Dual-Mesh Characteristics for Particle-Mesh Methods for Convection-Dominated Diffusion Problems

Seongjai Kim* and David P. Chock†

Abstract

The particle-mesh method (PMM) is a powerful computational tool for the simulation of convection-dominated diffusion flows. The method introduces *computational particles* each of which is given a finite size and represents a large number of physical particles with similar properties. The convection part of the flow can be solved by moving the computational particles along the characteristics, while the diffusion part is carried out by utilizing a heat solver on a regular mesh. However, the method in practical applications shows the so-called *ringing instability*, an amplitude fluctuation/noise in the computed solution. In this article, we suggest a new numerical technique of particle movement, called the *dual-mesh characteristics* (DMC) of which the second mesh is formed by tracking back the cells along the characteristics. The particle movement is carried out by interpreting the particle positions (in the former time step) in terms of the multilinear coordinates of the second mesh. Strategies for the average velocity and interpolations are also suggested to move the computational particles accurately with a minimum numerical dissipation. The resulting algorithm, *PMM-DMC*, turns out to be non-oscillatory, of negligible dissipation, and more efficient than the conventional schemes. Numerical results are shown to demonstrate its accuracy and efficiency.

Key words. Particle-mesh method, particle-in-cell (PIC) method, operator splitting technique, method of characteristics, dual-mesh characteristics.

*Department of Mathematics, University of Kentucky, Lexington, Kentucky 40506-0027 USA Email: skim@ms.uky.edu

†Chemistry Department, Ford Research Laboratory, 20000 Rotunda Drive, P.O.Box 2053, MD-3083, Dearborn, MI 48121-2053 Email: dchock@ford.com

Contents

1. Introduction

Let $\Omega \subset \mathbb{R}^2$ be a bounded domain with its boundary $\Gamma = \partial\Omega$ and $J = (0, T]$ the time interval, $T > 0$. The model to be considered is the following convection-diffusion flow problem: Find the concentration c such that

$$\begin{array}{lll} \text{(a)} & \dfrac{\partial c}{\partial t} + \nabla \cdot (\mathbf{v}c) - \nabla \cdot (D\nabla c) = f, & (\mathbf{x}, t) \in \Omega \times J, \\ \text{(b)} & (D\nabla c) \cdot \nu = 0, & (\mathbf{x}, t) \in \Gamma \times J, \\ \text{(c)} & c = c_0, & \mathbf{x} \in \Omega, \;\; t = 0, \end{array} \tag{1.1}$$

where $\mathbf{v}$ is the fluid velocity, ν is the outward normal from Ω, and f denotes the source/sink. The diffusion-dispersion tensor D is symmetric and positive definite:

$$D^T = D; \quad D_*|\mathbf{y}|^2 \le \mathbf{y}^T D(\mathbf{x})\mathbf{y} \le D^*|\mathbf{y}|^2, \quad \forall\, \mathbf{x} \in \Omega, \;\; \forall\, \mathbf{y} \in \mathbb{R}^2,$$

for some nonnegative constants D_* and D^*. The velocity either can be obtained by solving another equation such as the pressure equation or is given from experimental acquisition. The flow is assumed impressible:

$$\nabla \cdot \mathbf{v} = 0. \tag{1.2}$$

For the problem of the form (1.1), the convection is often dominant and the problem becomes difficult to solve, in particular, when the solution is not smooth. Upwinding methods cause a degree of numerical dissipation, while higher-order methods such as central finite difference (FD) and Galerkin finite element (FE) methods often provide nonphysical oscillation (dispersion) to the solution. That is, the conventional methods are lacking in either stability or accuracy.

Particle-in-cell (PIC) methods [9] have emerged as a powerful computational tool for the numerical solution of convection-dominated flows. The basic idea of the method is to introduce *computational particles* each of which is given a finite size and represents a large number of physical particles with similar properties. The PIC method can show distinct advantages over other methods, when the extra information carried by particles is properly used; although the converse is true when it is not properly used [10]. In fact, for realistic situations, PIC often introduces a level of noise or so-called *ringing instability* [2, 3] which is the fluctuation in the numerical solution, due to the statistic nature of the method that the computational model is composed of far fewer particles than the physical model. Noise suppression has been the major research field for the method [2, 3, 4, 10].

In this article, we study the *particle-mesh method* (PMM) [10], one of PIC methods, for the simulation of convection-dominated diffusion flows of the form (1.1). In

PMM, the convection part is integrated by advancing particles along the characteristics (Lagrangian), while the diffusion part is solved on a regular mesh (Eulerian). An appropriate computation order in a timestep can be particle movement, particle-mesh evaluation, diffusion solve, and mesh-particle evaluation. Thus the introduction of interpolation schemes is necessary for the evaluations between the particle and mesh values. Various basis functions for the interpolations, along with strategies for controlling the number of particles, have been studied to efficiently suppress the fluctuation [2, 3, 19].

However, it has been observed from various numerical experiments that basis functions that can suppress the nonphysical fluctuations can also easily smear out physical sharp-fronts. Such numerical dissipation can appear as a severe drawback of PMM. Here the question is: *how can we reduce the fluctuation efficiently with no or negligible numerical dissipation?* As an answer for the question, we first suggest a new strategy of particle movement, called the *dual-mesh characteristics* (DMC) which incorporates a second mesh that is formed by tracking the grid points (of the original mesh) back through the characteristics. In DMC, the particle movement is carried out by interpreting the particle locations (on the former time step) in terms of the multilinear coordinates of the second mesh. Thus DMC advances the particles in the slab-by-slab manner, parallel to the characteristic direction. An analytic formula for the average velocity is incorporated to move the particles more accurately.

To minimize the numerical dissipation from interpolations (particle-mesh and mesh-particle evaluations), the basis functions of the interpolations should have their supports as small as possible. Also the combined operation of particle-mesh evaluation, diffusion solve, and mesh-particle evaluation should not introduce an observable numerical dissipation for a small physical diffusion. To meet the requirements, we allow the interpolation schemes to have a minimum smooth property such as based on the Dirac delta function, and let the mesh-particle evaluation correct the *change-in-diffusion* only.

It has been numerically tested that the resulting algorithm (*PMM-DMC*), incorporating the average velocity and the change-in-diffusion interpolation, integrates the solution with no fluctuation and negligible dissipation. Furthermore, it turns out to be more efficient than conventional algorithms based on the individual particle movement.

Various numerical methods different from the one to be considered here, and having reasonable accuracy and stability properties, have been suggested to solve convection-dominated diffusion flow problems. Among others, interesting methods are the streamline diffusion methods [12, 13, 14, 22, 23] and the discontinuous Galerkin methods [11, 16, 20, 25]. However, the approximate solutions of these relatively new methods may still exhibit either over- or under-shoots, when the solution is nonsmooth.

In §2, we briefly review operator splitting techniques and methods of characteristics. A solution technique for the diffusion part is considered. Section 3 presents the scope of PMM. Besides considering ringing instability, a strategy of reducing numerical dissipation is discussed. In §4, we suggest DMC which suppresses both the ringing instability and numerical dissipation of PMM to be negligible. In §5, we suggest a strategy for an accurate average velocity. The importance of incorporating an accurate velocity can never be over-emphasized for methods of characteristics. Section 6 presents numerical results to show accuracy and efficiency of PMM-DMC. We add some discussion in §7. The last section includes conclusions.

2. Preliminaries

In this section, we review operator splitting techniques and methods of characteristics. We begin with partitioning J into

$$0 = t^0 < t^1 < \cdots < t^{N_t} = T, \tag{2.1}$$

for a positive integer N_t. Define $\Delta t^n = t^n - t^{n-1}$, and $g^n(\mathbf{x}) = g(\mathbf{x}, t^n)$ for a function g of independent variables $(\mathbf{x}, t) \in \Omega \times J$.

Solving the problem (1.1) requires the integration of the operator for each of the space-time slices

$$\mathcal{S}^n := \Omega \times (t^{n-1}, t^n], \quad n = 1, \cdots, N_t,$$

by applying numerical schemes either directly to (1.1) or to fractional steps of an operator splitting technique. Operator splitting is a common practice for modern large-scale computations, in particular, for convection-diffusion problems.

2.1. Operator splitting

The problem (1.1) can be solved by integrating along the space-time slices $\mathcal{S}^n$. In each slice, the operator (1.1) is split into two substeps: the transport part and the diffusion part. For a purpose of algorithm development for convection-diffusion problems, we will ignore the source/sink term, i.e., set $f \equiv 0$.

The transport part in $\mathcal{S}^n$ requires solving the hyperbolic problem: Find c^* (at $t = t^n$) satisfying

$$\begin{aligned} &\text{(a)} \quad \frac{\partial c}{\partial t} + \nabla \cdot (\mathbf{v} c) = 0, \qquad (\mathbf{x}, t) \in \mathcal{S}^n, \\ &\text{(b)} \quad c(\mathbf{x}, t^{n-1}) = c^{n-1}(\mathbf{x}), \quad \mathbf{x} \in \Omega. \end{aligned} \tag{2.2}$$

The problem is not easy to solve, in particular, when the exact solution is nonsmooth. Standard FD and FE methods can either introduce nonphysical oscillations (numerical dispersion) or smear out the solution (numerical dissipation). In PMM, (2.2) can be efficiently solved by moving the computational particles along the characteristics.

The diffusion step in $\mathcal{S}^n$ requires solving the problem of the form

$$\begin{aligned} &\text{(a)} \quad \frac{\partial c}{\partial t} - \nabla \cdot (D\nabla c) = 0, && (\mathbf{x},t) \in \mathcal{S}^n, \\ &\text{(b)} \quad (D\nabla c)\cdot \nu = 0, && (\mathbf{x},t) \in \Gamma \times (t^{n-1}, t^n], \\ &\text{(c)} \quad c(\mathbf{x}, t^{n-1}) = c^*(\mathbf{x}), && \mathbf{x} \in \Omega. \end{aligned} \tag{2.3}$$

When (2.3) is approximated by numerical schemes, the algebraic system reads

$$A\mathbf{c}^n = \mathbf{b}^n. \tag{2.4}$$

The problem (2.4) is relatively easy to solve, because its algebraic system is symmetric, positive definite, and strictly diagonally dominant. Furthermore, its iterative solvers can utilize good initial values, e.g.,

$$\mathbf{c}^{n,0} = \mathbf{c}^{n-1} \quad \text{or} \quad \mathbf{c}^{n,0} = \Big(1 + \frac{\Delta t^n}{\Delta t^{n-1}}\Big)\mathbf{c}^{n-1} - \frac{\Delta t^n}{\Delta t^{n-1}}\mathbf{c}^{n-2}, \tag{2.5}$$

where $\mathbf{c}^{n-1}$ and $\mathbf{c}^{n-2}$ respectively denote the concentrations at $t = t^{n-1}$ and $t = t^{n-2}$.

The operator splitting method introduces the numerical error of $\mathcal{O}\Big((\Delta t^n)^2\Big)$; it is exact only if the convection and diffusion operators commute each other. For a detailed analysis, see e.g. [18].

2.2. Methods of characteristics

The method of characteristics (streamlines) is a good method for linear convection problems, because it requires solving first the equation for the characteristics and then integrating along the characteristics utilizing numerical methods for ordinary differential equations (ODEs). However, the method cannot be easily used for a mixed hyperbolic-parabolic equation like (1.1). For these more general problems, one usually uses FD or FE methods based on a *fixed mesh* that is in general not adopted to fit the characteristics of the physical problem. The use of a fixed mesh gives methods that are easy to implement but it may cause numerical difficulties. For example, the modified method of characteristics (MMOC) suffers problems of mass conservation and numerical dissipation, while the locally conservative Eulerian-Lagrangian method (LCELM) excessively smear out sharp fronts; for details of these methods, see e.g. [1, 5, 6, 7, 17].

Methods of characteristics for the transport problem (2.2) require the computation of characteristic curves in each of the space-time slices $\mathcal{S}^n$, departing from $\mathbf{x}$ in the

t^n-level and tracking back to $\overline{\mathbf{x}}$ in the t^{n-1}-level. That is, we need to solve the final value problem: Find $\overline{\mathbf{x}} = \gamma(\mathbf{x}; t^{n-1})$ satisfying

$$\begin{aligned} \frac{d\gamma}{dt} &= \mathbf{v}, \quad t \in [t^{n-1}, t^n), \\ \gamma(\mathbf{x}; t^n) &= \mathbf{x}, \end{aligned} \tag{2.6}$$

for points $\mathbf{x}$ in a certain set. Here γ is called a *characteristic curve* or *streamline*. In practice, it is often expensive to solve the final value problem (2.6), in particular, when the velocity is nonsmooth.

The simplest solution for (2.6) can be obtained by applying a step of Euler method:

$$\overline{\mathbf{x}} = \mathbf{x} - \Delta t^n\, \mathbf{v}(\mathbf{x}), \tag{2.7}$$

which has been employed for MMOC [6, 7].

For the Lagrangian component of PMM simulation, we need to solve a similar problem: Find $\mathbf{x}_p^n = \gamma(\mathbf{x}; t^n)$ satisfying

$$\begin{aligned} \frac{d\gamma}{dt} &= \mathbf{v}, \quad t \in (t^{n-1}, t^n], \\ \gamma(\mathbf{x}; t^{n-1}) &= \mathbf{x}_p^{n-1}, \end{aligned} \tag{2.8}$$

for every particle position $\mathbf{x}_p^{n-1}$ in the t^{n-1}-level. Besides the Euler method,

$$\mathbf{x}_p^n = \mathbf{x}_p^{n-1} + \Delta t^n\, \mathbf{v}(\mathbf{x}_p^{n-1}), \tag{2.9}$$

higher-order schemes such as Heun's and Runge-Kutta methods, together with a few fractional substeps, can be applied to solve (2.8) more accurately. The fourth-order Runge-Kutta method (RK4) applied to (2.8) in the interval $[t^{n-1}, t^n]$ reads

$$\begin{aligned} &\mathbf{x}_p^n = \mathbf{x}_p^{n-1} + \frac{1}{6}(K_1 + 2K_2 + 2K_3 + K_4), \\ &\begin{cases} K_1 = \Delta t^n\, \mathbf{v}^{n-1}(\mathbf{x}_p^{n-1}), \\ K_2 = \Delta t^n\, \mathbf{v}^{n-1/2}(\mathbf{x}_p^{n-1} + \frac{1}{2}K_1), \\ K_3 = \Delta t^n\, \mathbf{v}^{n-1/2}(\mathbf{x}_p^{n-1} + \frac{1}{2}K_2), \\ K_4 = \Delta t^n\, \mathbf{v}^{n}(\mathbf{x}_p^{n-1} + K_3), \end{cases} \end{aligned} \tag{2.10}$$

where the superscripts on $\mathbf{v}$ indicate the time levels for time-dependent velocities, with $\mathbf{v}^{n-1/2}(\mathbf{x}) = (\mathbf{v}^{n-1}(\mathbf{x}) + \mathbf{v}^n(\mathbf{x}))/2$.

The velocity is given on grid points or the centroids of cells or cell interfaces. PMM incorporating RK4 requires four interpolations for each component of the velocity for a particle to advance a time step. Thus it is expensive when the numerical model has to introduce a large number of computational particles.

2.3. The diffusion part

Iterative algorithms are preferred to solve large-size algebraic systems. In this subsection, we consider a preconditioned conjugate gradient (PCG) method to solve the algebraic system (2.4). Recall that the matrix A is strictly diagonally-dominant, symmetric, and positive definite.

It is well known that the conjugate gradient (CG) for symmetric positive definite problems converges with the convergence rate

$$\frac{\sqrt{\kappa(A)}-1}{\sqrt{\kappa(A)}+1}, \tag{2.11}$$

where $\kappa(A) := \lambda_{\max}/\lambda_{\min}$ is the *condition number* of matrix A, where $\lambda_{\max}$ and $\lambda_{\min}$ denote the maximum and minimum eigenvalues of A, respectively. (See e.g. [8, §10.2].) If we can find a matrix M which is *spectrally* near to A and easier to invert than A, we may try to apply the CG method to the following system

$$M^{-1}A\mathbf{c} = M^{-1}\mathbf{b}. \tag{2.12}$$

(Here we have dropped the superscript n.) When $\kappa(M^{-1}A) \ll \kappa(A)$ (hopefully, $\kappa(M^{-1}A) \approx 1$), the CG method will converge much faster. Here M is called the *preconditioner*. In practice, we do not have to multiply M^{-1} to the original algebraic system. The CG method applied to the preconditioned system (2.12) reads

$$\begin{aligned}
&\text{Select } \mathbf{c}_0,\ \varepsilon;\\
&\mathbf{r}_0 = \mathbf{b} - A\mathbf{c}_0;\ \ M\mathbf{z}_0 = \mathbf{r}_0;\\
&\mathbf{p}_0 = \mathbf{z}_0;\ \ \text{Set } \rho_0 = \mathbf{z}_0 \cdot \mathbf{r}_0;\\
&\text{Do } k = 0, 1, \cdots\\
&\qquad \alpha_k = \rho_k/\mathbf{p}_k \cdot A\mathbf{p}_k;\\
&\qquad \mathbf{c}_{k+1} = \mathbf{c}_k + \alpha_k \mathbf{p}_k;\\
&\qquad \mathbf{r}_{k+1} = \mathbf{r}_k - \alpha_k A\mathbf{p}_k;\\
&\qquad \text{if } \| \mathbf{r}_{k+1} \|_2 > \varepsilon, \text{continue};\\
&\qquad M\mathbf{z}_{k+1} = \mathbf{r}_{k+1};\\
&\qquad \rho_{k+1} = \mathbf{z}_{k+1} \cdot \mathbf{r}_{k+1};\\
&\qquad \beta_k = \rho_{k+1}/\rho_k;\\
&\qquad \mathbf{p}_{k+1} = \mathbf{z}_{k+1} + \beta_k \mathbf{p}_k;\\
&\text{Enddo}
\end{aligned} \tag{2.13}$$

When the incomplete LU-factorization of level zero (ILU0) is adopted for the preconditioner M, the resulting algorithm is called the CG method preconditioned by the incomplete LU-factorization of level zero (PCG-ILU0), which is popular in scientific computation. Here ILU0 of A is obtained by ignoring the fill-in that arises during the LU-factorization of A.

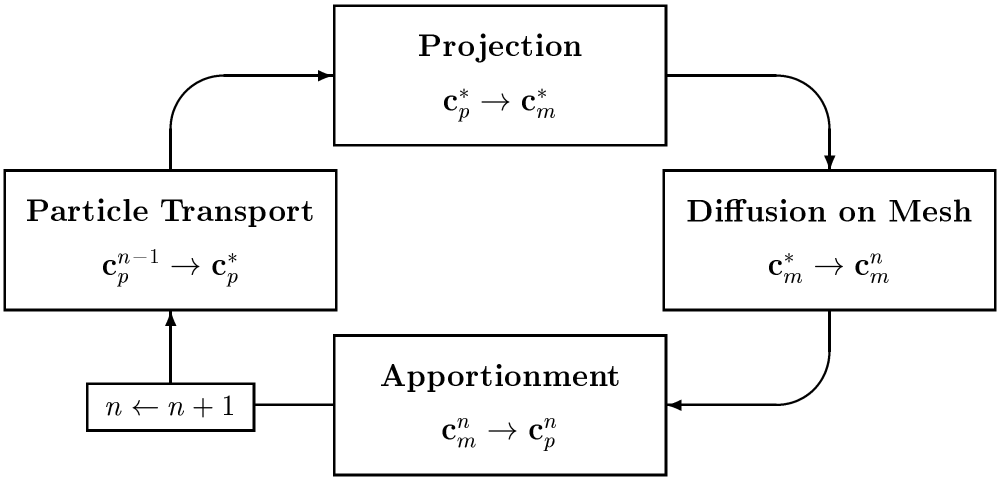

Figure 1: Basic steps of the time cycle in PMM.

3. Particle-mesh method

In this section we consider two computational issues, the ringing instability and numerical dissipation which arise when PMM is applied to the numerical solution of convection-dominated diffusion flow problems.

3.1. Ringing instability

For a systematic treatment of evaluation/interpolation schemes between particle and mesh values in PMM, we may introduce the dual representation [2]

$$c(\mathbf{x}) = \begin{cases} \sum_p c_p \delta(\mathbf{x} - \mathbf{x}_p), & \text{(particle)}, \\ \sum_{ij} c_{ij} \varphi_{ij}(\mathbf{x}), & \text{(mesh)}, \end{cases} \tag{3.1}$$

where $\mathbf{x}_p$ are the positions of particles, $\delta(\mathbf{x})$ is the Dirac delta function, and $\varphi_{ij}(\mathbf{x})$ is the basis function corresponding to the cell K_{ij} in the mesh. Figure 1 shows basic steps of the time cycle in PMM, where subscripts p and m indicate respectively the particle and mesh values. In this article, the evaluation of mesh values from particle values is called the *projection* and the interpolation of particle values from mesh quantities is called the *apportionment*. The projection and apportionment require the particle-mesh and mesh-particle interpolations, respectively.

To illustrate the ringing instability of PMM, we select a model problem: Let $\Omega = (a_x, b_x) \times (a_y, b_y) = (-3, 3)^2$. For $\mathbf{x} = (x, y) \in \Omega$, the velocity is given

$$\mathbf{v}(\mathbf{x}) = (v_1(\mathbf{x}), v_2(\mathbf{x})) = \mu\,(-2\pi y,\, 2\pi x), \quad \mu \geq 0, \tag{3.2}$$

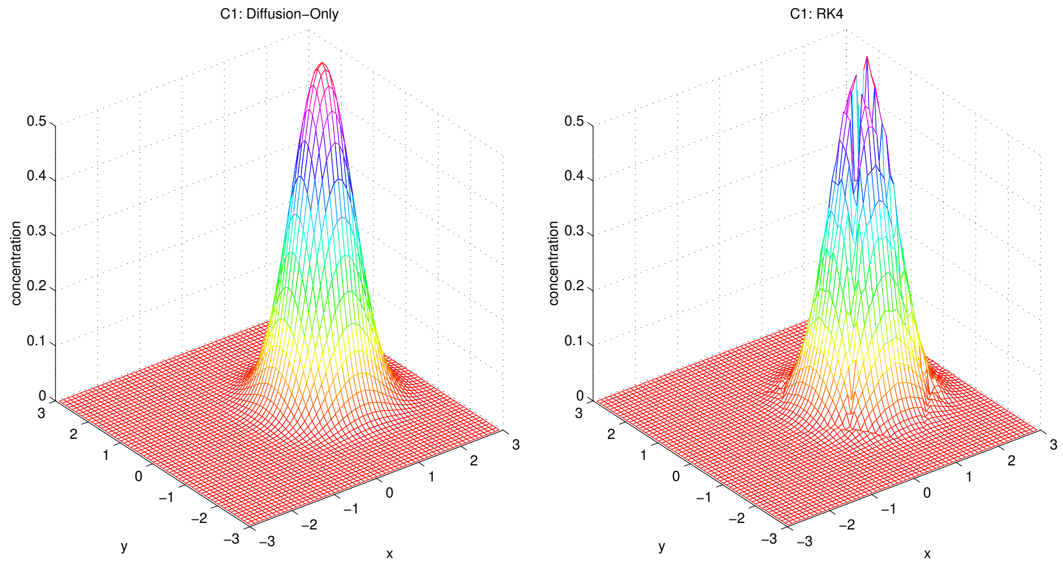

Figure 2: The numerical solutions of (1.1) and (3.2)-(3.4) when $t = 1$: (left) diffusion only and (right) convection-diffusion with the transport part performed by RK4. Set $\varepsilon = 0.1$, $h = 0.1$, and $\sigma = 0.5$.

where μ is the frequency, and the diffusion tensor is chosen

$$D(\mathbf{x}) = \varepsilon I = \varepsilon \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad \varepsilon \geq 0. \tag{3.3}$$

The initial condition is given

$$c^0(\mathbf{x}) = \begin{cases} 1, & \text{if } \mathbf{x} \in (0.5, 1.5) \times (-0.5, 0.5), \\ 0, & \text{else.} \end{cases} \tag{3.4}$$

For the velocity (3.2), the initial value will rotate around the origin $(0, 0)$ with the frequency of μ. Since the convection and diffusion operators commute each other, the analytic solution of the diffusion problem ($\mu = 0$ and $\varepsilon > 0$) is identical to that of convection-diffusion problem ($\mu = 1$ and $\varepsilon > 0$) when t is an integer.

Let $h_x = (b_x - a_x)/N_x$ and $h_y = (b_y - a_y)/N_y$, for some positive integers N_x and N_y. The uniform time step size $\Delta t = T/N_t$ is set such that

$$\Delta t \approx \sigma \cdot \min\left(\frac{h_x}{\max_{\mathbf{x}} |v_1(\mathbf{x})|}, \frac{h_y}{\max_{\mathbf{x}} |v_2(\mathbf{x})|}\right),$$

where σ is the Courant number to be chosen ($\sigma \leq 1$).

Figure 2 depicts computed solutions of the diffusion problem ($\mu = 0$ and $\varepsilon = 0.1$) and the convection-diffusion problem ($\mu = 1$ and $\varepsilon = 0.1$), when $t = 1$, $h = h_x = h_y = 0.1$, and $\sigma = 0.5$. For PMM, four particles are initially located at each cell

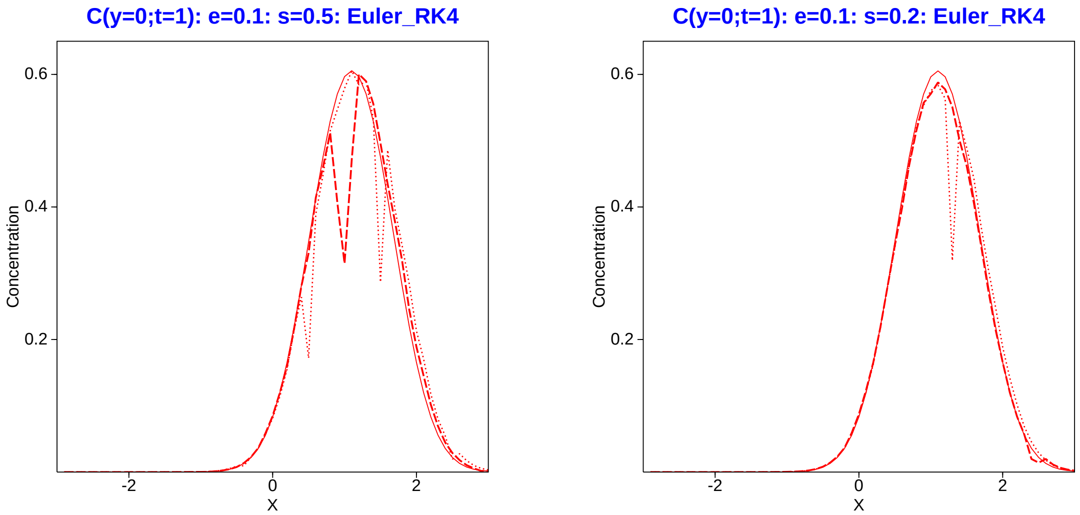

Figure 3: The numerical solution of (1.1) and (3.2)-(3.4) on Y_0 when $t = 1$, for $\sigma = 0.5$ (left) and $\sigma = 0.2$ (right). Set $\varepsilon = 0.1$ and $h = 0.1$. In each side, the thin solid curve indicates the solution of the diffusion problem, while the dotted curve and thick dashed curve are the solutions of convection-diffusion problem by PMM-Euler and PMM-RK4, respectively.

uniformly and RK4 is utilized for solving the transport part. The solution from PMM incorporating RK4 (PMM-RK4) looks very oscillatory. To see details, we plot the solutions together on the cross section

$$Y_0 = \{\mathbf{x} \in \Omega : y = 0\}.$$

Figure 3 presents computed solutions on Y_0 when $\sigma = 0.5$ (left) and $\sigma = 0.2$ (right). (The other parameters are set the same as in Figure 2.) In each side, the solid curve indicates the solution of the diffusion problem ($\mu = 0$), while the dotted curve and thick dashed curve are the solutions of convection-diffusion problem ($\mu = 1$) by PMM-Euler and PMM-RK4, respectively.

In the experiments, we have adopted the interpolation schemes for the projection

$$c_{ij}^* = \sum_p c_p^* \delta(\mathbf{x} - \mathbf{x}_p), \quad \mathbf{x}_p \in K_{ij}, \tag{3.5}$$

and for the apportionment

$$\begin{aligned} &(\mathrm{a}) \quad \widehat{c}_p^n = \begin{cases} c_p^* + (c_{ij}^n - c_{ij}^*)/N_{ij}^{n-1}, & \text{if } c_{ij}^n \ge c_{ij}^*, \\ c_p^* \cdot (c_{ij}^n / c_{ij}^*), & \text{else,} \end{cases} \qquad \mathbf{x}_p \in K_{ij}, \\ &(\mathrm{b}) \quad c_p^n = (1-\eta) \cdot \widehat{c}_p^n + \eta \cdot (c_{ij}^n / N_{ij}^{n-1}), \quad 0 \le \eta \le 1, \end{aligned} \tag{3.6}$$

where N_{ij}^{n-1} is the number of particles in the cell K_{ij}. (When $N_{ij}^{n-1} = 0$, four new particles of zero values are constructed as in the initialization and set $N_{ij}^{n-1} = N_{p,ij}^{n} = 4$.) The basic idea of (3.6.a) and its reasoning would be presented in §3.2. The intermediate quantities $\widehat{c}_p^n$ correspond to the updated concentration from the diffusion (global). The internal diffusion originated from the interaction between the particles in K_{ij} can be applied by setting η appropriately. For the computation presented in Figures 2 and 3, we set

$$\eta = \frac{\alpha}{1+\alpha}, \quad \alpha = \varepsilon \frac{\Delta t^n}{\sqrt{h_x^2 + h_y^2}}. \tag{3.7}$$

Such interpolation schemes have little smooth property. The computed solutions show high amplitude fluctuations. A certain degree of noise is yet observed for the smaller Courant number ($\sigma = 0.2$). When $\sigma = 0.1$, both the Euler scheme and RK4 produce non-oscillatory solutions.

The solution shows a larger fluctuation when the internal diffusion (3.6b) is not applied, i.e., $\eta \equiv 0$. An appropriate choice of η seems important for controlling the numerical dissipation and fluctuation. The larger η is set, the more diffusive and therefore smoother the solution is. The choice in (3.7) has numerically proven effective for the example considered in this section; however, it seems quite problematic.

We have tried higher-order interpolation schemes such as suggested in [3, 4]. They reduce the noise significantly; however, they also shrink the maximum height of the peak to about three quarters the expected, in one rotation. For these cases, PMM performs not better than in both accuracy and efficiency than other methods such as the streamline diffusion [14], an implicit method, and the third-order upwind difference scheme [24] and line methods [15, 21], explicit schemes. Here the point is that numerical dissipation should be carefully analyzed when an interpolation scheme is to be introduced. The main goal of the paper is to study strategies for the noise suppression with a minimum numerical dissipation.

3.2. Numerical dissipation

Let $\mathcal{T}$, $\mathcal{I}_p^m$, $\mathcal{D}$, and $\mathcal{I}_m^p$ be the operators respectively representing the transport solver, the projection, the diffusion solver, and the apportionment. Then, the solution procedure for the space-time slice $\mathcal{S}^n$ can be expressed as

$$\mathbf{c}_p^n = \mathcal{I}_m^p \, \mathcal{D} \, \mathcal{I}_p^m \, \mathcal{T} \, \mathbf{c}_p^{n-1}. \tag{3.8}$$

When the diffusion coefficient is zero, i.e., $\mathcal{D} = I$, where I is the identity operator, the above equation becomes $\mathbf{c}_p^n = \mathcal{I}_m^p \, \mathcal{I}_p^m \, \mathcal{T} \, \mathbf{c}_p^{n-1}$. So the numerical dissipation introduced from the interpolations (projection and apportionment) can be measured from the distance between $\mathcal{I}_m^p\,\mathcal{I}_p^m$ and the identity operator I. A necessary condition for the non-dissipative interpolation is

$$\mathcal{I}_m^p\,\mathcal{I}_p^m = I, \quad \text{when } D = I. \tag{3.9}$$

Research in PMM has been focused on interpolation schemes ($\mathcal{I}_p^m$ and $\mathcal{I}_m^p$) and particle rezoning which can suppress the ringing instability effectively [2, 3, 4, 19]. However, interpolations introduce a degree of numerical dissipation, in particular, when the basis functions utilized in the interpolation have their support larger than the cell size. Such interpolation schemes cannot satisfy (3.9).

In certain physical situations, minimizing numerical dissipation is not as important as suppressing the ringing instability. However, a suppression scheme not considering numerical dissipation can smear interesting physical sharp-fronts out to disappear.

To minimize numerical dissipation in the apportionment step for problems of no or small diffusion, one can apply the interpolation to the *change-in-diffusion*, $\mathbf{c}_m^n - \mathbf{c}_m^*$, and add the result to $\mathbf{c}_p^*$:

$$\begin{aligned}
\mathbf{c}_p^n &= \mathcal{I}_m^p(\mathbf{c}_m^n - \mathbf{c}_m^*) + \mathbf{c}_p^* \\
&= \mathcal{I}_m^p(\mathcal{D}\,\mathcal{I}_p^m\,\mathcal{T}\,\mathbf{c}_p^{n-1} - \mathcal{I}_p^m\,\mathcal{T}\,\mathbf{c}_p^{n-1}) + \mathcal{T}\,\mathbf{c}_p^{n-1} \\
&= [\mathcal{I}_m^p(\mathcal{D} - I)\mathcal{I}_p^m + I]\,\mathcal{T}\,\mathbf{c}_p^{n-1}.
\end{aligned} \tag{3.10}$$

Thus, (3.9) would be satisfied for any interpolation schemes for the projection and apportionment. When (3.10) is directly applied, some particles can have negative mass, although the net cell concentration is positive. The scheme (3.6.a) has been suggested to avoid negative particle values and yet to satisfy (3.9).

4. Dual-mesh characteristics (DMC)

As has been known, PMM can develop a large amplitude fluctuation. Such fluctuation appears particularly when the characteristic disordering happens [3]. Thus it is natural to design a numerical technique which aligns the characteristics when physically lined up. The characteristic disordering seems appearing due to the error when the particles are crossing the cell boundaries, in particular, crossing more than once in a time interval.

The basic idea of DMC is to move the particles based on the dual-mesh of which the second mesh is formed by tracking back the corners of the cells of the original mesh along the characteristics, keeping the edges straight. (Thus the back-tracking projection is affine.) The particle transport can be carried out by interpreting the

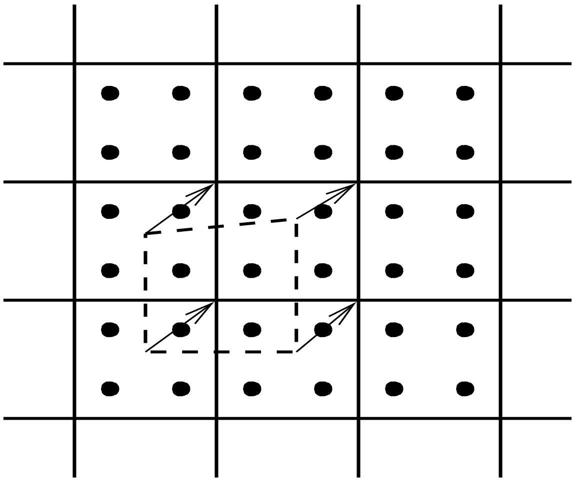

Figure 4: An initialization of the particle-mass at the t^0-level. The dashed box indicates the affine projection of a cell along the characteristics.

particle locations in terms of the bilinear coordinates of the second mesh. See Figure 4, where the computational particles are initialized on the cells at the t^0-level. The following explains the steps of DMC for solving (2.2) in the space-time slice $\mathcal{S}^n$:

1. Choose $\Delta t^n (:= t^n - t^{n-1})$ such that

$$\Delta t^n = \sigma \cdot \min\left(\frac{h_x}{\max_{\mathbf{x}} |v_1^n(\mathbf{x})|}, \frac{h_y}{\max_{\mathbf{x}} |v_2^n(\mathbf{x})|}\right), \quad \sigma \leq 1, \tag{4.1}$$

   where σ is the Courant number and $\mathbf{v}^n = (v_1^n, v_2^n)$ is the velocity at $t = t^n$.

2. For each grid point $\mathbf{x}$, find its tracking-back $\overline{\mathbf{x}}$ by solving (2.6).

3. For each cell K_{ij}, construct its affine projection $\overline{K}_{ij}$ by connecting the tracking-back points of its corners. (The dashed box in Figure 4.) The whole collection of these affine projections of cells forms the second mesh.

4. Interpret the particles in $\overline{K}_{ij}$ in terms of the bilinear coordinates of $\overline{K}_{ij}$. The bilinear interpretations of the particles in $\overline{K}_{ij}$ are their new coordinates in the cell K_{ij} at the t^n-level. The operation is equivalent to advancing the second mesh to the original mesh together with the particles.

A few comments are in order for the above scheme.

In the first step, (4.1) is equivalent to choosing the temporal step size such that the maximum distance of particle transport in the space-time slice $\mathcal{S}^n$ is at most a cell-distance. The second step requires solving final value problems (2.6) for grid points. When the velocity is assumed to be bilinear on each cell, the final value problems can be solved analytically; see §5.

When the velocity is time-independent and $\Delta t^n = \Delta t^{n-1}$, the first three steps of the above process are identical to those of the former time step. Only the fourth step is required to perform; its computation cost is a bilinear interpretation per particle.

The introduction of the affine projection of cells and the bilinear interpretation of particle locations can avoid possible complications and errors arising from the individual particle movement. When a particle is around corner of a cell, for example, it can transport to the next time level passing through three different cells. An accurate computation of the characteristic curve for the particle requires a few applications of an ODE solver, (2.9) or (2.10) or others, to solve (2.8) in $\mathcal{S}^n$; it is clearly more expensive than DMC.

Note that when the fluid is compressible or of a highly oscillatory velocity, some cells can have either more than four particles or no particle. In practice, it is reasonable to set the maximum number of particles to be able to reside in a cell. Thus, whenever the number of particles in a cell is larger than allowed, one has to combine either nearest or smallest two particles into a new particle. Of course, particles of zero mass can be easily eliminated or included [19]. For the examples presented in this paper, we set the maximum particle number in a cell to be 12. Four particles of no mass are included when a cell happens to have no particle.

5. The average velocity

Accuracy of numerical schemes dealing with characteristics can deteriorate unless the final value problem (2.6) (or (2.8)) is solved accurately. Most of experiments incorporating the tracking-back solve the problem by a conventional ODE solver such as the Euler method, the Heun's method, and RK4. However, when $\mathbf{v}(\overline{\mathbf{x}})$ is different from $\mathbf{v}(\mathbf{x})$, the tracking-forward departing from $\overline{\mathbf{x}}$ may not reach at $\mathbf{x}$, due to numerical error. In this section, we discuss strategies to find the *average velocity* which can track back/forward along the characteristics exactly. The explicit formulation of the problem we want to solve is as follows: Find $\overline{\mathbf{x}}$ such that

$$\overline{\mathbf{x}} = \mathbf{x} - \overline{\mathbf{v}}\Delta t^n, \tag{5.1}$$

where $\overline{\mathbf{v}}$ is the average velocity between $\mathbf{x}$ and $\overline{\mathbf{x}}$. The velocity is assumed to be bilinear in each cell.

In most modeling practice, the timestep Δt^n is often chosen such that $\overline{\mathbf{x}}$ is within a cell distance from $\mathbf{x}$, i.e.,

$$|x - \overline{x}| \le h_x, \quad |y - \overline{y}| \le h_y,$$

due to accuracy reasons, where $\mathbf{x} = (x, y)$ and $\overline{\mathbf{x}} = (\overline{x}, \overline{y})$, and h_x and h_y are respectively the edge lengths of the cells in x- and y-directions. Such choice of timestep is equivalent to setting the Courant number not larger than one; see (4.1). Note that $\overline{\mathbf{x}}$ is on the closure of a cell which has $\mathbf{x}$ as one of its vertices. Thus we can see that the velocity is bilinear between $\mathbf{x}$ and $\overline{\mathbf{x}}$.

Since (5.1) can be solved component by component, it suffices to consider the following problem formulated in the one-dimensional space: Find $\overline{x}$ such that

$$\overline{x} = x - \overline{v}\Delta t^n, \tag{5.2}$$

where $\overline{v}$ is the x-component of the average velocity between x and $\overline{x}$.

Let $\widehat{x}$ be the other nodal point of the interval including $\overline{x}$, i.e.,

$$\widehat{x} = \begin{cases} x - h_x, & \text{if } v(x) \geq 0, \\ x + h_x, & \text{else.} \end{cases}$$

Then, the linear velocity between x and $\widehat{x}$ can be formulated as

$$v(\xi) = a(\xi - \widehat{x}) + v(\widehat{x}), \quad a = \frac{v(x) - v(\widehat{x})}{x - \widehat{x}}. \tag{5.3}$$

In particular,

$$v(\overline{x}) = a(\overline{x} - \widehat{x}) + v(\widehat{x}). \tag{5.4}$$

The average velocity $\overline{v}$ and its corresponding tracking-back $\overline{x}$ can be obtained as follows.

Theorem 5.1. *Assume the velocity is locally linear near x as in* (5.3)*. Then the exact average velocity can be found as*

$$\overline{v} = \begin{cases} v(x), & \textit{if } a = 0, \\ \dfrac{v(x) - v(\overline{x})}{\log(v(x)/v(\overline{x}))}, & \textit{else,} \end{cases} \tag{5.5}$$

and (5.2) *can be solved for $\overline{x}$ as*

$$\overline{x} = \begin{cases} x - v(x)\Delta t^n, & \textit{if } a = 0, \\ x - \dfrac{1}{a}\Big[1 - \exp(-a\Delta t^n)\Big]v(x), & \textit{else.} \end{cases} \tag{5.6}$$

Proof. Since $v(x)dt = dx$, we have

$$\int_{t^{n-1}}^{t^n} dt = \int_{\overline{x}}^{x} \frac{d\xi}{v(\xi)}.$$

The exact average velocity (5.5) follows from (5.2), (5.4), and the above equation. Equation (5.6) holds clearly when $a = 0$. Let $a \neq 0$. Note that

$$\begin{aligned} v(x) - v(\overline{x}) &= v(x) - a(\overline{x} - \widehat{x}) - v(\widehat{x}) \\ &= v(x) - v(\widehat{x}) + a\widehat{x} - a\overline{x} \\ &= ax - a\overline{x} \;=\; a(x - \overline{x}). \end{aligned}$$

So it follows from (5.2) and (5.5) that

$$\log\left(\frac{v(\overline{x})}{v(x)}\right) = -a\Delta t^n. \tag{5.7}$$

Since

$$\frac{v(\overline{x})}{v(x)} = \frac{v(x) - a(x - \overline{x})}{v(x)} = 1 - \frac{a(x - \overline{x})}{v(x)},$$

we obtain from (5.7) that

$$1 - \frac{a(x - \overline{x})}{v(x)} = \exp(-a\Delta t^n).$$

The second choice of (5.6) follows from the above. □

Note that (5.6) comes nearer to (2.7) as a approaches zero.

6. Numerical experiments

The dual-mesh characteristics (DMC) has been implemented and tested for various velocities and diffusivities. We present some of the numerical results to demonstrate stability and efficiency of the resulting algorithm, PMM-DMC, incorporating the average velocity and its tracking-back, (5.5)-(5.6). To minimize numerical dissipation, PMM-DMC utilizes the localized interpolation schemes (3.5)-(3.7). Let $\Omega = (-3, 3)^2$. Uniform meshes are selected: $h = h_x = h_y$. The main and driver routines are written in C++, the convection and diffusion solvers are in F77 and C, respectively. A 400 MHz PC of a Linux operating system is utilized for all experiments.

Figure 5 shows the numerical solutions of PMM-DMC for the example treated in §3.1. The thin solid curves and the thick dashed curves denote the solutions on Y_0 of the diffusion problem ($\mu = 0$ and $\varepsilon = 0.1$) and the convection-diffusion problem ($\mu = 1$ and $\varepsilon = 0.1$), respectively, when $t = 1$. The shape and the peak are quite precisely simulated. The highest value of the diffusion solution is 0.609; those of the convection-diffusion solutions after one rotation are 0.590 (97%) and 0.592 (97%) for $\sigma = 0.5$ and $\sigma = 0.2$, respectively. Compared with the results in Figure 3, PMM-DMC shows a superior property in stability, although the incorporated interpolation

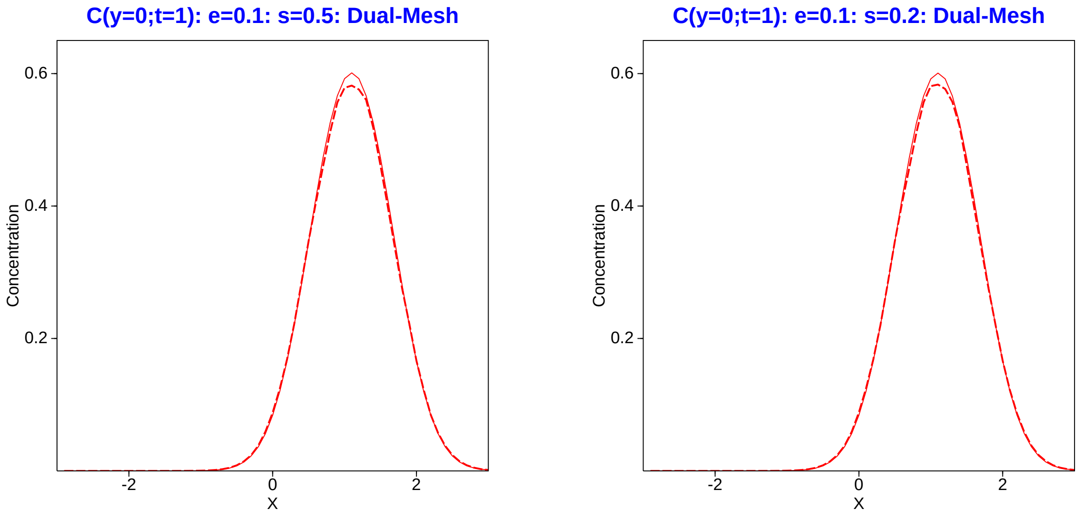

Figure 5: The numerical solution of PMM-DMC on Y_0 at $t = 1$ for (3.2)-(3.4), when for $\sigma = 0.5$ (left) and $\sigma = 0.2$ (right). Set $\varepsilon = 0.1$ and $h = 0.1$. In each side, the thin solid curve indicates the solution of the diffusion problem ($\mu = 0$), while and thick dashed curve is the solution of convection-diffusion problem ($\mu = 1$).

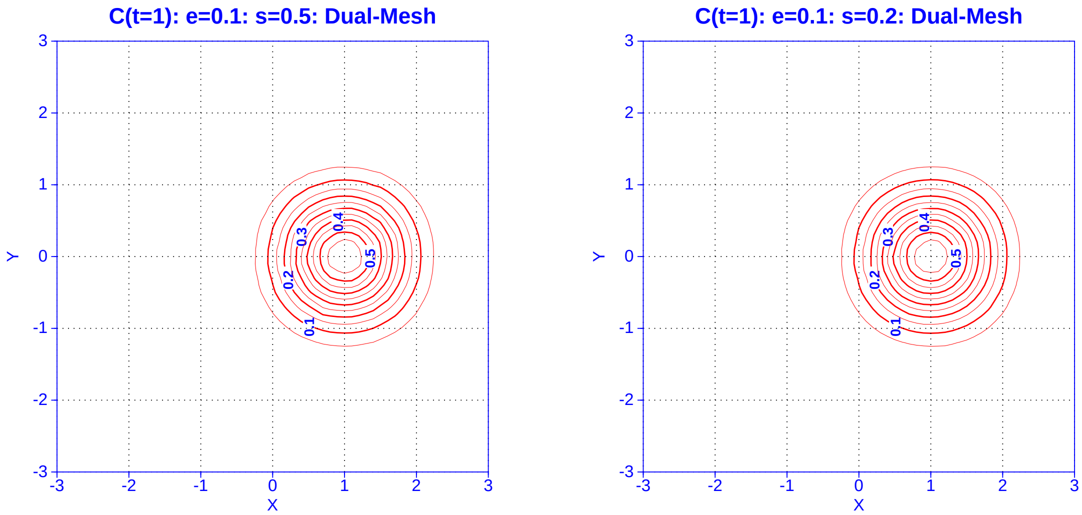

Figure 6: The numerical solutions of PMM-DMC with $\mu = 1$ and $\varepsilon = 0.1$, for $\sigma = 0.5$ (left) and $\sigma = 0.2$ (right). The other algorithm parameters are set the same as in Figure 5.

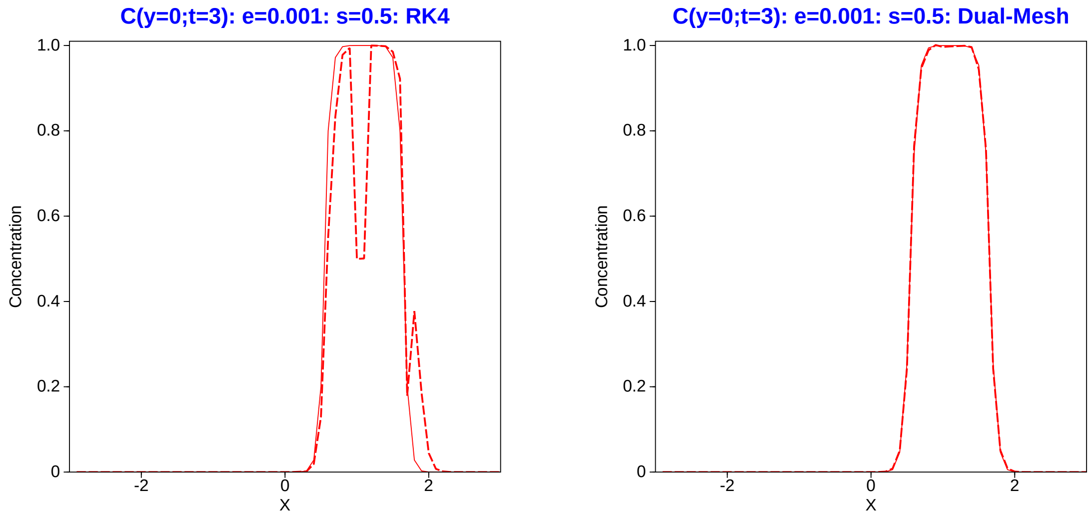

Figure 7: The computed solutions of (1.1) and (3.2)-(3.4) on Y_0 at $t = 3$, for PMM-RK4 (left) and PMM-DMC(right). Set $\varepsilon = 0.001$, $h = 0.1$, and $\sigma = 0.5$.

σ	PMM-Euler	PMM-RK4	PMM-DMC
0.5	5.2	16.1	4.3
0.2	12.5	37.2	10.5

Table 1: The elapsed times (second), for $\mu = 1$, $\varepsilon = 0.1$, $h = 0.1$, and $t = 1$.

schemes have little smooth property. In Figure 6, the contour plots are shown for the convection-diffusion solutions ($\mu = 1$).

In Figure 7, we present plots of the numerical solutions of (1.1) and (3.2)-(3.4) on Y_0 at $t = 3$, when $\varepsilon = 0.001$, $h = 0.1$, and $\sigma = 0.5$. The thin solid curves indicate the diffusion solutions ($\mu = 0$). The thick dashed curves are the convection-diffusion solutions incorporating PMM-RK4 (left) and PMM-DMC (right). PMM-DMC chases the peak and shape excellently for three rotations; no fluctuation and negligible numerical dissipation!

DMC shows advantages on efficiency as shown in Table 1. Set $\mu = 1$ and $t = 1$; the other algorithm parameters are set the same as in Figure 5. The elapsed times are measured in second. PMM-DMC runs slightly faster than the PMM-Euler method and about 3.5 times faster than PMM-RK4, as expected. In solving the diffusion part by PCG-ILU0, the initial value is selected as $\mathbf{c}^{n,0} = \mathbf{c}^*$, the solution of the convection problem.

In Figure 8, we present numerical solutions of PMM-DMC for the convection

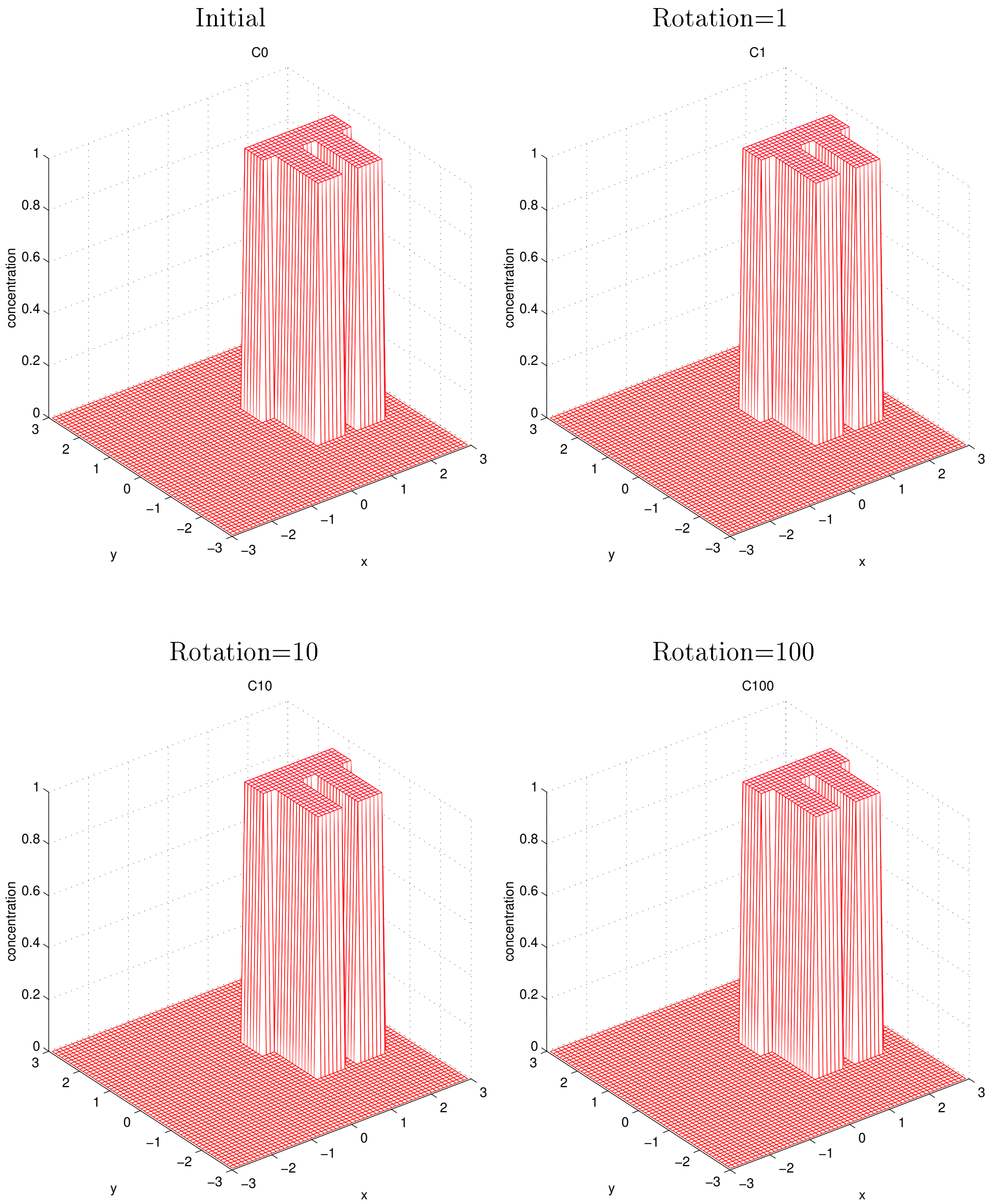

Figure 8: Numerical solutions of PMM-DMC for the convection problem, (3.2)-(3.3) with $\mu = 1$ and $\varepsilon = 0$, having the initial values Π-shaped.

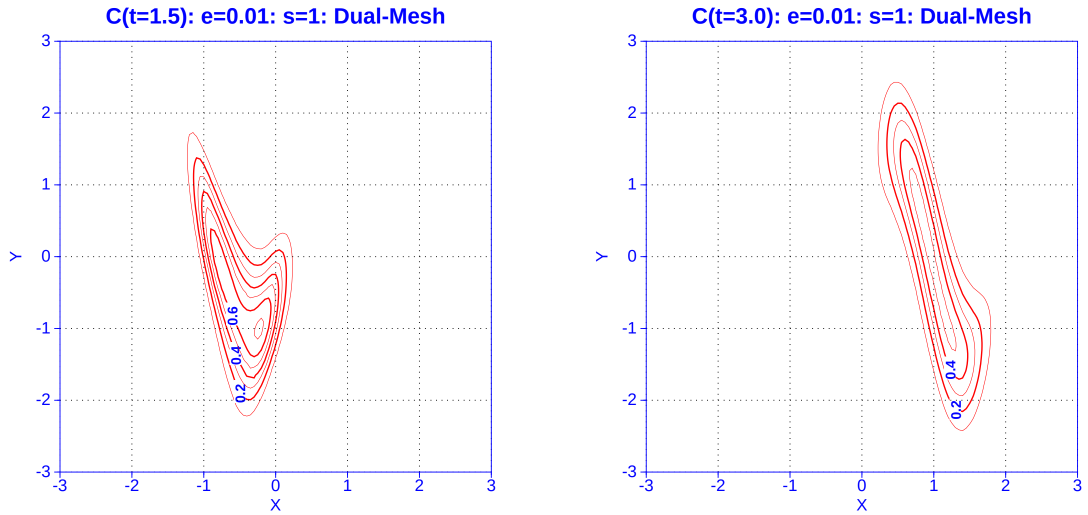

Figure 9: The computed solutions of PMM-DMC for (6.1)-(6.2), when $t = 1.5$ (left) and $t = 3$ (right). Set $\varepsilon = 0.01$, $h = 0.05$, and $\sigma = 1$.

problem, (3.2)-(3.3) with $\mu = 1$ and $\varepsilon = 0$, having the initial values Π-shaped. For the computation, we set $h = 0.1$ and $\sigma = 0.5$. As one can see from the figure, the solution after 100 rotations has exactly the same shape as the initial values. The experiments are repeated with smaller mesh sizes ($h = 0.05$ and $h = 0.025$) with $\sigma = 0.5$ or $\sigma = 1$; the same results have been observed. From these experiments, we conclude that DMC advances the computational particles with *no/negligible* error.

Figure 9 shows the computed solutions of PMM-DMC for $t = 1.5$ (left) and $t = 3$ (right), when the diffusion coefficient $\varepsilon = 0.01$, the velocity is chosen

$$\mathbf{v}(x, y) = (1,\, 2.5\cos(\pi x)), \tag{6.1}$$

and the initial value is given as

$$c_0(\mathbf{x}) = \begin{cases} 1, & \text{if } \mathbf{x} \in (-2.5, -1.5) \times (-0.5, 0.5), \\ 0, & \text{else.} \end{cases} \tag{6.2}$$

The Courant number is chosen to be one. PMM-DMC incorporating the interpolation schemes in (3.5)-(3.7) advances the particles without introducing noise.

7. Discussion

We have seen that DMC advances the computational particles very accurately and more efficiently than individual particle movements, and the interpolation schemes

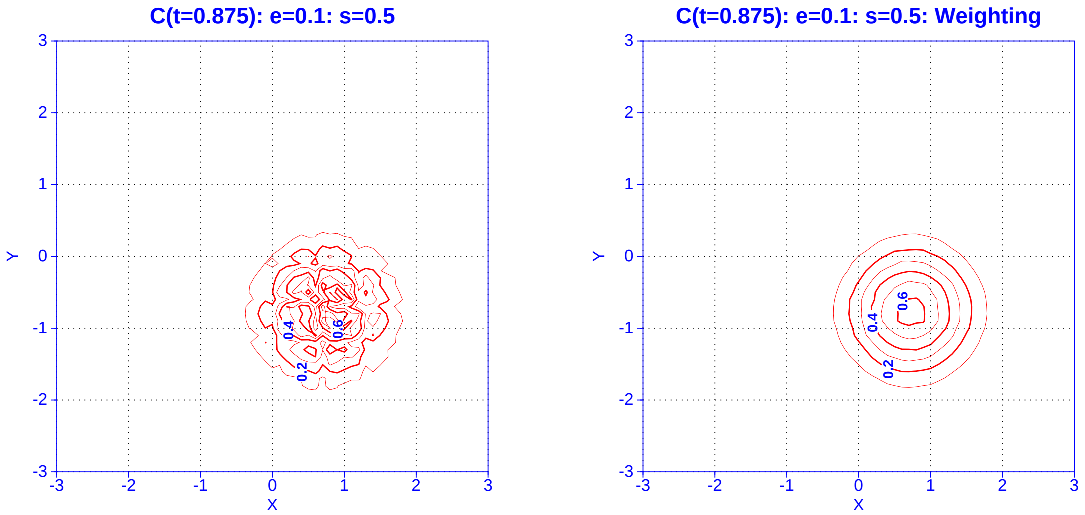

Figure 10: The numerical solution of PMM-DMC for (3.2)-(3.4) at $t = 0.875$. Set $\varepsilon = 0.1$, $\sigma = 0.5$, $\mu = 1$, and $h = 0.1$. The left shows the computed solution, while the right is its area-weighting evaluation.

(3.5)-(3.7) are barely diffusive. In this section we address a *removable* ringing instability, which in principle does not affect the quality of the numerical solution computed by the techniques presented in previous sections.

Figure 10 presents the computed solution of PMM-DMC for (3.2)-(3.4) and its area-weighting evaluation, when $t = 0.875$. The computed solution is oscillatory when no other interpolation schemes are utilized than (3.5)-(3.7); see the left of Figure 10. Since the number of particles in a cell at a certain moment can be larger than that of neighboring cells, an oscillatory solution can easily appear when the cell values are evaluated by the projection scheme (3.5). However, it does not imply that the numerical scheme is instable, because the oscillation is in fact a side effect of the *fixed mesh*. Indeed, we can introduce an appropriate numerical technique to represent the solution into an appropriate form. Consider the following area-weighting scheme

$$c_{ij}^* = \frac{1}{h_x h_y} \sum_p c_p^* \cdot \text{area}(W_{\mathbf{x}_p}(\mathbf{x}) \cap K_{ij}), \quad \mathbf{x}_p \in \Omega, \tag{7.1}$$

where

$$W_{\mathbf{x}_p}(\mathbf{x}) = \begin{cases} 1, & \text{if } |x - x_p| \le h_x/2 \text{ and } |y - y_p| \le h_y/2, \\ 0, & \text{else.} \end{cases}$$

The right of Figure 10 shows the representation of the solution, when (7.1) is applied to the oscillatory solution of the left. The maximum concentration for the representation becomes 0.64, while the solution of the diffusion problem ($\mu = 0$) has its maximum

0.65 at the center of the initial value when $t = 0.875$. The representation shows a great quality! The area-weighting scheme can be viewed as a linear interpolation [3].

It should be noted that the area-weighting evaluation is performed at the last moment just for a representation of the numerical solution; it is never utilized for any of convection and diffusion steps during the time-marching. The localized interpolation schemes (3.5)-(3.7) seem to minimize numerical dissipation and not to introduce non-removable nonphysical oscillations when incorporated with DMC. The area-weighting scheme (7.1) is applied to the solutions obtained from PMM-Euler and PMM-RK4; the results are not satisfactory. One can introduce various higher-order interpolation schemes for a better representation, once for all at the end of simulation.

8. Conclusions

We have studied the particle-mesh method (PMM) for the simulation of incompressible convection-dominated diffusion flows in 2D. For the particle transport, the so-called dual-mesh characteristics (DMC) has been introduced; DMC turns out to track the particles with an excellent accuracy and stability. To minimize numerical dissipation for the problems of no or small diffusion, the interpolation schemes for the projection and apportionment are limited to have lowest-orders of smoothness and only the "change-in-diffusion" has been incorporated in the mesh-particle evaluation. The resulting algorithm, called PMM-DMC, has shown its excellent performance: no oscillation, no/negligible dissipation, and faster computation. The fluctuation as a side effect of a fixed mesh is indispensable; however, the desired solution can be evaluated from particle values by utilizing a smooth interpolation scheme at the end of simulation.

References

[1] T. Arbogast, A. Chilakapati, and M. Wheeler, *A characteristic-mixed finite element method for contaminant transport and miscible displacement*, in Computational Methods in Water Resources, T. Russell, R. Ewing, C. Brebbia, W. Gray, and G. Pindar, eds., Southampton, U.K., 1992, Computational Mechanics Publications, pp. 77–84.

[2] W. Arter and J. Eastwood, *Particle-mesh schemes for advection dominated flows*, J. Comput. Physic., 117 (1995), pp. 194–204.

[3] J. Brackbill, *The ringing instability in particle-in-cell calculations of low-speed flow*, J. Comput. Physic., 75 (1988), pp. 469–492.

[4] G. Coppa, G. Lapenta, G. Dellanpiana, F. Donato, and V. Riccardo, *Blob method for kinetic plasma simulation with variable-size particles*, J. Comput. Physic., 127 (1996), pp. 268–284.

[5] J. Douglas, Jr., F. Pereira, and L.-M. Yeh, *A locally conservative Eulerian-Lagrangian method and its application to nonlinear transport in porous media.* (preprint), 1999.

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

[7] R. Ewing, T. Russell, and M. Wheeler, *Convergence analysis of an approximation of miscible displacement in porous media by mixed finite elements and a modified method of characteristics*, Comput. Meth. Appl. Mech. Eng., 47 (1984), pp. 73–92.

[8] G. Golub and C. V. Loan, *Matrix Computations, 2nd Ed.*, The Johns Hopkins University Press, Baltimore, 1989.

[9] F. Harlow, *The particle-in-cell computing method for fluid dynamics*, in Methods in Computational Physics, B. Alder, S. Fernbach, and M. Rotenberg, eds., vol. 3, Academic Press, New York and London, 1964, pp. 319–343.

[10] R. Hockney and J. Eastwood, *Computer simulation using particles*, Institute of Physics Publishing, Bristol and Philadelphia, 1988.

[11] C. Hu and C.-W. Shu, *A discontinuous Galerkin finite element method for Hamilton-Jacobi equations*, SIAM J. Sci. Comput., 21 (1999), pp. 666–690.

[12] T. J. Hughes and A. Brooks, *A multidimensional upwind scheme with no crosswind diffusion*, in Finite element methods for convection dominated flows, T. J. Hughes, ed., vol. 34 of AMD, ASME, NY, 1979.

[13] T. J. Hughes, L. Franca, and M. Mallet, *A new finite element formulation computational fluid dynamics: I. Symmetric forms of the compressible Euler and Navier-Stokes equations and the second law of thermodynamics*, Comp. Meth. in Appl. Mech. Eng., 54 (1986), pp. 223–234.

[14] T. J. Hughes and M. Mallet, *A new finite element formulation computational fluid dynamics: IV. A discontinuity-capturing operator for multidimensional advective-diffusive systems*, Comp. Meth. in Appl. Mech. Eng., 58 (1986), pp. 329–336.

[15] W. Hundsdorfer and R. Trompert, *Method of lines and direct discretizatio: a comparion for linear advection*, Appl. Numer. Math., 13 (1994), pp. 469–490.

[16] C. Johnson and J. Pitkaranta, *An analysis of the discontinuous Galerkin method for a scalar hyperbolic equation*, Math. Comput., 46 (1986), pp. 1–26.

[17] S. Kim, *A numerical algorithm for fluid flow in 3D naturally fractured porous media.* Submitted to SIAM J. Sci. Comput. (in revision), 2000.

[18] D. Lanser and J. Verwer, *Analysis of operator splitting for advection-diffusion-reaction problems from air pollution modeling*, J. Comp. Appl. Math., 111 (1999), pp. 201–216.

[19] G. Lapenta and J. Brackbill, *Dynamic and selective control of the number of particles in kinetic plasma simulation*, J. Comput. Physic., 115 (1994), pp. 213–227.

[20] P. Lesaint and P. Raviart, *On a finite element method for solving the neuron transport equation*, in Mathematical aspects of finite elements in partial differential equations, C. de Boor, ed., Academic Press, NY, 1974.

[21] R. LeVeque, *Numerical methods for conservation laws*, Lectures in Mathematics ETH Zurich, Birkhauser Verlag, Basel, Switzerland, 1990.

[22] N. Madden and M. Stynes, *Linear enhancements of the streamline diffusion method for convection-diffusion problems*, Comput. Math. Appl., 32 (1996), pp. 29–42.

[23] Y.-T. Shih and H. C. Elman, *Modified streamline diffusion schemes for convection-diffusion problems*, Comput. Methods Appl. Mech. Engr., 174 (1999), pp. 137–151.

[24] J. Verwer, J. Blom, and W. Hundsdorfer, *An implicit-implicit approach for atmospheric transport-chemistry problems*, Appl. Numer. Math., 20 (1996), pp. 191–209.

[25] T. C. Warburton and G. E. Karniadakis, *A discontinuous Galerkin method for the viscous MHD equations*, J. Comput. Phys., 152 (1999), pp. 608–641.