Journal of Computational Physics **152,** 493–516 (1999)
Article ID jcph.1999.6242, available online at http://www.idealibrary.com on IDEAL®

An Integer Lattice Realization of a Lax Scheme for Transport Processes in Multiple Component Fluid Flows

Sauro Succi,* Hudong Chen,† Chris Teixeira,† Gino Bella,‡
A. De Maio,‡ and Kim Molvig§

*Instituto Applicazioni Calcolo, 137 Viale Policlinico, Rome, Italy 00144; †Exa Corporation, 450 Bedford Street, Lexington, Massachusetts 02173; ‡Department of Mechanical Engineering, University of Rome "Tor Vergata," Rome, Italy; §Department of Nuclear Engineering, Massachusetts Institute of Technology, Memorial Drive, Cambridge, Massachusetts 02139

Received August 17, 1998; revised February 2, 1999

A Lax–Wendroff-like finite-difference representation for the transport of multiple chemical components is formulated via integer variables. This representation ensures exactly the desired conservation laws at all times and achieves low numerical diffusivity. The algorithm requires less memory as compared to its floating-point predecessor, hence *much* less than standard lattice gas and lattice Boltzmann methods to date. Analytical and numerical studies demonstrate that the algorithm is stable under subsonic conditions.

1. INTRODUCTION

The importance of computation with integer representations has become rapidly recognized since the emergence of lattice gas (LG) and lattice Boltzmann (LB) methods for computational fluid dynamics [1–8]. The most obvious reasons for this include less computational memory requirements and better consistency with modern computer architectures. More importantly, such representations are free of roundoff errors, so that fundamental conservation laws can be exactly enforced at all times. This is highly desirable in circumstances such as stiff and very long time numerical computations, in which error accumulation may severely affect the quality of the results. We are particularly interested in the simulation of reacting flows, possibly involving a significant number of chemical components, each at a different density and evolving on disparate spatial and time scales. As a result, exact preservation of the conservation properties becomes crucial in order to produce an accurate simulation [9].

0021-9991/99 $30.00

Reacting flow dynamics consists of three fundamental processes. These are: (1) reaction, in which some chemical components are turned into other chemical components; (2) mechanical feedback, in which the local fluid properties are altered due to heat release or consumption as a result of reactions; and (3) transport, in which these components are advected from place to place by the flow. In this paper, however, we only address the specific issue of how to simulate the transport of multiple chemical components: reaction and mechanical feedback will be presented in future work. Besides reacting flows, the present algorithm can, and already has been used for the numerical simulations of high-Reynolds flows within the framework of $k-\epsilon$ turbulence models, in which the turbulent kinetic energy k and the turbulent dissipation ϵ evolve according to passive-scalar-like dynamic equations plus local source terms which are readily incorporated within the formalism described in this work [10].

The paper is organized as follows. In the next section, we present a finite-difference scheme for solving the multiple component transport process (see Eq. (1) below). We provide a detailed stability analysis for such a scheme in the Appendix. In Section 3, we describe how to convert the scheme into an integer representation while satisfying all necessary conservation requirements. In addition, we discuss some possible methods to ensure the positivity of the density distribution functions. In Section 4, we present results of direct numerical simulations for selected test cases demonstrating the accuracy of the method. Finally, a discussion of the results is presented.

2. LAX–WENDROFF DISCRETIZATION SCHEME

Mathematically, the transport problem is defined by a set of passive scalar equations in which the motion of the various components is carried by a prescribed fluid velocity field,

$$\partial_t \rho_s + \boldsymbol{\nabla} \cdot (\rho_s \mathbf{u}) = \boldsymbol{\nabla} \cdot [D\rho \boldsymbol{\nabla}(\rho_s/\rho)], \tag{1}$$

where $\rho_s \equiv \rho_s(\mathbf{x}, t)$ is the local mass density of the sth component at $(\mathbf{x}, t)$; $s = 1, \ldots, S$. The fluid velocity, $\mathbf{u}$ is a prescribed function of space and time. The term on the right-hand side of Eq. (1) describes the diffusion of the sth component with diffusivity coefficient D. For simplicity we shall assume D is the same for all species. The form of this equation is consistent with the continuity equation for the overall mass density, $\rho(\mathbf{x}, t) \equiv \sum_{s=1}^{S} \rho_s(\mathbf{x}, t)$,

$$\partial_t \rho + \boldsymbol{\nabla} \cdot (\rho \mathbf{u}) = 0 \tag{2}$$

because the right-hand side of (1) vanishes upon summing over all components. One of the most popular finite difference schemes for approximating an advection process, ignoring the diffusion effect, is the Lax scheme [11]. In a d-dimensional Cartesian mesh spanned by integers l_α along direction x_α according to $\mathbf{x} = \sum_{\alpha=1}^{d} l_\alpha \Delta_\alpha \hat{x}_\alpha$, the Lax scheme may be represented by the difference form

$$\rho_s(\mathbf{x}, t + \Delta t) = \frac{1}{2} \sum_{\alpha} \left[\left(\frac{1}{d} + U_\alpha^- \right) \rho_s(\mathbf{x} - \Delta_\alpha \hat{\alpha}, t) + \left(\frac{1}{d} - U_\alpha^+ \right) \rho_s(\mathbf{x} + \Delta_\alpha \hat{\alpha}, t) \right], \tag{3}$$

where the subindex $\alpha (= x, y, z)$ runs over the Cartesian components and $\hat{\alpha} \equiv \hat{x}_\alpha$. $U_\alpha^\pm \equiv u_\alpha(\mathbf{x} \pm \Delta_\alpha \hat{\alpha}, t)(\Delta t/\Delta_\alpha)$ is the Courant number in the αth dimension.

A natural way for dealing with the diffusion term on the right-hand side of Eq. (1) is to use the centered-difference approach. As detailed in the sequel, this would lead to numerical instability because the right-hand-side of Eq. (1) contains a hidden convective component. Consequently, an alternative formulation is called for. To this purpose, we rewrite the diffusion term in the form

$$\begin{aligned}\boldsymbol{\nabla}\cdot[D\rho\boldsymbol{\nabla}(\rho_s/\rho)] &= -\boldsymbol{\nabla}\cdot\left[\left(D\frac{\boldsymbol{\nabla}\rho}{\rho}\right)\rho_s\right]+\boldsymbol{\nabla}\cdot[D\boldsymbol{\nabla}\rho_s]\\ &= -\boldsymbol{\nabla}\cdot\left[\left(D\frac{\boldsymbol{\nabla}\rho}{\rho}\right)\rho_s\right]+\boldsymbol{\nabla}\cdot[(\boldsymbol{\nabla}D)\rho_s]+D\boldsymbol{\nabla}^2\rho_s-(\boldsymbol{\nabla}^2D)\rho_s \quad (4)\end{aligned}$$

so that the third term on the right-hand side in the above expression becomes the ordinary diffusion term, while the first two terms can be combined and together act like an effective advection (reminiscent of the so-called Fick's law) with the velocity,

$$\mathbf{u}_h = D\frac{\boldsymbol{\nabla}\rho}{\rho}-\boldsymbol{\nabla}D. \quad (5)$$

Recognizing the advection nature of this combined term, we can immediately understand why the naive approach suggested above does not work: it is known that a centered-difference representation for advection as combined with explicit first order time marching is unconditionally unstable [11, 12]. Equation (1) can be cast into a new form,

$$\partial_t\rho_s+\boldsymbol{\nabla}\cdot(\rho_s\tilde{\mathbf{u}}) = D\boldsymbol{\nabla}^2\rho_s-(\boldsymbol{\nabla}^2D)\rho_s, \quad (6)$$

where $\tilde{\mathbf{u}}\equiv\mathbf{u}+\mathbf{u}_h$ is now the overall generalized advection velocity. Both the term involving the Laplacian of the diffusivity in Eq. (6) and the second term in the expression for the additional effective velocity, Eq. (5), vanish in the case of constant molecular diffusivity, D.

The Lax finite-difference approximation for advection can still be adopted, except that $U_\alpha^\pm$ is now understood to be the generalized $\tilde{\mathbf{u}}$ as opposed to the original fluid velocity. The centered difference scheme can be safely used for approximating the regular diffusion term,

$$D\boldsymbol{\nabla}^2\rho_s \approx \sum_\alpha \frac{D}{\boldsymbol{\Delta}_\alpha^2}[\rho_s(\mathbf{x}+\Delta_\alpha\hat{\alpha},t)-2\rho_s(\mathbf{x},t)+\rho_s(\mathbf{x}-\Delta_\alpha\hat{\alpha},t)] \quad (7)$$

as well as the term involving the Laplacian of the diffusivity.

As it is well known, the Lax scheme is stable as long as the extended CFL (Courant–Friedrichs–Lewy) condition is satisfied, $\sum_{\alpha=1}^d(u_\alpha(\mathbf{x},t)\Delta t/\Delta_\alpha)^2\le 1/d$ [11, 13]. However, it generates a numerical diffusion of the form

$$D_{n,\alpha} = \frac{1}{2d}\left[1-d\tilde{U}_\alpha^2\right]\frac{\Delta_\alpha^2}{\Delta t}, \quad (8)$$

where $\tilde{U}_\alpha\equiv\tilde{u}_\alpha(\Delta t/\Delta_\alpha)$ is the local Courant number along direction α. This indicates that the numerical diffusivity vanishes only when the magnitude of the Courant number takes the value $1/d$. For a given practical resolution, such numerical diffusion is often unacceptable, and we must seek a way to eliminate it. Based on direct analysis, it can be shown that one way to remove it is to simply introduce an artificial negative diffusion of the same form as in (8) but with an opposite sign. This is equivalent to the Lax–Wendroff scheme [11].

It is quite interesting to realize (see details in the Appendix) that such a modification, together with considerations for cross-product effects in higher dimensions, removes the leading order numerical diffusion in the standard Lax scheme and yet it maintains stability. Obviously we can discretize this additional term via a centered difference approximation. After combining all these considerations, we obtain a generalized Lax-like scheme for the multiple component transport process, and the final form of the finite-difference equation becomes

$$\begin{aligned}\rho_s(\mathbf{x}, t+\Delta t) = {} & \frac{1}{2}\sum_\alpha \left[\left(\frac{1}{d}+\tilde{U}_\alpha^-\right)\rho_s(\mathbf{x}-\Delta_\alpha\hat{\alpha}, t)+\left(\frac{1}{d}-\tilde{U}_\alpha^+\right)\rho_s(\mathbf{x}+\Delta_\alpha\hat{\alpha}, t)\right] \\ & + \sum_\alpha [\rho_s(\mathbf{x}+\Delta_\alpha\hat{\alpha}, t) - 2\rho_s(\mathbf{x}, t) + \rho_s(\mathbf{x}-\Delta_\alpha\hat{\alpha}, t)]\mathcal{D}_\alpha \\ & - \sum_\alpha [D(\mathbf{x}+\Delta_\alpha\hat{\alpha}, t) - 2D(\mathbf{x}, t) + D(\mathbf{x}-\Delta_\alpha\hat{\alpha}, t)]\frac{\Delta t}{\Delta_\alpha^2}\rho_s(\mathbf{x}, t) \\ & - \frac{1}{8}\sum_{\alpha,\beta\neq\alpha}\{\tilde{U}_\alpha^-\tilde{U}_\beta(\mathbf{x}-\Delta_\alpha\hat{\alpha}, t)[\rho_s(\mathbf{x}-\Delta_\alpha\hat{\alpha}+\Delta_\beta\hat{\beta}, t) \\ & - \rho_s(\mathbf{x}-\Delta_\alpha\hat{\alpha}-\Delta_\beta\hat{\beta}, t)] - \tilde{U}_\alpha^+\tilde{U}_\beta(\mathbf{x}+\Delta_\alpha\hat{\alpha}, t)[\rho_s(\mathbf{x}+\Delta_\alpha\hat{\alpha}+\Delta_\beta\hat{\beta}, t) \\ & - \rho_s(\mathbf{x}+\Delta_\alpha\hat{\alpha}-\Delta_\beta\hat{\beta}, t)]\}, \end{aligned} \tag{9}$$

where

$$\tilde{U}_\alpha^\pm = \tilde{U}_\alpha(\mathbf{x}\pm\Delta_\alpha\hat{\alpha}, t)$$

and

$$\mathcal{D}_\alpha = \left[D\frac{\Delta t}{\Delta_\alpha^2} - \frac{1}{2d}\left(1-d(\tilde{U}_\alpha)^2\right)\right].$$

The terms in the double summations are for the purpose of eliminating cross-product errors when $d > 1$. Note that while the discretization of the diffusion term (terms on the second line above) involves a combination of the molecular and numerical diffusivities, the Laplacian of the diffusion (terms on the third line above) involves the molecular diffusion *only*. For the remaining sections of this paper, we adopt the simplifying assumption that the molecular diffusion, D, is constant.

Before moving further, it is worth mentioning that the stability problem discussed in the present section would not arise had we chosen the mass fractions $Y_s = \rho_s/\rho$ instead of the partial densities ρ_s, as dependent variables. This is because in the mass-fraction representation, the right-hand-side of Eq. (1) is purely diffusive and can be harmlessly center-differenced.

However, since our highlight is the use of integers, it appears much more natural and convenient to use densities as dependent variables than mass fractions since the latter are intrinsically fractional numbers. Hence, the idea is to prioritize and leverage exact conservativeness over straightforward stability. Note that in light of the stability analysis presented in the Appendix, this choice does not imply *any* sacrifice in terms of numerical stability. All it takes is a different, slightly more elaborated, formulation of the problem, namely the one detailed in the present section. This is a mostly welcome and non-trivial result

(in more than one dimension) as it permits us to do away with implicit time marching, hence preserve a basic asset of lattice methods, namely high amenability to parallel computing.

3. INTEGER REALIZATION AND EXACT CONSERVATION LAWS

The Lax–Wendroff scheme satisfies mass conservation for each chemical component up to the precision of the representation of the component densities. The main goal of the integer formulation is to make this conservation exact. For the purpose of the consequent integer formulation, it is advantageous to take the following perspective. Equation (9) can be viewed as a result of a two-step partitioning process as described below: During each time t, the first step is to split the component density, $\rho_s(\mathbf{x}, t)$, at the node into $2d+1$ *directional densities*, each associated with one of the $2d+1$ nearest-neighbor nodes (including itself) on a given Cartesian mesh. This splitting step can be represented as

$$\rho_s(\mathbf{x}, t) \to \sum_{l=-1}^{+1} \sum_{\alpha} \rho_{s,\alpha}^{l}(\mathbf{x}, t) + \rho_s^0(\mathbf{x}, t), \tag{10}$$

where the $l=0$ has been explicitly singled out and is not included in the first summation, i.e., $\sum_{l=-1}^{+1} \sum_{\alpha} 1 = 2d$. Here

$$\rho_{s,\alpha}^{l}(\mathbf{x}, t) = P_{\alpha}^{l}(\mathbf{x}, t)\rho_s(\mathbf{x}, t) + Q_{\alpha}^{l}(\mathbf{x}, t) \tag{11}$$

and

$$\rho_s^0(\mathbf{x}, t) = P^0(\mathbf{x}, t)\rho_s(\mathbf{x}, t) + Q^0(\mathbf{x}, t). \tag{12}$$

Such a partition has been designed to satisfy mass conservation, i.e., the sign $\to$ becomes $=$, if the weights obey the normalization condition,

$$\sum_{l=-1}^{+1} \sum_{\alpha} P_{\alpha}^{l}(\mathbf{x}, t) + P^0(\mathbf{x}, t) = 1,$$

and

$$\sum_{l=-1}^{+1} \sum_{\alpha} Q_{\alpha}^{l}(\mathbf{x}, t) + Q^0(\mathbf{x}, t) = 0.$$

For the Lax–Wendroff scheme given in the previous section, these weights are specifically defined as

$$P_{\alpha}^{l}(\mathbf{x}, t) = \frac{1}{2}\left[\frac{1}{d} + l\tilde{U}_{\alpha}(\mathbf{x}, t)\right]\mathcal{D}_{\alpha}(\mathbf{x}, t) \tag{13}$$

$$P^0(\mathbf{x}, t) = -2\sum_{\alpha} \mathcal{D}_{\alpha}(\mathbf{x}, t) \tag{14}$$

$$Q_{\alpha}^{l}(\mathbf{x}, t) = -\frac{l}{8}\tilde{U}_{\alpha}(\mathbf{x}, t) \sum_{\beta \neq \alpha} \tilde{U}_{\beta}(\mathbf{x}, t)[\rho_s(\mathbf{x} + \Delta_{\beta}\hat{\beta}, t) - \rho_s(\mathbf{x} - \Delta_{\beta}\hat{\beta}, t)] \tag{15}$$

and

$$Q^0(\mathbf{x}, t) = 0 \tag{16}$$

which can be directly verified as satisfying the normalization condition.

The second step in the scheme is to move the directional densities (11) and (12) constructed above to each of their own corresponding neighboring nodes. This is the advection step. That is, we define new quantities, $\bar{\rho}_s$, such that

$$\rho_{s,\alpha}^{-l}(\mathbf{x}, t) = \rho_{s,\alpha}^{l}(\mathbf{x} - l\Delta_\alpha \hat{\alpha}, t) \tag{17}$$

and

$$\rho_s^{-0}(\mathbf{x}, t) = \rho_s^0(\mathbf{x}, t) \tag{18}$$

so that the new density distribution for each component at the next time step is a result of the summation of the advected $\bar{\rho}_s$,

$$\rho_s(\mathbf{x}, t+1) = \sum_{l=-1}^{+1} \sum_{\alpha} \rho_{s,\alpha}^{-l}(\mathbf{x}, t) + \rho_s^{-0}(\mathbf{x}, t). \tag{19}$$

Because this step is simply a spatial relocation of the directional densities, it obviously conserves the global mass in the system for each component.

Using the above two-step partitioning point of view, we can easily construct an integer realization of the transport scheme. In an integer formalism, each component density $\rho_s(\mathbf{x}, t)$ is represented by an integer, which can directly be interpreted as a number of "macromolecules" of the given species, and for the sake of simplicity, we rename this density as $N_s(\mathbf{x}, t)$. In order to ensure exact mass conservation, the splitting step in the above must be implemented carefully. First we integerize $\rho_{s,\alpha}^{l}(\mathbf{x}, t)$ and $\rho_s^0(\mathbf{x}, t)$ as

$$N_{s,\alpha}^{l}(\mathbf{x}, t) = \mathrm{Int}\left[\rho_{s,\alpha}^{l}(\mathbf{x}, t) + \mathrm{rand}_{s,\alpha}^{l}(\mathbf{x}, t)\right], \tag{20}$$

where the operator, Int, truncates off the values to the right of the decimal point. Also, $\mathrm{rand}_{s,\alpha}^{l}(\mathbf{x}, t)$ is a random number uniformly distributed between 0 and 1, with $\langle \mathrm{rand}_{s,\alpha}^{l}(\mathbf{x}, t) \rangle = 0.5$. Note that the quantity $\rho_{s,\alpha}^{l}(\mathbf{x}, t)$ is represented and computed in real numbers as defined previously according to Eq. (11). This integer operation achieves the real number accuracy in a statistical sense, i.e., $\langle N_{s,\alpha}^{l}(\mathbf{x}, t) \rangle = \rho_{s,\alpha}^{l}(\mathbf{x}, t)$. Furthermore, instead of using Eq. (12), the integer $N_s^0(\mathbf{x}, t)$ is computed in the following way to enforce the mass conservation explicitly,

$$N_s^0(\mathbf{x}, t) = N_s(\mathbf{x}, t) - \sum_{l=-1}^{+1} \sum_{\alpha} N_{s,\alpha}^{l}(\mathbf{x}, t), \tag{21}$$

i.e., all of the remaining density after the advected components are integerized is placed in the original location.

The advection step is unchanged except for replacing ρ_s with N_s in Eqs. (17), (18), and (19).

It can be seen that the integer process defined above enforces exact conservation laws at all times and realizes the same averaged transport evolution equations given by Eq. (9) in a statistically averaged sense. This is true because the transport equations are linear.

4. NUMERICAL SIMULATIONS

The present algorithm has been implemented with 16-bit integer ($0 \le N_s \le 2^{16} - 1 = 65{,}535$) representation and tested on the following two-dimensional cases:

1. Purely diffusive flow (no advection)
2. Uniform advection-diffusion (uniform flow)
3. Poiseuille advection-diffusion flow (Taylor hydrodynamic dispersion).

In all cases the species densities are initialized according to a Gaussian profile superimposed on an uniform background

$$\rho_s(x, 0) = b_s + a_s \exp\left(-\frac{r_s^2}{2}\right), \qquad s = 0, 1, \tag{22}$$

where $r_s^2 = ((x - X_s)/\sigma_{xs})^2 + ((y - Y_s)/\sigma_{ys})^2$, and X_s, Y_s, σ_{xs}, and σ_{ys} are the centers and half-widths of the Gaussian profile, respectively. This is a convenient choice in order to measure advection and diffusion in terms of translation and spreading of the initial profile as it evolves in time. The parameters of the Gaussian profile and background are chosen in such a way as to ensure that the total density $\rho = \rho_0 + \rho_1$ is uniform in space. Note that since the diffusivities are constants, this implies $\mathbf{u}_h = 0$ throughout all test cases (see Eq. (5)). The boundary conditions are as follows: periodic on inlet/oulet and no species flow at top/bottom walls.

4.1. Purely Diffusive Flow ($\mathbf{u} = 0$, D = const)

The main aim of these simulations is to demonstrate that our numerical scheme has virtually no numerical diffusion.

To this purpose, three sets of simulations at high ($D = 0.1$), low ($D = 0.001$), and no-diffusivity at all ($D = 0$) have been performed.

The simulations are run on a 100×50 lattice. The other parameters are set as follows: $\sigma_{0x} = \sigma_{1x} = \sigma_{0y} = \sigma_{1y} = 5$, $X_0 = X_1 = 50$, $Y_0 = Y_1 = 25$, $b_0 = 32{,}000$, $b_1 = 320$, $a_0 = -3200$, $a_1 = 3200$. This corresponds to $N_0 = 159{,}494{,}235$ particles in the majority species and $N_1 = 2{,}101{,}844$ particles in the minority one. Both numbers are *exactly* conserved throughout the simulation.

4.1.1. $\mathbf{u} = 0$, $D = 0$

The mid-plane ($y = 25$) densities ρ_0, ρ_1 as a function of the streamwise coordinate x at time $t = 0$, 100, 200, 300, 400, 500 are shown in Figs. 1 and 2. From these figures we see that the initial profiles are basically unperturbed as time unfolds. This is exactly what they should do in the absence of numerical diffusion, since the physical diffusivity of both species is set to zero.

4.1.2. $\mathbf{u} = 0$, $D = 0.001$

The same test with $D = 0.001$ is presented in Figs. 3 and 4, from which a small broadening of the initial profiles is visible. A quantitative assessment of this broadening is shown in Fig. 5, where the squared variance of the density profile $\sigma^2 = \sum_l p_l[(x_l - \bar{x})^2 + (y_l - \bar{y})^2]$, is represented as a function of time for both species. In this equation $p_l = \rho_l / \sum_l \rho_l$, $\rho_l \equiv \rho(x_l, y_l)$ and $\bar{x} = \sum_l p_l x_l$, $\bar{y} = \sum_l p_l y_l$ are the average positions associated with the

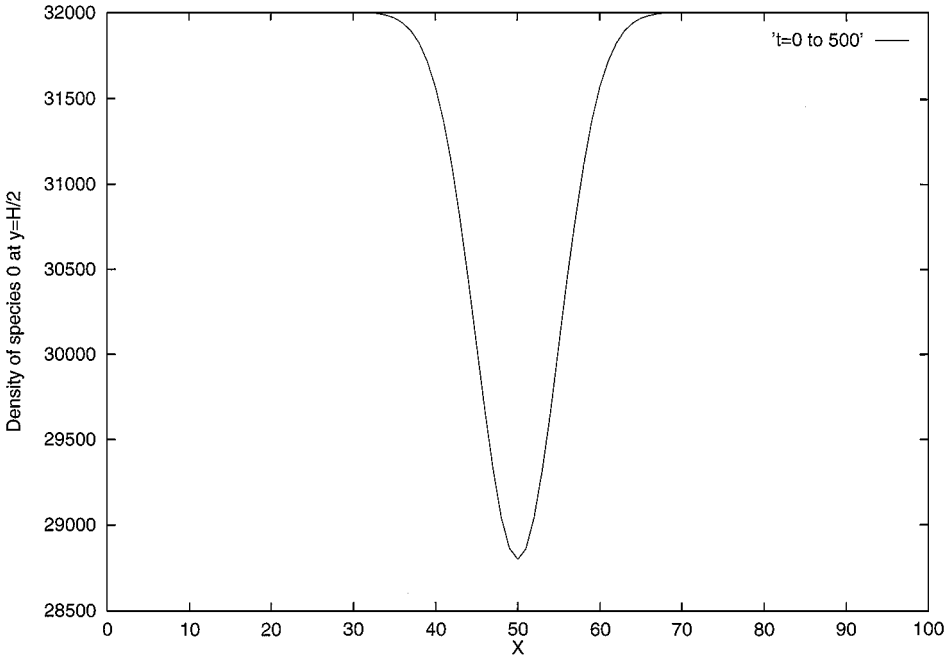


FIG. 1. Mid-plane density of species zero as a function of x at $t = 0, 100, 200, 300, 400, 500$, for $U_0 = 0,\ D = 0$. Here and throughout, time is given in lattice time-steps. Note that all six curves collapse into a single one, showing that the system does not move away from the initial condition.

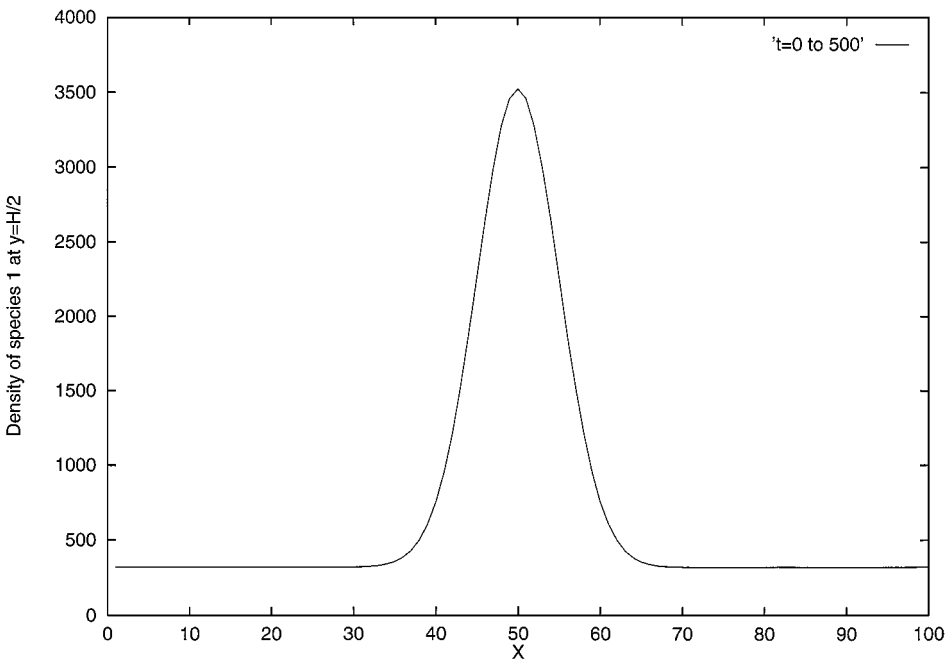


FIG. 2. Mid-plane density of species one as a function of x at $t = 0, 100, 200, 300, 400, 500$ for $U_0 = 0,\ D = 0$. Note that all six curves collapse into a single one since the system does not move from the initial condition.

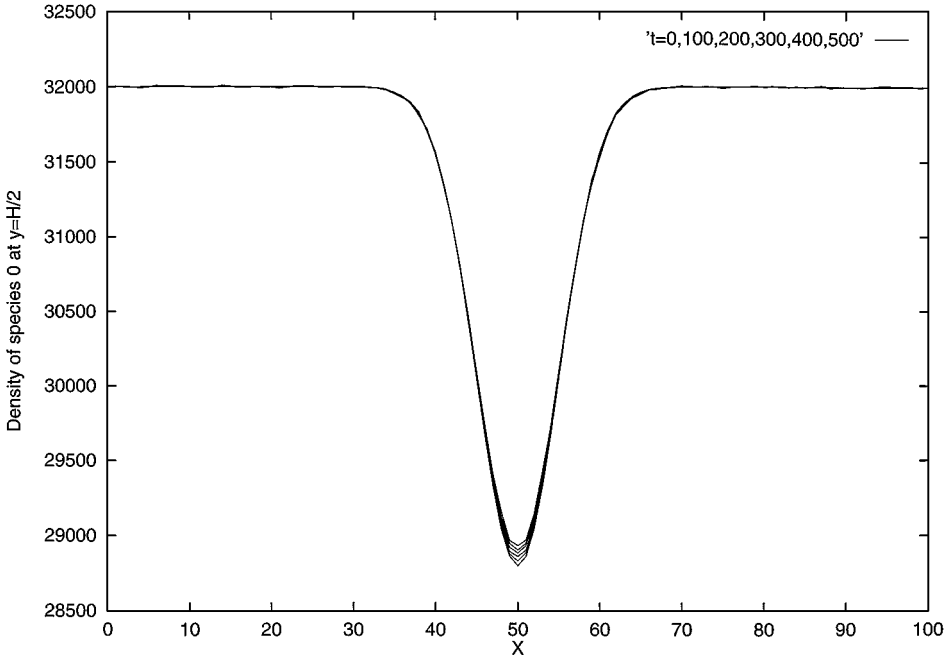


FIG. 3. Mid-plane density of species zero as a function of x for $t = 100$, 200, 300, 400, 500, for $U_0 = 0$, $D = 0.001$. Note the mild broadening of the profiles as time unfolds, due to non-zero diffusivity.

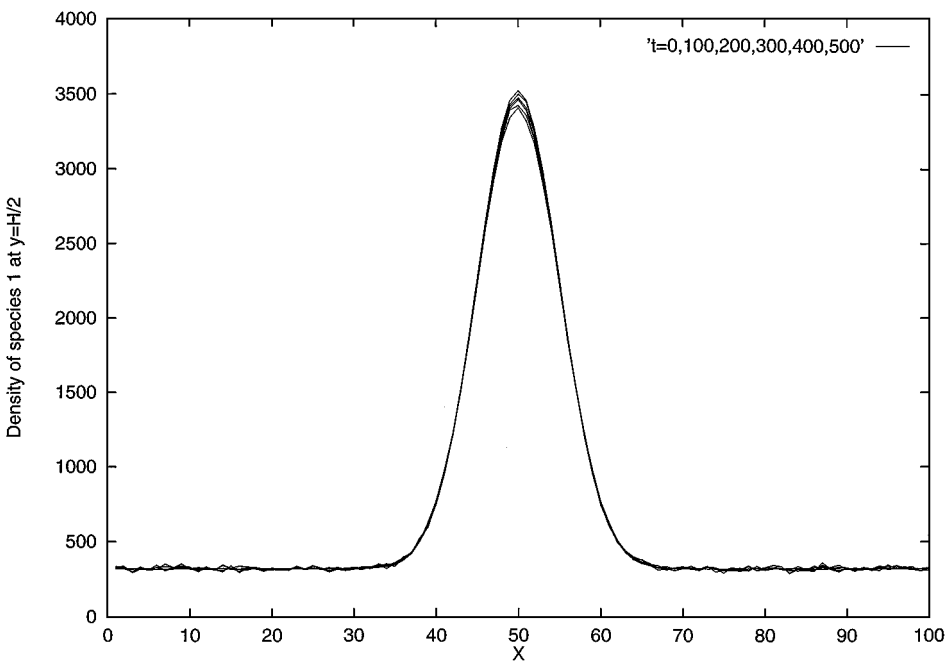


FIG. 4. Mid-plane density of species one as a function of x for $t = 100$, 200, 300, 400, 500, for $U_0 = 0$, $D = 0.001$.

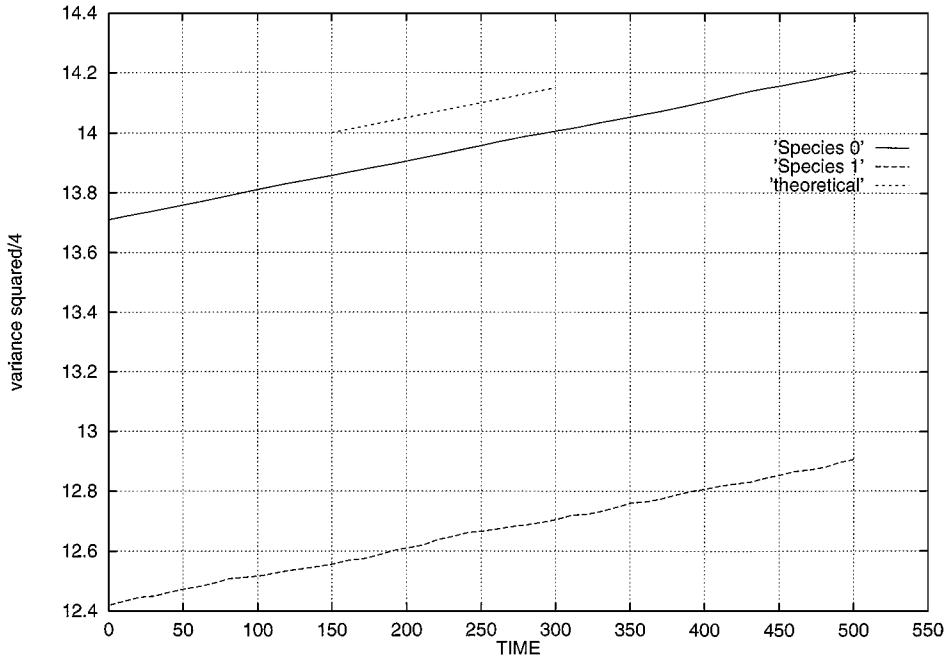


FIG. 5. Variance squared of species zero and one as a function of time. The straight short-dashed line represents the analytical result $D = 0.001$.

species densities. The discrete index l runs over the two-dimensional lattice and the species index s is dropped for simplicity.

The diffusivity is measured in the numerical experiment according to the standard relation $D_e(t) = \sigma^2(t)/4t$, the expectation being that if numerical diffusion is negligible, $D_e(t)$ should match exactly the input value of physical diffusivity D.

This expectation is basically confirmed by the results shown in Fig. 5 (the short-dashed line denotes the analytical value $D = 0.001$). The quantitative values from a linear best-fit to the data give $D = 0.995\ 10^{-3}$ for the bulk species (species 0) and $D = 0.974\ 10^{-3}$ for the tracer species (species 1) which agree well with the analytical value. We note that the estimate of the diffusivity on the tracer species is clearly affected by statistical noise. This is not surprising due to the small number of integer particles used to represent the minority species. In fact, the same estimate on the majority species is much less exposed to fluctuations.

Similar tests on the high diffusivity case ($D = 0.1$) provide basically the same sort of quantitative agreement between D_e and D.

We conclude that the numerical scheme is indeed virtually free from numerical diffusion effects.

4.2. Uniform Flow (u = const, D = const)

The next set of simulations refers to the case of a Gaussian profile moving in an uniform flow. The goal is to show that the low-diffusivity property of this scheme is still preserved under convective flow, and also to gain insight into the dispersive effects associated with the numerical discretization.

We have performed a series of simulations with various values of the speed $\mathbf{u} = U_0\hat{x}$ and two different molecular diffusion coefficients $D = 0.1$ and $D = 0.0$. The other parameters are

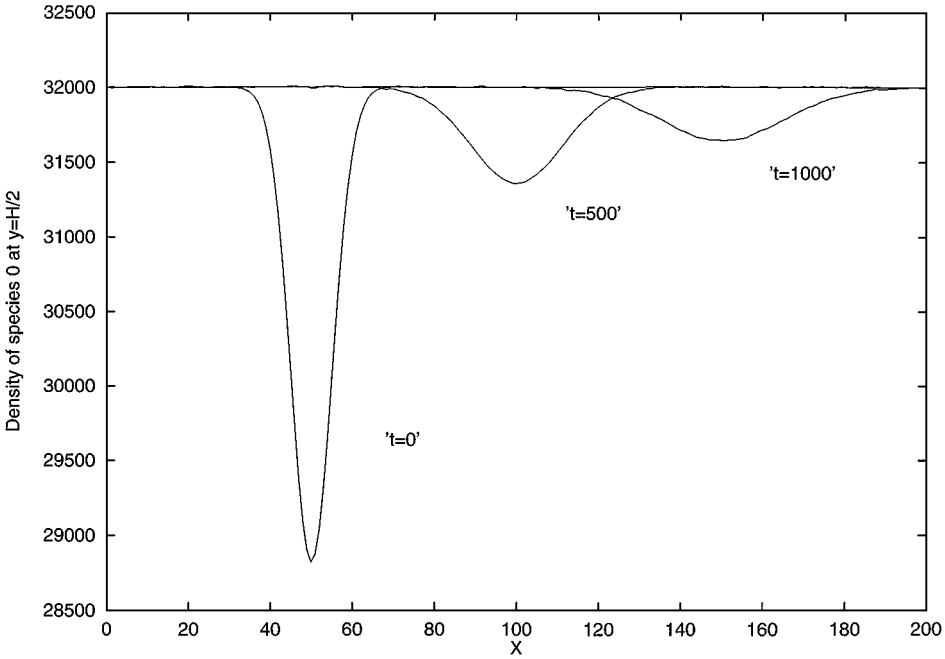


FIG. 6. Mid-plane density of species one as a function of x for $t = 0$, 500, 1000, for $U_0 = 0.1$, $D = 0.1$. Flow is moving to the right.

set as follows: grid 200×25, $b_0 = 32000$, $b_1 = 320$, $a_0 = -3200$, $a_1 = 3200$, $\sigma_{0x} = \sigma_{1x} = 5$, $\sigma_{0y} = \sigma_{1y} = 5$, $X_0 = X_1 = 50$, $Y_0 = Y_1 = 12$. This corresponds to $N_0 = 153{,}104{,}318$, $N_1 = 2{,}020{,}795$.

4.2.1. $U_0 = 0.1$, $D = 0.1$

Indeed, the diffusive properties of the algorithm stay basically unchanged, as they should on account of Galilean invariance. This is shown in Figs. 6–7, which refer to the density profiles of both species at $t = 0, 500, 1000$ for the case $U_0 = 0.1$, $D = 0.1$. From these figures, we observe that the profiles advect and diffuse at the correct rates, as quantitatively shown in Figs. 8 and 9 reporting the mean positions along x and y, as well as the variances squared, as a function of time. As expected, the mean positions and variances obey the theoretical relation $\overline{x}_s(t) - \overline{x}_s(0) = U_0 t$, $\overline{y}_s(t) - \overline{y}_s(0) = 0$, $s = 0, 1$, and $\sigma_s^2(t) - \sigma_s^2(0) = 4Dt$, $s = 0, 1$.

4.2.2. $U_0 = 0.1$, $D = 0$

As physical diffusion is lowered, dispersive effects start to appear upstream of the Gaussian profile. These are evidenced in Figs. 10–11 which refer to the extreme case of zero physical diffusion ($D = 0$) and flow speed $U_0 = 0.1$. Dispersion materializes in the form of density ripples on the rear side of the moving profile, the well-known Gibbs phenomenon. These ripples grow with increasing flow speed and smear out with increasing physical diffusivity. The control parameter governing the intensity of these ripples is the cell Peclet Number $Pe_\Delta = \frac{U\Delta x}{D}$, which in lattice units ($\Delta x = 1$) is simply the ratio U/D.

Our numerical experiments indicate that the ripples amplitude does not take on significant values for cell-Peclet numbers below approximately 10. Higher Pe_Δ result in sustained growth of the ripple amplitudes up to the point where the minority species is driven to negative values. At this stage, an underflow control mechanism is required. Since the main

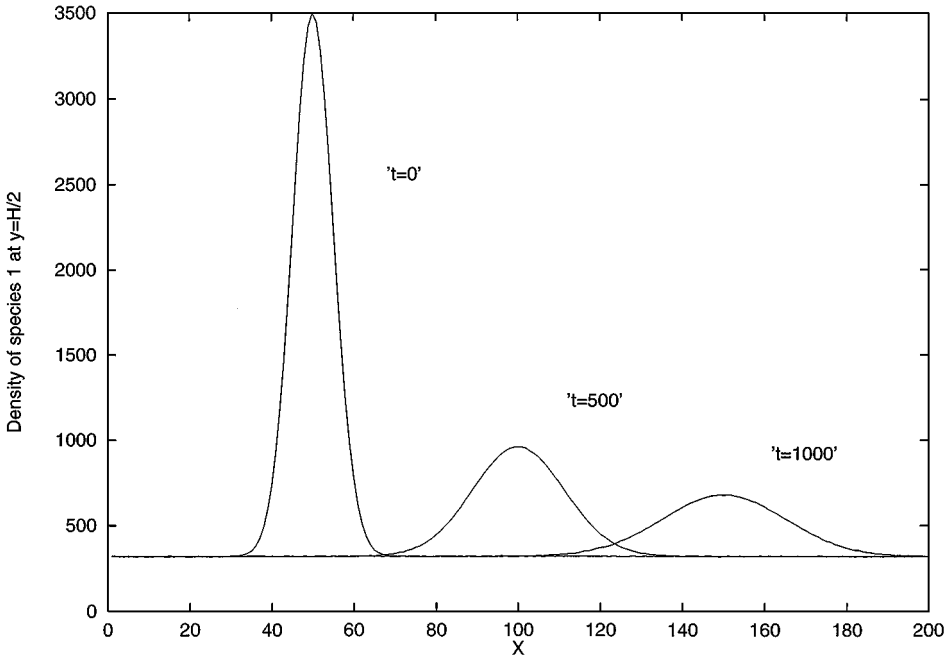


FIG. 7. Mid-plane density of species one as a function of x at $t = 0$, 500, 1000, for $U_0 = 0.1$, $D = 0.1$. The flow is moving rightwards.

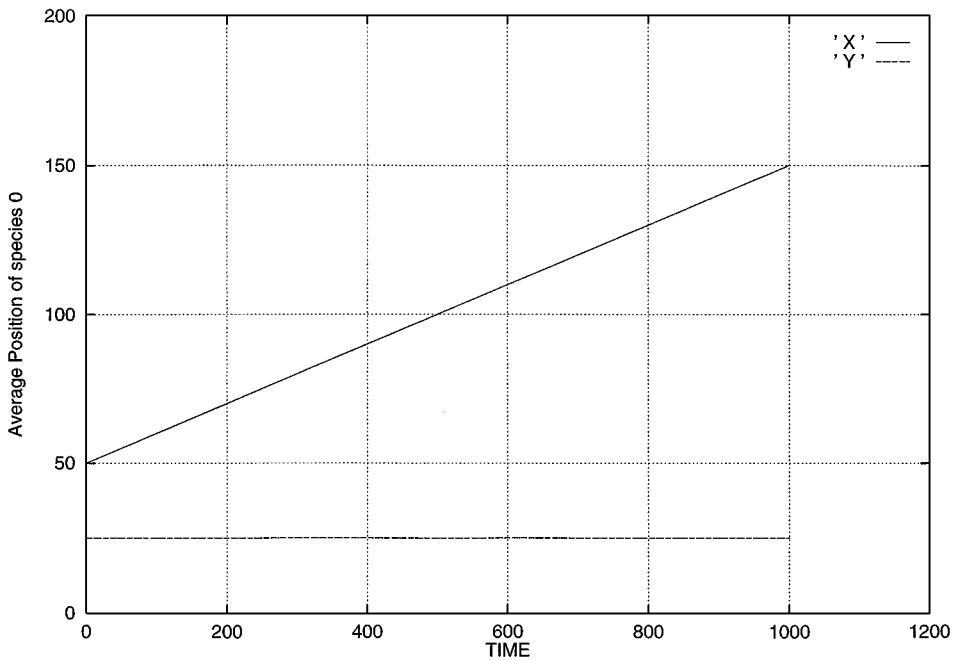


FIG. 8. Average position along x and y of species zero as a function of time for $U = 0.1$, $D = 0.1$.

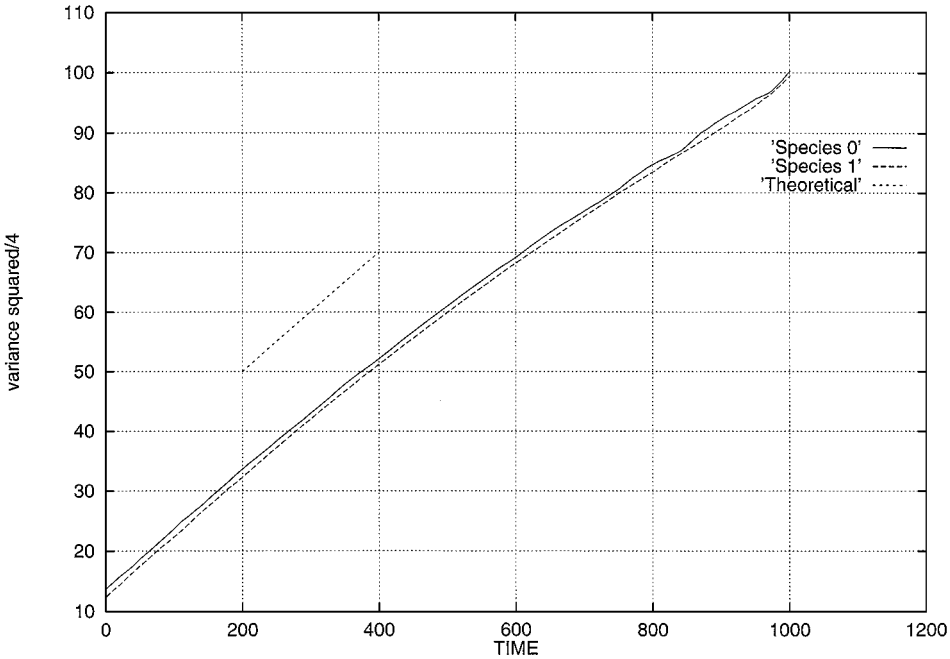


FIG. 9. Variance squared of species zero and one as a function of time for $U_0 = 0.1$, $D = 0.1$.

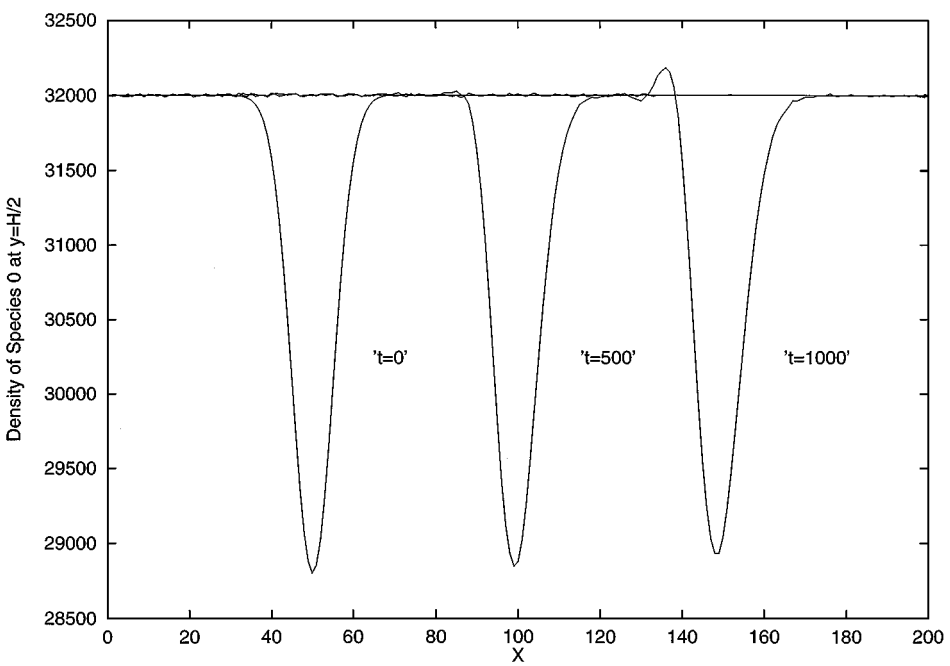


FIG. 10. Mid-plane density of species zero as a function of x for $t = 0, 500, 1000$, for $U_0 = 0.1$, $D = 0$. The flow is moving to the right.

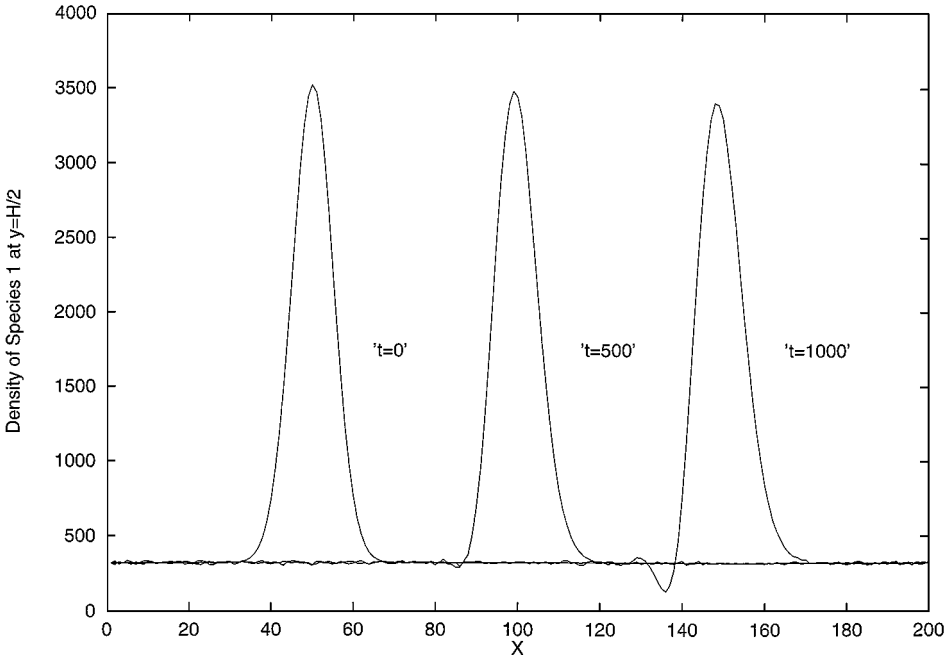


FIG. 11. Mid-plane density of species one as a function of x for $t = 0$, 500, 1000, for $U_0 = 0.1$, $D = 0$. The flow is moving to the right.

purpose of this work is to analyze the numerical properties of the "plain" integer Lax–Wendroff scheme, to date, the underflow control consists in simply resetting negative values to zero. Since this breaks particle conservation, future work shall be devoted to the development of more sophisticated dispersion-limiters, possibly in the direction of monotonicity-preserving schemes. For instance, preliminary tests show that by prefactoring the Wendroff term (8) with a disposable parameter $0 < \theta < 1$, mass positivity can be ensured without violating exact conservation. By so doing, further stability gains have been observed in numerical simulations of $k - \epsilon$ turbulence models [10].

4.3. Taylor's Hydrodynamic Dispersion

As a third test case we consider the tracer diffusion and hydrodynamic dispersion in a Poiseuille velocity profile (Taylor hydrodynamic dispersion).

Here, besides molecular diffusion and convective motion, the species undergo hydrodynamic dispersion on account of the stretching effect exerted by the sheared flow configuration. The aim of this section is to test whether our integer scheme correctly captures this hydrodynamic stretch mechanism. We have performed a series of simulations with various values of the centerline speed U_0 and two different molecular diffusion coefficients $D = 0.01$ and $D = 0.001$. The other parameters are set as follows: grid 200×24, $b_0 = 32000$, $b_1 = 3200$, $a_0 = -16000$, $a_1 = 16000$, $\sigma_{0x} = \sigma_{1x} = 5$, $\sigma_{0y} = \sigma_{1y} = 50$, $X_0 = X_1 = 50$, $Y_0 = Y_1 = 12$. The total number of particles is $N_0 = 148{,}831{,}768$, $N_1 = 20{,}136{,}193$.

4.3.1. *Longitudinal Dispersion Coefficient*

As a quantitative test, we measure the longitudinal dispersion coefficient D_L defined as the variance of the species density along the streaming direction x, $D_L(t) = \sum_l (x_1 - \bar{x})^2 p_l / 2t$

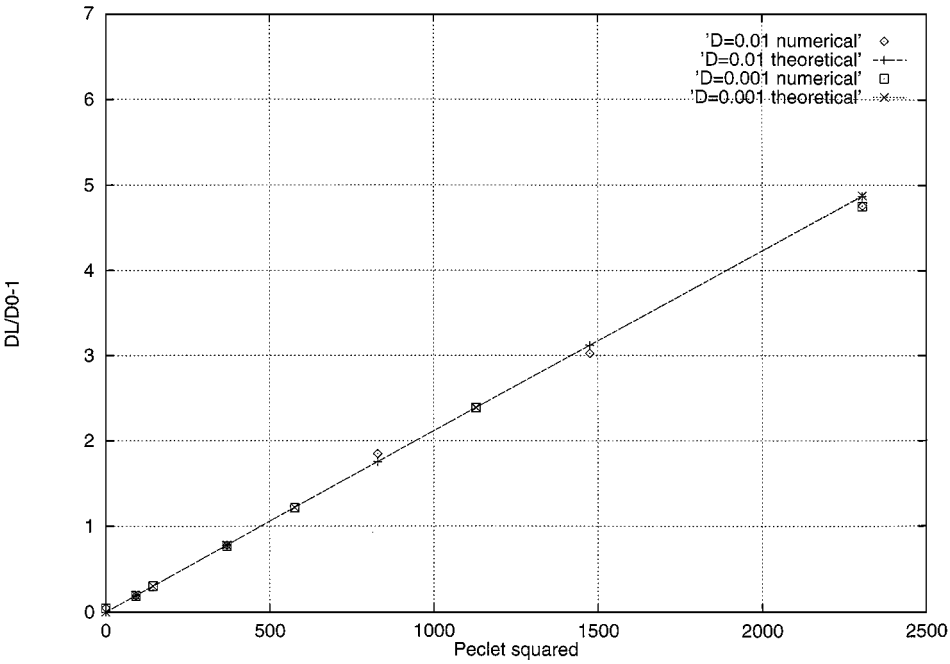


FIG. 12. Relative dispersion δ_L as a function of the Peclet number, for $D = 0.01$ and $D = 0.001$. Due to the definition of $\delta_L \equiv D_L/D - 1$, the theoretical lines are coincident.

where, as usual, $\overline{x}$ is the mean position along x. Current literature (e.g., see [14]) yields

$$D_L = D\left(1 + \frac{Pe_0^2}{470}\right), \tag{23}$$

where $Pe_0 = U_0 H/D$ is the Peclet number computed with the maximum flow speed U_0 and $H = 24$ is the channel width. The quadratic factor in Eq. (23) reflects the effect of hydrodynamic stretch produced by the sheared flow configuration. This term is dominant for $Pe_0 > Pe_c = \sqrt{470} \sim 23$, whereas below this value molecular diffusion prevails.

The numerical comparison is best organised in terms of the relative dispersion $\delta_L = (D_L - D)/D$, which, according to (23) should scale like $Pe_0^2/470$ independent of the molecular diffusivity D. The results are summarized in Fig. 12, where δ_L as obtained by simulations is compared with the analytical results (continuous lines) for the two values of molecular diffusion $D = 0.01$ and $D = 0.001$. Due to the normalization, the values of δ_L as a function of *Pe* should collapse into a single curve, which is exactly what Fig. 12 shows.

4.3.2. *Spatial Distribution of the Species Density*

Next, we examine the spatial distribution of the density field. Since we don't know of any analytical solution, we compare our method with a standard floating point calculation based upon an explicit finite volume method.

The finite-volume code is based on the classical Patankar scheme [15]. In a nutshell, the method is based on an explicit update of the generic scalar unknown Φ_c located in the center of cell Ω_c, via the advective (suffix A) and diffusive (suffix D) fluxes crossing the

four boundaries of the cell Ω_c,

$$\Phi_c(t) = \frac{\Delta t}{V_c} \sum_{n=0,4} \left(C_n^A + C_n^D\right) \Phi_n(t - \Delta t), \tag{24}$$

where $V_c = \Delta x \Delta y$ is the cell volume. Here Φ_n denotes the scalar value in the center of the cell Ω_n sharing the nth boundary with Ω_c. The subscript 0 labels the standing component of the flux, the one that does not leave the cell Ω_c in the time lapse Δt. As an example, the coefficients of advective and diffusive fluxes crossing the east boundary (subscript 1) are given by $C_1^A = u_1 \Delta y$ and $C_1^D = D \Delta y / \Delta x$.

This method incorporates a (non-linearly) Peclet dependent flux limiter protecting against negative densities induced by high local Peclet numbers. The specific form adopted in this work is as follows: $F^A \to \tilde{F}^A = \text{Max}[0, F^A]$ and $D^A \to \tilde{D}^A = D^a \text{ Max}[0, (1 - (Pe/10)^5)]$. Interestingly enough, this flux limiter was independently designed to cut-off at $Pe > 10$, which is pretty close to the limiting value emerging from our simulations (see previous section). Positivity does not come for free, however, but only at the price of a non-zero numerical diffusivity. The effects of such a diffusivity are well visible in the numerical simulations, as we shall detail in what follows. The simulations were run on a 200×50 grid with the same set of parameters given earlier in this subsection. The flow speed is $U_0 = 0.1$ and the diffusivity is $D = 0.001$.

The initial condition for both species is shown in Figs. 13–14, reporting density contours normalized to the background value of the majority species, b_0.

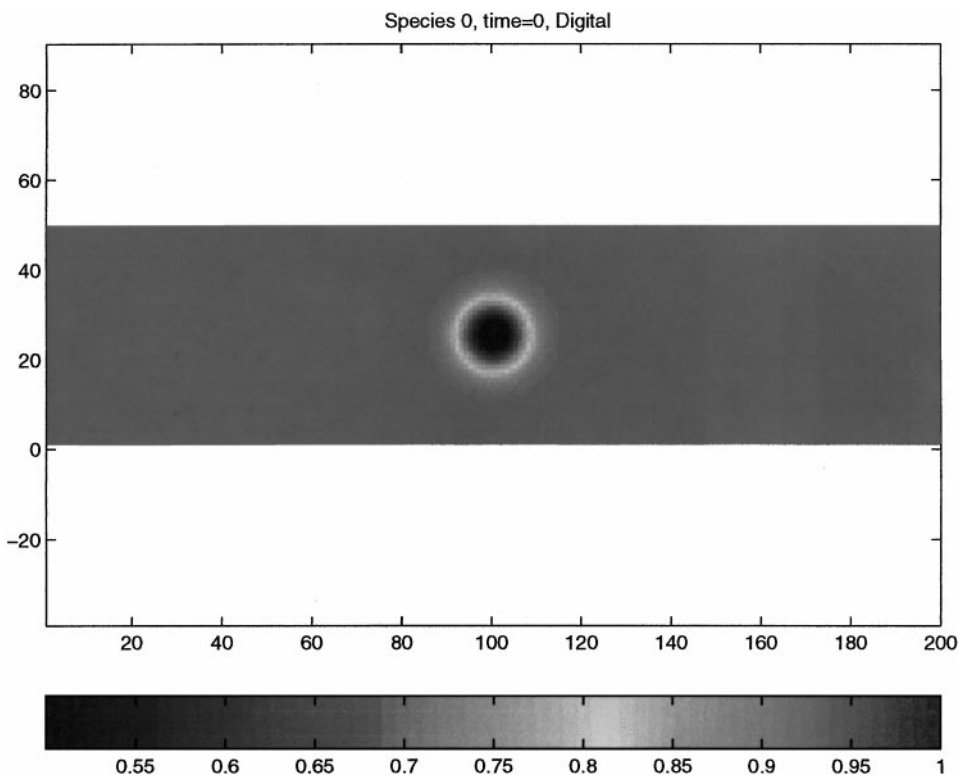


FIG. 13. Digital density of species zero at $t = 0$ (initial condition). Here and throughout, digital indicates the results obtained with the interger Lax–Wendroff scheme.

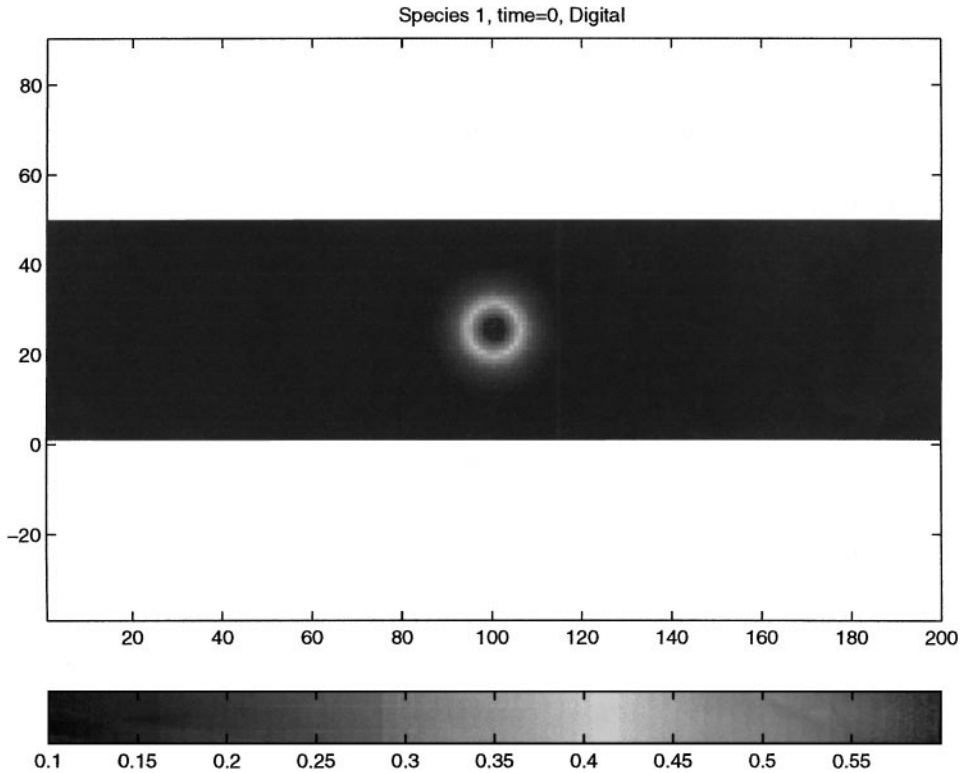


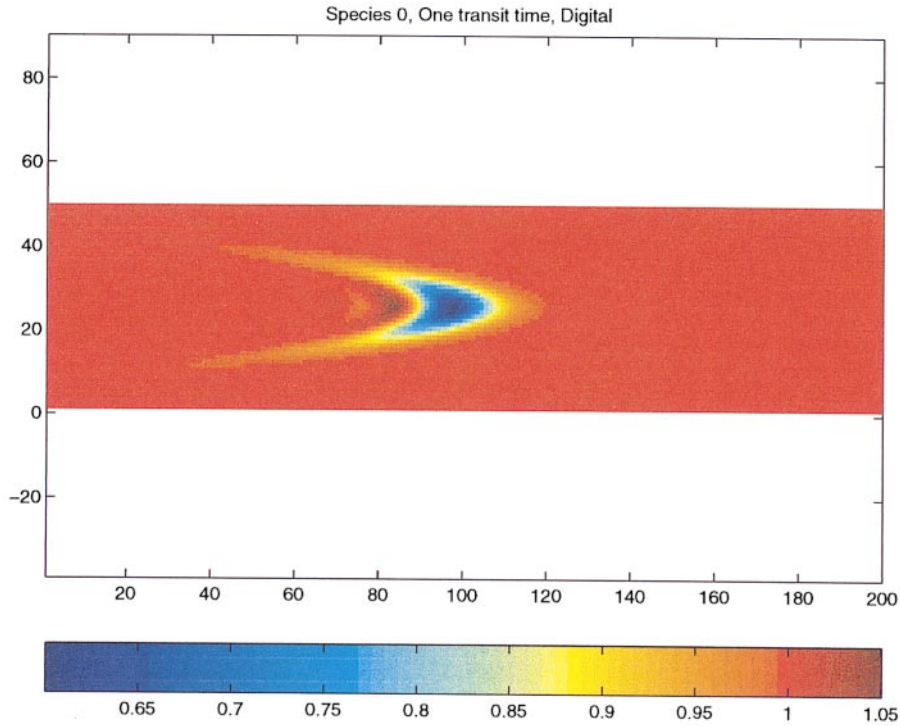
FIG. 14. Digital density of species one at $t = 0$ (initial condition).

Under the (negligible) effects of molecular diffusion and hydrodynamic shear-driven stretch, as time evolves, the initial circles are turned into a "boomerang shape." This effect is well visible in Figs. 15–16, which show the species densities after one recirculation time ($L/U_0 = 2000$ time steps, $L = 200$ being the channel length). Note the change in the color scale (the normalization staying the same throughout) which is introduced for merely graphical purposes. Small ripples due to numerical dispersion are again visible.

After 2000 steps (one transit time), molecular diffusion ($D = 0.001$) should have little effect as compared to hydrodynamic stretch. Indeed the effect of hydrodynamic stretch is very apparent and looks quite similar for both the integer and floating-point calculation. Diffusion is however higher for the finite-volume computation, as witnessed by the fact that the inner core of the density pit (species zero) and bump (species one) is significantly less diffuse for the integer calculation than for its floating-point counterpart (see Figs. 17–18).

Finally, we performed a very long time integration of 20,000 time steps corresponding to 10 longitudinal transit times. Here, we expect physical diffusion to produce a visible spreading effect in the integer simulation since $Dt/\sigma^2 \sim 1$. Such a diffusive effect is indeed visible from Figs. 19–20 (note the change in scale with respect to the previous figures). We observe that, unlike the case of uniform convection, this simulation did not develop any significant Gibbs phenomena, as witnessed by the fact that no action from the underflow control algorithm was ever requested in the course of the long simulation.

Before concluding, we give one word on computational performance. The integer and floating-point code take about 40 and 15 μs/step/site respectively on a Sparc1 computer.



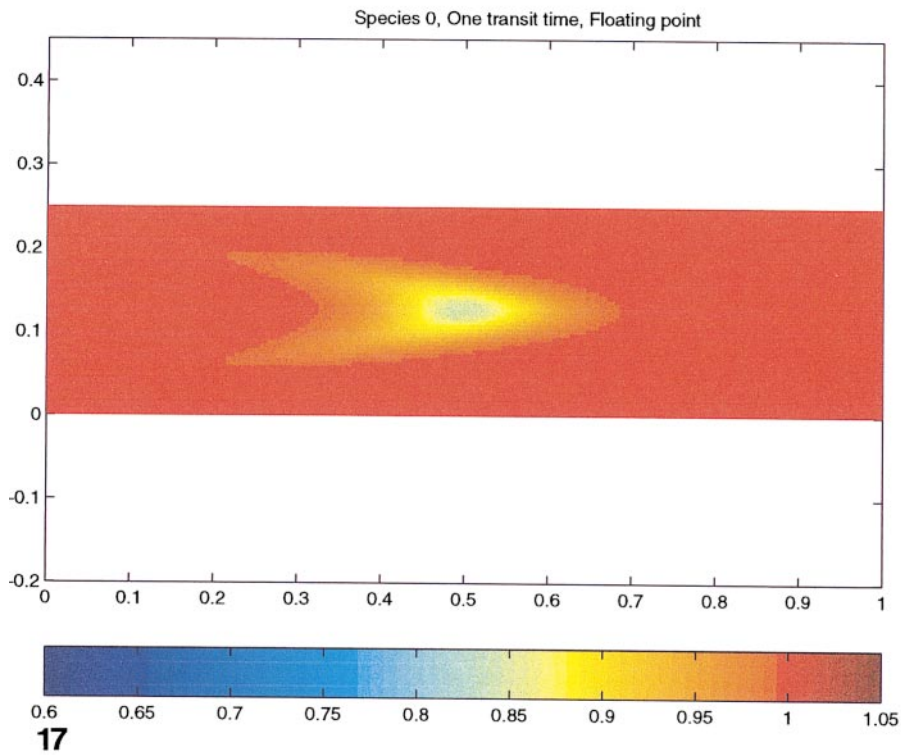


FIG. 15. Digital density of species zero at $t = 2000$ (one transit time), for $U = 0.1$, $D = 0.001$.

FIG. 17. Floating-point density of species zero at $t = 2000$ (one transit time), with the same parameters as Figs. 15–16.

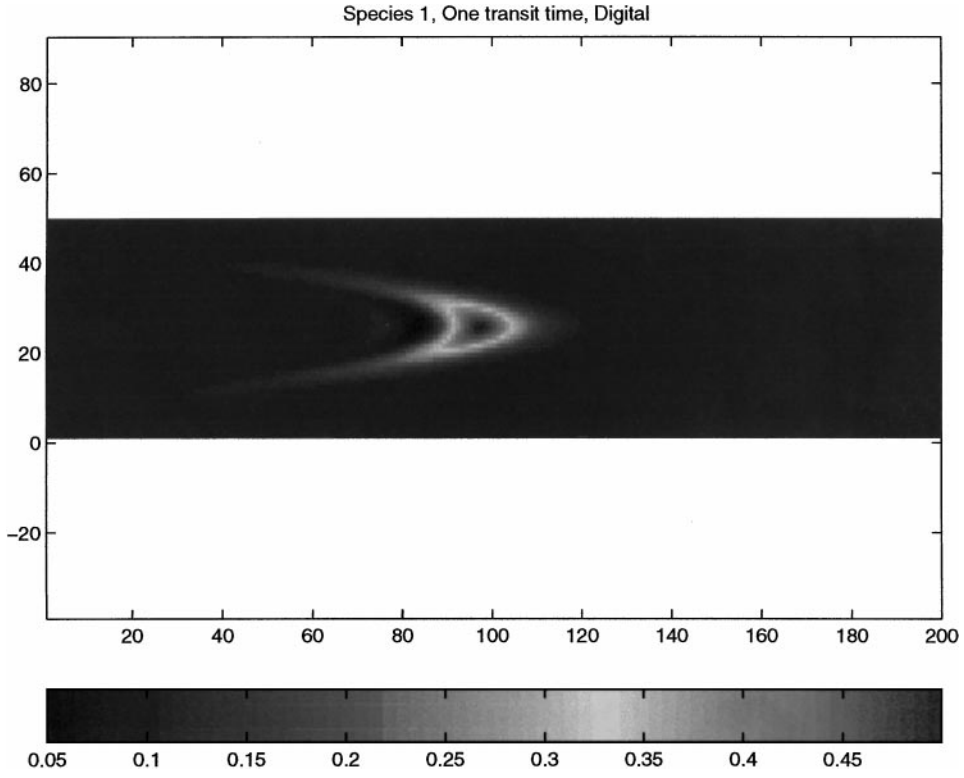


FIG. 16. Digital density of species one at $t = 2000$ (one transit time), for $U = 0.1$, $D = 0.001$.

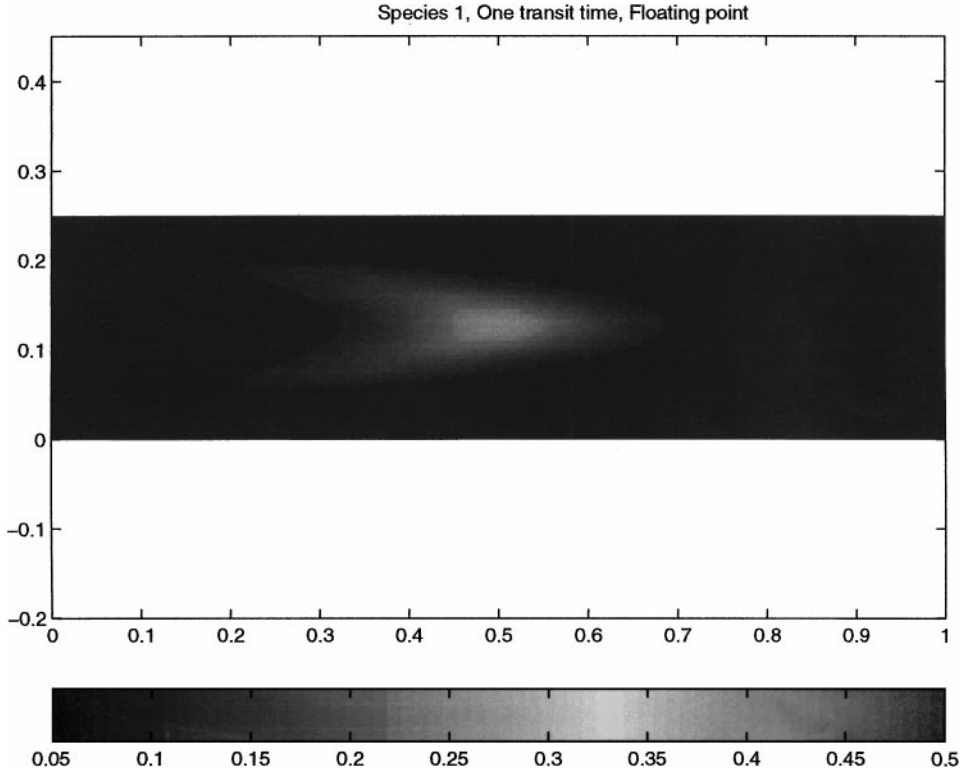


FIG. 18. Floating-point density of species one at $t = 2000$ (one transit time), with the same parameters as Figs. 15–16.

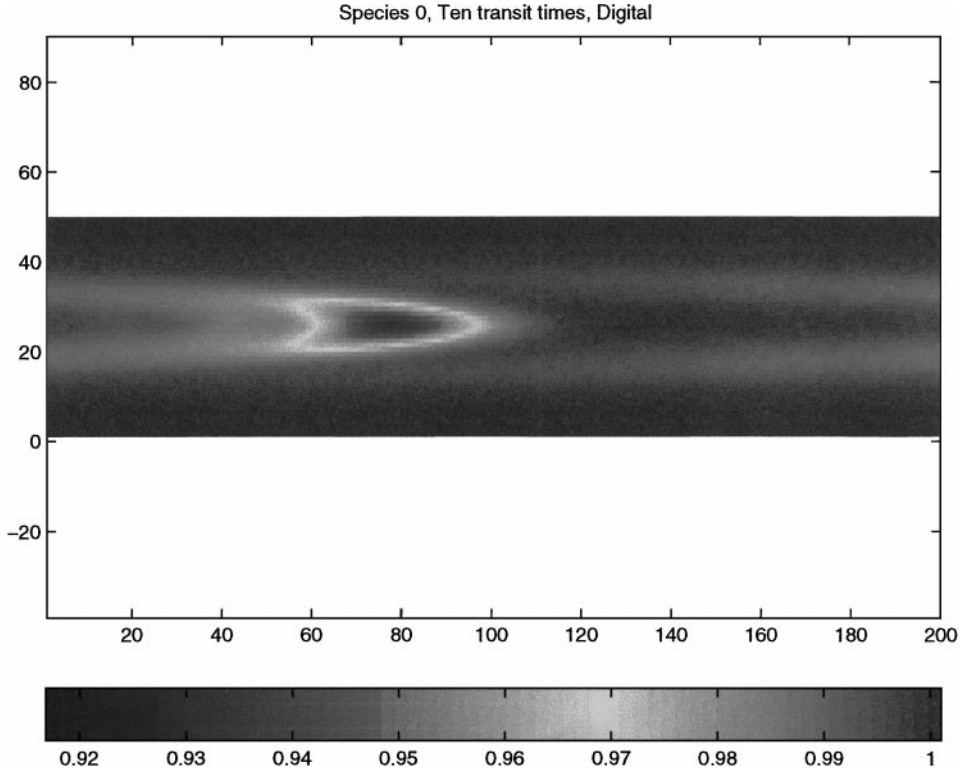


FIG. 19. Digital density of species zero at $t = 20000$ (ten transit times) for $U = 0.1$, $D = 0.001$.

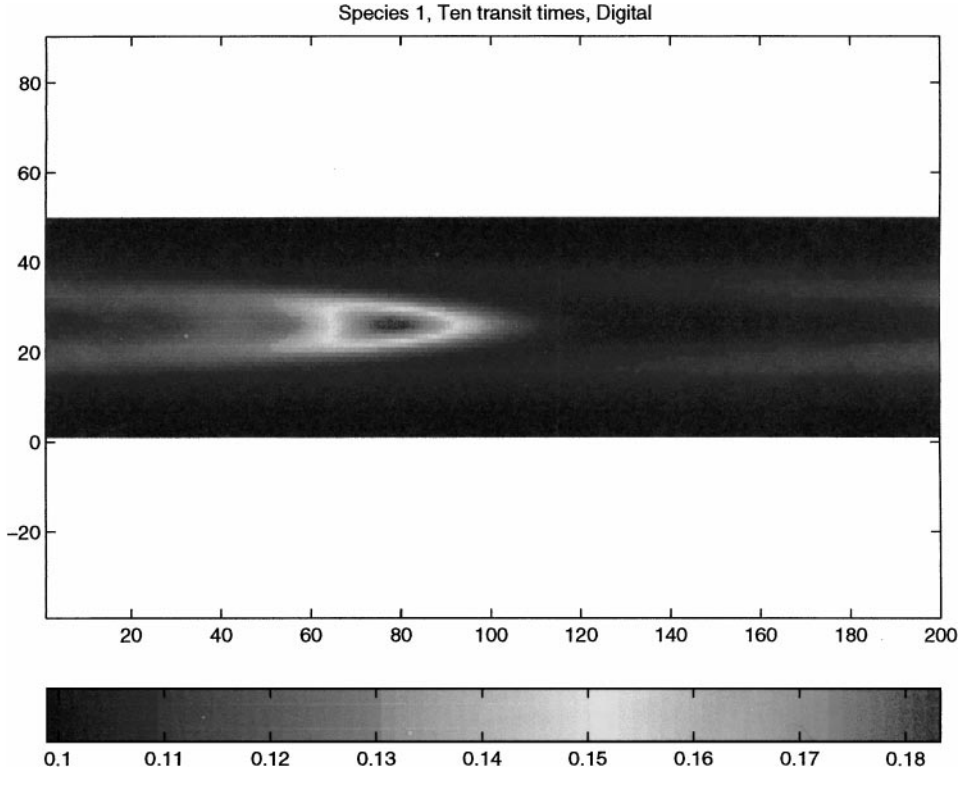


FIG. 20. Digital density of species one at $t = 20000$ (ten transit times) for $U = 0.1$, $D = 0.001$.

These figures are purely qualitative, for more than one reason. First, the language, C for the integer and Fortran for the floating-point codes, is not the same; second, no specific optimization efforts have been undertaken on either codes. Judging on mere operation count, it is reasonable to expect that the integer and floating-point versions of the same Lax–Wendroff algorithm would deliver essentially the same performance on a general purpose computer.

This set of results provides satisfactory evidence that the species advection-diffusion process is handled properly by the present integer Lax–Wendroff scheme. As already mentioned, while numerical diffusivity appears fairly well mastered, control of numerical dispersion calls for further refinements. This will be the subject of future research.

5. CONCLUSIONS

In this paper, we have presented an integer formulation of the Lax–Wendroff finite-difference scheme for the transport of multicomponent flows. The scheme satisfies exact integer particle conservation laws for each chemical component. This feature is highly desirable in view of future applications involving chemical reactions with disparate time scales, and also to perform long time simulations. Though particle density distribution functions for each chemical component are all represented by integers, their ensemble averaged property obeys the standard passive-scalar multi-component transport equation in the large scale limit.

In the context of lattice gas and lattice Boltzmann simulations, the integer algorithm developed here, where only *one* scalar variable is needed per species, achieves a significant memory savings over the traditional method of dealing with multiple species in lattice methods, which required the storage of d different density components per each additional species, d being the number of discrete directions on the underlying lattice.

Theoretical analysis shows that the algorithm yields negligible numerical diffusion at relevant orders. It is also shown that it is stable under the fairly mild conditions we have tested to date. Some benchmark cases have been analyzed with this algorithm, and the accuracy compares favorably with standard floating-point numerical algorithms. In fact, we find significantly reduced numerical diffusion for the integer scheme compared with the floating-point code, although we make no claims as to the relation of this latter code to current state-of-the-art. The algorithm is a simple and completely parallel numerical tool that can be used to perform practical and efficient large-scale computations.

Work left for the future includes an additional procedure explicitly enforcing positivity of the particle distribution functions at all times. Such a procedure must be constructed via local dynamics in order to preserve the parallel feature of the algorithm, as well as to prevent numerical diffusion effects. Most importantly, the present scheme needs be coupled with a dynamic fluid solver so as to assess its robustness in situations where non-negligible density gradients arise.

Investigations along these lines are under way.

APPENDIX: STABILITY ANALYSIS

For the sake of clarity and in order to focus attention on the essential physics, we analyze the basic stability properties of the extended Lax system in the $2d$ single component case at the critical situation with zero molecular diffusivity (i.e., $D = 0$). In addition, to make the

conclusion unambiguous, we present the analysis by ignoring flow field variations, so that the system is essentially linear.

In $2d$, Eq. (9) has the form

$$\begin{aligned}\rho_s(x,y,t+\Delta t) = &\left(\frac{1+2\tilde{U}_x}{4}\right)\rho_s(x-\Delta_x,y,t) + \left(\frac{1-2\tilde{U}_x}{4}\right)\rho_s(x+\Delta_x,y,t)\\ &+\left(\frac{1+2\tilde{U}_y}{4}\right)\rho_s(x,y-\Delta_y,t) + \left(\frac{1-2\tilde{U}_y}{4}\right)\rho_s(x,y+\Delta_y,t)\\ &+\mathcal{D}_x[\rho_s(x+\Delta_x,y,t) - 2\rho_s(x,y,t) + \rho_s(x-\Delta_x,y,t)]\\ &+\mathcal{D}_y[\rho_s(x,y+\Delta_y,t) - 2\rho_s(x,y,t) + \rho_s(x,y-\Delta_y,t)]\\ &-\frac{1}{4}\tilde{U}_x\tilde{U}_y[\rho_s(x-\Delta_x,y+\Delta_y,t) - \rho_s(x-\Delta_x,y-\Delta_y,t)\\ &-\rho_s(x+\Delta_x,y+\Delta_y,t) + \rho_s(x+\Delta_x,y-\Delta_y,t)], \end{aligned} \tag{25}$$

where now the diffusivities in the cartesian directions are

$$\mathcal{D}_\alpha = -\frac{1}{4}\left(1-2\tilde{U}_\alpha^2\right).$$

Fourier transforming the variables from $(\mathbf{x}, t)$ to $(\mathbf{k}, \omega)$, we obtain

$$\begin{aligned}e^{-i\omega} = &\left(\frac{1+2\tilde{U}_x}{4}\right)e^{-ik_x} + \left(\frac{1-2\tilde{U}_x}{4}\right)e^{ik_x} + \left(\frac{1+2\tilde{U}_y}{4}\right)e^{-ik_y} + \left(\frac{1-2\tilde{U}_y}{4}\right)e^{ik_y}\\ &+\mathcal{D}_x[e^{ik_x} - 2 + e^{-ik_x}] + \mathcal{D}_y[e^{ik_y} - 2 + e^{-ik_y}]\\ &+\frac{1}{4}\tilde{U}_x\tilde{U}_y(e^{ik_x} - e^{-ik_x})(e^{ik_y} - e^{-ik_y}). \end{aligned} \tag{26}$$

Separating the real and imaginary parts, $\omega = \Omega + i\gamma$, the above equation results in

$$\begin{aligned}e^{\gamma}\cos(\Omega) = &\frac{1}{2}(\cos(k_x) + \cos(k_y)) + 2\mathcal{D}_x(\cos(k_x) - 1)\\ &+2\mathcal{D}_y(\cos(k_y) - 1) - \tilde{U}_x\tilde{U}_y\sin(k_x)\sin(k_y)\\ e^{\gamma}\sin(\Omega) = &\tilde{U}_x\sin(k_x) + \tilde{U}_y\sin(k_y). \end{aligned} \tag{27}$$

The equations can be expressed in a more compact form with new variables,

$$\delta \equiv (\delta_x, \delta_y) \equiv \left(\frac{1}{2} + 2\mathcal{D}_x, \frac{1}{2} + 2\mathcal{D}_y\right), \tag{28}$$

$$\mathbf{c} \equiv (c_x, c_y) \equiv (\cos(k_x), \cos(k_y)), \tag{29}$$

$$\mathbf{s} \equiv (s_x, s_y) \equiv (\sin(k_x), \sin(k_y)), \tag{30}$$

and $\tilde{\mathbf{U}} \equiv (\tilde{U}_x, \tilde{U}_y)$. Hence we can rewrite (27) as

$$e^{\gamma}\cos(\Omega) = \delta\cdot(\mathbf{c} - \mathbf{1}) + 1 - \tilde{U}_x\tilde{U}_y s_x s_y \tag{31}$$

$$e^{\gamma}\sin(\Omega) = \tilde{\mathbf{U}}\cdot \mathrm{s}, \tag{32}$$

where $\mathbf{1} \equiv (1, 1)$. In order to investigate the stability properties of the system, we take the square of (31) and (32) and add them together. We then arrive at the following equation governing the imaginary part of the frequency,

$$e^{2\gamma} = [\delta \cdot (\mathbf{c} - \mathbf{1}) + 1 - \tilde{U}_x \tilde{U}_y s_x s_y]^2 + (\tilde{\mathrm{U}} \cdot s)^2. \tag{33}$$

Or equivalently,

$$\begin{aligned} e^{2\gamma} - 1 = {} & 2\delta \cdot (\mathbf{c} - \mathbf{1}) + [\delta \cdot (\mathbf{c} - \mathbf{1})]^2 - 2\tilde{U}_x \tilde{U}_y s_x s_y [\delta \cdot (\mathbf{c} - \mathbf{1})] \\ & + \tilde{U}_x^2 \tilde{U}_y^2 s_x^2 s_y^2 + \tilde{U}_x^2 s_x^2 + \tilde{U}_y^2 s_y^2. \end{aligned} \tag{34}$$

Stability is guaranteed if $\gamma \leq 0$, or equivalently $e^{2\gamma} - 1 \leq 0$.

It is difficult to gain insight from this expression in general. Instead, we examine (34) in order of wave number $\mathbf{k}$. Because

$$\begin{aligned} \mathbf{c} - \mathbf{1} &= \left(-\frac{k_x^2}{2} + \frac{k_x^4}{24} + \cdots, -\frac{k_y^2}{2} + \frac{k_y^4}{24} + \cdots \right), \\ \mathbf{s} &= \left(k_x - \frac{k_x^3}{6} + \cdots, k_y - \frac{k_y^3}{6} + \cdots \right), \end{aligned}$$

to $O(k^2)$ we have

$$e^{2\gamma} - 1 = \left(\tilde{U}_x^2 - \delta_x\right) k_x^2 + \left(\tilde{U}_y^2 - \delta_y\right) k_y^2. \tag{35}$$

Recall that $\mathcal{D}_\alpha = -\frac{1}{4}(1 - 2\tilde{U}_\alpha^2)$, so we have $\delta_\alpha = \tilde{U}_\alpha^2$ (see Eq. (28)). Hence the right-hand side of (35) vanishes, which indicates there is no second order diffusion effect in the extended Lax scheme, a desired result.

Next we look at the order of $O(k^4)$. Stability is significantly enhanced if contributions from this order provide a negative contribution to the right-hand side of (34).

Using the result of vanishing $O(k^2)$ from (35), and after some straightforward algebra, we have

$$\begin{aligned} e^{2\gamma} - 1 = {} & -\frac{1}{4}\tilde{U}_x^2 \left(1 - \tilde{U}_x^2\right) k_x^4 - -\frac{1}{4}\tilde{U}_y^2 \left(1 - \tilde{U}_y^2\right) k_y^4 + \frac{3}{2}\tilde{U}_x^2 \tilde{U}_y^2 k_x^2 k_y^2 \\ & + \tilde{U}_x \tilde{U}_y k_x k_y \left(\tilde{U}_x^2 k_x^2 + \tilde{U}_y^2 k_y^2\right). \end{aligned} \tag{36}$$

If we use the following readily deducible inequalities,

$$\begin{aligned} \tilde{U}_x \tilde{U}_y k_x k_y &\leq \frac{1}{2}\left(\tilde{U}_x^2 k_x^2 + \tilde{U}_y^2 k_y^2\right), \\ \tilde{U}_x^2 \tilde{U}_y^2 k_x^2 k_y^2 &\leq \frac{1}{2}\left(\tilde{U}_x^4 k_x^4 + \tilde{U}_y^4 k_y^4\right), \end{aligned}$$

we obtain

$$e^{2\gamma} - 1 \leq -\frac{1}{4}\tilde{U}_x^2 \left(1 - 8\tilde{U}_x^2\right) k_x^4 - \frac{1}{4}\tilde{U}_y^2 \left(1 - 8\tilde{U}_y^2\right) k_y^4. \tag{37}$$

Therefore, stability is indicated at $O(k^4)$ if the fluid velocity component obeys the condition

$$|\tilde{U}_\alpha| \leq \frac{1}{2\sqrt{2}} \approx 0.35. \tag{38}$$

ACKNOWLEDGMENT

S.S. acknowledges EXA Corporation for their kind hospitality and financial support during several stays in Boston.

REFERENCES

1. U. Frisch, B. Hasslacher, and Y. Pomeau, Lattice gas automata for the Navier–Stokes equation, *Phys. Rev. Lett.* **56**, 1505 (1986).
2. U. Frisch, D. d'Humières, B. Hasslacher, P. Lallemand, Y. Pomeau, and J.-P. Rivet, Lattice gas hydrodynamics in two and three dimensions, *Complex Systems* **1**, 649 (1987).
3. S. Wolfram, Cellular automaton fluids. I. Basic theory, *J. Stat. Phys.* **45**, 471 (1986).
4. K. Molvig, P. Donis, J. Myczkowski, and G. Vichniac, Removing the discreteness artifacts in three-dimensional lattice gas fluids, in *Discrete Kinetic Theory, Lattice Gas Dynamics and Foundations of Hydrodynamics*, edited by R. Monaco (World Scientific, Singapore, 1988).
5. G. D. Doolen, Ed., Lattice gas methods for PDE's: Theory, applications and hardware, *Phys. D* **47** (1993).
6. F. Higuera, S. Succi, and R. Benzi, Lattice gas dynamics with enhanced collisions, *Europhys. Lett.* **9**, 345 (1989); R. Benzi, S. Succi, and M. Vergassola, The lattice Boltzmann equation: Theory and applications, *Phys. Rep.* **222**, 145 (1992).
7. S. Chen, H. Chen, D. Martinez, and W. Matthaeus, Lattice Boltzmann model for simulation of magnetohydrodynamics, *Phys. Rev. Lett.* **67**, 3776 (1991); Y. Qian, D. d'Humières, and P. Lallemand, Lattice BGK models for the Navier–Stokes equations, *Europhys. Lett.* **17**, 479 (1992).
8. H. Chen, C. Teixeira, and K. Molvig, Digital physics approach to computational fluid dynamics: Some basic theoretical features, *Int. J. Modern Phys.* **8**, 675 (1997).
9. E. Oran and J. Boris, *Numerical Simulation of Reactive Flows* (Elsevier, New York, 1987).
10. C. Teixeira, Incorporating turbulence models into the Lattice Boltzmann method, *Int. J. Modern Phys.* **9**, 1159 (1998).
11. W. F. Ames, *Numerical Methods for Partial Differential Equations* (Academic Press, New York, 1977).
12. D. Potter, *An Introduction to Computational Physics* (Wiley, London, 1976).
13. C. Hirsch, *Numerical Computation of Internal and External Flows* (Wiley, New York 1988), Vol. I.
14. G. I. Taylor, Dispersion of soluble matter in solvent flowing slowly through a tube, *Proc. R. Soc. London Ser. A* **219**, 186 (1953).
15. S. V. Patankar, *Numerical Heat Transfer and Fluid Flow* (Hemisphere, Washington, DC/New York, 1989).