 'A higher order numerical scheme for scalar transport', *Comput. Methods Appl. Mech. Eng.*, **31**, 337–358 (1982).
9. F.M. Holly Jr. and J.M. Usseglio-Polatera, 'Dispersion simulation in two-dimensional flow', *J. Hydr. Eng.*, **110**, 905–926 (1984).
10. J.C. Yang, K.-N. Chen and H.-Y. Lee, Investigation of use of reach-back characteristic method for 2-D dispersion equations, *Int. j. numer. methods fluids*, **13**, 841–855 (1991).
11. L.A. Khan and P.L.-F. Liu, 'An operator splitting algorithm for coupled one-dimensional advection–diffusion–reaction equations', *Comput. Methods Appl. Mech. Eng.*, **127**, 181–201 (1995).
12. G. Strang, 'On the construction and comparison of difference schemes', *SIAM J. Numer. Anal.*, **5**, 506–517 (1968).
13. T. Komatsu, F.M. Holly Jr, N. Nakashiki and K. Ohgushi, 'Numerical calculation of pollutant in one and two dimensions', *J. Hydrosci. Hydr. Eng.*, **3**, 15–30 (1985).
14. T. Komatsu, K. Ohgushi, K. Asai and F.M. Holly Jr, 'Accurate numerical simulation of scalar advective transport', *J. Hydrosci. Hydr. Eng.*, **7**, 63–73 (1989).
15. P.J. Rasch and D.L. Williamson, 'On shape preserving interpolation and semi-Lagrangian transport', *SIAM J. Sci. Stat. Comput.*, **11**, 656–687 (1990).
16. D.L. Williamson and P.J. Rasch, 'Two-dimensional semi-Lagrangian transport with shape preserving interpolation', *Mon. Weather Rev.*, **117**, 102–129 (1989).
17. A.M. Baptista, E.E. Adams and K.D. Stolzenbach, 'Accuracy analysis of the backward method of characteristics', in A.S. Costa *et al.* (eds.), *Finite Elements in Water Resources*, Springer, Berlin, 1986, pp. 477–488.
18. R.J. LeVeque and J. Oliger, 'Numerical methods based on additive splitting for hyperbolic partial differential equations', *Math. Comput.*, **40**, 469–497 (1983).
19. L. Lapidus and G.F. Pinder, *Numerical Solution of Partial Differential Equations in Science and Engineering*, Wiley, New York, 1982.
20. L.A. Khan, 'An operator splitting algorithm for the three-dimensional advection–diffusion equation', *Ph.D. Thesis*, School of Civil and Environmental Engineering, Cornell University, Ithaca, New York, 1994.
21. L.A. Khan and P.L.-F. Liu, 'An operator splitting algorithm for the three-dimensional diffusion equation', *Numer. Methods PDE*, **11**, 617–624 (1995).
22. N.-S. Park and J.A. Liggett, 'Application of Taylor-Least squares finite element to three-dimensional advection–diffusion equation', *Int. j. numer. methods fluids*, **13**, 759–73 (1991).

