Available online at www.sciencedirect.com
ScienceDirect
Journal of Computational Physics 221 (2007) 198–225
JOURNAL OF COMPUTATIONAL PHYSICS
www.elsevier.com/locate/jcp

A fractional step ELLAM approach to high-dimensional convection–diffusion problems with forward particle tracking

Dong Liang [a,*], Chuanbin Du [a], Hong Wang [b]

[a] Department of Mathematics and Statistics, York University, 4700 Keele Street, Toronto, Ont., Canada M3J 1P3
[b] Department of Mathematics, University of South Carolina, Columbia, SC 29208, USA

Received 12 June 2005; received in revised form 5 June 2006; accepted 7 June 2006
Available online 14 August 2006

Abstract

In this paper, a fractional step method combined with an Eulerian–Lagrangian localized adjoint method (ELLAM) is proposed to solve high-dimensional convection–diffusion problems. The method reduces high-dimensional problems to a series of uncoupled one-dimensional problems in each time step interval, in which one-dimensional ELLAM is used to solve the one-dimensional splitting equations. The approach takes the attractive advantages of the ELLAM method and the fractional step technique. It reduces computational complexities, large memory requirements, and long computation durations due to the application of the splitting technique. It reduces temporal errors and generates accurate numerical solutions even if large time and coarse spatial step sizes are used in computation. It effectively eliminates non-physical oscillation or excessive numerical dispersion and treats boundary conditions well and in a natural way. Numerical experiments show the efficient performance of the approach.

AMS: 65M55; 65M60; 76M10; 76S05

Keywords: Convection–diffusion problems; High-dimensions; Fractional step; ELLAM

1. Introduction

Unsteady convection–diffusion equations that involve a combination of advection and diffusion dynamical processes are among the most widespread in various areas of science and technology, e.g., heat and mass transfer, oil reservoir simulation, groundwater modelling, and aerodynamics and physiology (see, for example [1,2,11,21]). In many such applications, the convection terms essentially dominate the diffusion terms, which leads to a nearly hyperbolic set of governing partial differential equations. The numerical approximation to the problems presents a challenging computational task. It is well documented that when the governing equation is convection-dominated, many standard numerical methods developed for

* Corresponding author. Tel.: +1 416 736 2100x77743; fax: +1 416 736 5757.
E-mail address: dliang@yorku.ca (D. Liang).

0021-9991/$ - see front matter
doi:10.1016/j.jcp.2006.06.022

diffusion-dominated processes often exhibit some combination of difficulties ranging from non-physical oscillations (central difference/Galerkin scheme) to excessive numerical diffusions (upstream scheme) at steep fronts.

Many works have been done to overcome these difficulties and to allow accurate numerical solutions with reasonable computational efforts. One class of approximations is the Eulerian method. In the framework of the Eulerian approach, the traditional finite difference and finite element methods are improved, e.g., the flux corrected transport scheme, total variation diminishing scheme, streamline-upwind Petrov–Galerkin method, and optimal test function method. However, the methods in the Eulerian framework are limited by Courant number restrictions. A second class of approximations is based on treatment of the hyperbolic part by the Lagrangian method, in which the remainder of the equation is treated by an Eulerian-type approximation. While these methods reduce temporal errors and overcome the Courant number restrictions, most of them do not conserve mass. To overcome the limitations associated with these Eulerian–Lagrangian approximations, Celia et al. [4] proposed the Eulerian–Lagrangian localized adjoint method (ELLAM) for one-dimensional convection–diffusion problems. The ELLAM formulation provides a general characteristic solution procedure and a consistent framework for conserving mass and treating boundary conditions. It overcomes the principal shortcoming of some characteristic methods while maintaining their numerical advantages. It reduces temporal errors and therefore allows for large time step sizes in computation without the loss of accuracy, and is highly resistant to numerical dispersion in the presence of small dispersivities. This method is an accurate and efficient solver of the linear convection-dominated diffusion problems with large Courant numbers. The further study of the ELLAM has been successfully taken for high-dimensional problems by Binning and Celia [3], Healy and Russell [13], and Wang et al. [26]. Furthermore, the ELLAM technique combined with the mixed finite element method has successfully been developed to solve the miscible fluid flows in porous media with point sources and sinks in [27,28], which can accurately simulate incompressible and compressible fluid flows in porous media for oil reservoir simulation, as well as for the highly compressible multicomponent fluid flows in porous media.

Due to the complexities and huge computational costs in realistic long term and large scale simulations, there is strong interest in developing an efficient solution technique. This motivates us to study a fractional step ELLAM method (FS-ELLAM) to solve multi-dimensional convection–diffusion problems. The fractional step method is an efficient numerical technique for solving multi-dimensional parabolic problems (see, for example [5,6,8–10,12,14,16,17,19,23–25]). For multi-dimensional problems, when the spatial discretization step sizes are decreased, the number of unknowns of the algebraic equation system arising from the discretization procedure will increase quickly, which leads to rapid increase in computer memory and CPU time. The fractional step method reduces multi-dimensional problems to a series of uncoupled one-dimensional problems, which results in very low execution time and storage. In this paper, we combine a fractional step method with the ELLAM method to develop a fractional step ELLAM (FS-ELLAM) approach for two-dimensional unsteady convection–diffusion problems. In the framework of the fractional step method, we split the two-dimensional convection–diffusion problem into two uncoupled one-dimensional convection–diffusion equations in every time step interval. Then, at each time level, we apply the ELLAM method to solve the one-dimensional subproblems, which simulates accurately the corresponding sub-dimensional equations. The developed FS-ELLAM approach takes the advantages of both the ELLAM method and the fractional step method. It reduces temporal errors and eliminates non-physical oscillation and excessive numerical diffusion. It generates an accurate numerical solution even if large time and coarse spatial step sizes are used. It also treats boundary conditions effectively. Because of the application of the fractional step technique, the FS-ELLAM approach reduces the computation duration and the requirement of large memory in simulation. Numerical experiments show the efficient performance of the developed FS-ELLAM approach. The technique can be easily applied to three-dimensional large-scale convection–diffusion problems and the procedure can be solved by parallel computing systems.

This paper is organized as follows. In Section 2, we introduce the mathematical model and the fractional step technique. Then, we propose the fractional step ELLAM approach for two-dimensional convection–diffusion problems in Section 3. Numerical experiments are given and analysed in detail in Section 4. Finally, some conclusions are addressed in Section 5.

2. Mathematical model and fractional step technique

2.1. The mathematical model

We consider the following unsteady convection–diffusion problem:

$$\frac{\partial c}{\partial t} + \nabla(\vec{u}c) - \nabla(D\nabla c) = f(x,y,t), \quad (x,y,t) \in \Omega \times (0,T], \tag{1}$$

$$c(x,y,t) = g_{\text{in}}(x,y,t), \quad (x,y,t) \in \Gamma_{\text{in}} \times (0,T], \tag{2}$$

$$D\nabla c(x,y,t) \cdot \vec{n} = h_{\text{out}}(x,y,t), \quad (x,y,t) \in \Gamma_{\text{out}} \times (0,T], \tag{3}$$

$$c(x,y,0) = c_0(x,y), \quad (x,y) \in \bar{\Omega}, \tag{4}$$

where $t > 0$ is the time and (x, y) is the spatial location; $T > 0$ is the time period; $\Omega = [a_x, b_x] \times [a_y, b_y]$ is the domain of flow, Γ_{in} denotes the inflow boundary of Ω, Γ_{out} denotes the outflow boundary of Ω, $\partial\Omega = \Gamma_{\text{in}} + \Gamma_{\text{out}}$, and $\vec{n}$ is the unit outer normal to the boundary $\partial\Omega$; $\vec{u} = (u_x(x,y), u_y(x,y))^{\tau}$ is the velocity of the flow field; $D = \text{diag}(D_x, D_y)$ is the diffusion tensor with $D_x > 0$ and $D_y > 0$; $f = f(x, y, t)$ is the given source term; $g_{\text{in}}(x, y, t)$, $h_{\text{out}}(x, y, t)$ and $c_0(x, y)$ are given boundary and initial data, respectively.

Because of the nature of hyperbolic types, the convection–diffusion problems involve moving sharp fronts or boundary layers. For improving numerical results and reducing numerical oscillation and excessive dispersion, the efficient grid refinements are required if the standard methods are used in computation. But for high-dimensional large-scale problems, it may lead to very large linear systems and need long computation times (see, for example [1,2,11,15,18,21]). The fractional step method leads multi-dimensional problems to a series of uncoupled one-dimensional problems and can be solved by parallel computing systems. However, the efficient numerical method, used for the one-dimensional convection–diffusion subproblems, will play an important role in the fractional step procedure of solving multi-dimensional convection–diffusion problems. The most effective advantage of the ELLAM makes us develop a fractional step ELLAM approach for high-dimensional convection–diffusion problems.

2.2. The fractional step technique

In this part, we first give a review for the splitting technique of the governing equations (1)–(4). Then, we discuss briefly the construction feature of the fractional step approach. We recall the model in the following form:

$$\frac{\partial c}{\partial t} - (L_x + L_y)c = f, \tag{5}$$

where $L = L_x + L_y$ such that

$$L_x c \equiv -\frac{\partial}{\partial x}(u_x c) + \frac{\partial}{\partial x}\left(D_x \frac{\partial}{\partial x} c\right), \tag{6}$$

$$L_y c \equiv -\frac{\partial}{\partial y}(u_y c) + \frac{\partial}{\partial y}\left(D_y \frac{\partial}{\partial y} c\right). \tag{7}$$

The operator L_x can be considered as a family of one-dimensional differential operators with a given parameter $y \in (a_y, b_y)$ as well as the operator L_y is considered as another family of one-dimensional differential operators with a given parameter $x \in (a_x, b_x)$. Then, the fractional step method can be described as follows. In each time interval $t \in (t^n, t^{n+1}]$, along x-direction, we have the splitting equation:

$$\frac{\partial c}{\partial t} = L_x c + f(x,y,t), \quad (x,t) \in (a_x, b_x) \times (t^n, t^{n+1}], \quad y \in (a_y, b_y), \tag{8}$$

$$c^n(x,y) = c(x,y,t^n), \quad x \in (a_x, b_x), \quad y \in (a_y, b_y), \tag{9}$$

and let $c^{(n+1)*} = c(x, y, t^{n+1})$, $x \in (a_x, b_x)$, $y \in (a_y, b_y)$. Then, along y-direction, we have the splitting equation:

$$\frac{\partial c}{\partial t} = L_y c, \quad (y,t) \in (a_y, b_y) \times (t^n, t^{n+1}], \quad x \in (a_x, b_x), \tag{10}$$

$$c^n(x,y) = c^{(n+1)*}, \quad y \in (a_y, b_y), \quad x \in (a_x, b_x). \tag{11}$$

If we denote the solution of the splitting approximation by $\hat{c}$, we will have the following comment on the convergence of the splitting approach. The sequential solution of the splitting system of (8), (9) and (10), (11) is called to converge to the exact solution of equation (5) (see [19,24]) if it holds that

$$\lim_{\Delta t \to 0} \|c(x,y,t^{n+1}) - \hat{c}(x,y,t^{n+1})\|_{L^2(\Omega)} = 0 \tag{12}$$

for $n \geqslant 0$.

Since the splitting equations are generally solved by numerical computations, the ELLAM method will be applied for obtaining an efficient splitting approach to the governing equations in the present study. Another thing worth mentioning about the fractional step approach is the intermediate boundary conditions. It is necessary to give appropriate intermediate boundary conditions for the splitting equations (8) and (9) so that near the boundaries the approach keeps the same accuracy as that in the interior domain. A detailed discussion of proper intermediate boundary conditions will be given in the next section.

3. The FS-ELLAM approach

In this section we will present our FS-ELLAM algorithm for unsteady convection–diffusion problems in two dimensions. It is straightforward to describe the proposed algorithm in three or higher dimensions.

3.1. The x-directional ELLAM scheme

After splitting the governing equation by the Godunov algorithm, we obtain two uncoupled one-dimensional convection–diffusion subproblems (8) and (10) in x-direction and y-direction, respectively. In this part, we formulate an ELLAM scheme for the x-directional splitting equation in each time step interval $(t^n, t^{n+1}]$. For simplicity, let $\Omega_x = (a_x, b_x)$ and $\Omega_y = (a_y, b_y)$.

For a fixed $y = \bar{y} \in \Omega_y$, the x-dimensional splitting equation is: for $t \in (t^n, t^{n+1}]$

$$\frac{\partial c(x,\bar{y},t)}{\partial t} + \frac{\partial (u_x c(x,\bar{y},t))}{\partial x} - \frac{\partial}{\partial x}\left(D_x \frac{\partial c(x,\bar{y},t)}{\partial x}\right) = f(x,\bar{y},t), \quad x \in \Omega_x, \tag{13}$$

and the corresponding boundary conditions are

$$c(a_x,\bar{y},t) = g_x^*(\bar{y},t), \quad t \in (t^n, t^{n+1}],$$

$$D_x \frac{\partial c}{\partial x}(b_x,\bar{y},t) = h_x(\bar{y},t), \quad t \in (t^n, t^{n+1}],$$

where the intermediate boundary value $g_x^*(\bar{y},t)$ will be defined later in Section 3.3, and the initial condition is:

$$c^n(x,\bar{y}) = c(x,\bar{y},t^n), \quad x \in \Omega_x.$$

Let $w(x,t)$ be the space–time test function in $\bar{\Omega}_x \times (t^n, t^{n+1}]$. Then, the weak form of the x-dimensional splitting equation is

$$\int_{t^n}^{t^{n+1}} \int_{a_x}^{b_x} \left[\frac{\partial c}{\partial t} + \frac{\partial (u_x c)}{\partial x} - \frac{\partial}{\partial x}\left(D_x \frac{\partial c}{\partial x}\right)\right] w(x,t)\,\mathrm{d}x = \int_{t^n}^{t^{n+1}} \int_{a_x}^{b_x} f(x,\bar{y},t) w(x,t)\,\mathrm{d}x\,\mathrm{d}t.$$

Integrating by parts, we get the reference equation of (13):

$$\int_{a_x}^{b_x} c(x,\bar{y},t^{n+1})w(x,t^{n+1})\,\mathrm{d}x + \int_{t^n}^{t^{n+1}}\int_{a_x}^{b_x} D_x \frac{\partial c(x,\bar{y},t)}{\partial x}\frac{\partial w(x,t)}{\partial x}\,\mathrm{d}x\,\mathrm{d}t + \int_{t^n}^{t^{n+1}} \left(u_x c - D_x\frac{\partial c}{\partial x}\right)(x,\bar{y},t)w(x,t)\Big|_{a_x}^{b_x}\mathrm{d}t - \int_{t^n}^{t^{n+1}}\int_{a_x}^{b_x} c(x,\bar{y},t)\left(\frac{\partial w}{\partial t}+u_x\frac{\partial w}{\partial x}\right)(x,t)\,\mathrm{d}x\,\mathrm{d}t = \int_{a_x}^{b_x} c(x,\bar{y},t^n)w(x,t_+^n)\,\mathrm{d}x + \int_{t^n}^{t^{n+1}}\int_{a_x}^{b_x} f(x,\bar{y},t)w(x,t)\,\mathrm{d}x\,\mathrm{d}t, \tag{14}$$

where we denote $w(x,t_+^n)=\lim_{t\to t_+^n} w(x,t)$ since our test function $w(x,t)$ is discontinuous at time level $t=t^n$.

For designing the simple and effective test function $w(x,t)$, we consider the test function to satisfy

$$\frac{\partial w}{\partial t}+u_x\frac{\partial w}{\partial x}=0. \tag{15}$$

Its characteristic is, for any given point $(\bar{x},\bar{t})$ with $\bar{x}\in[a_x,b_x]$, $\bar{t}\in[t^n,t^{n+1}]$, the solution of equation:

$$\frac{\mathrm{d}X}{\mathrm{d}\theta}=u_x(X,\theta), \tag{16}$$

$$X(\theta;\bar{x},\bar{t})|_{\theta=\bar{t}}=\bar{x}. \tag{17}$$

From Eq. (15), the test function $w(x,t)$ is constant along the characteristic satisfying (16) and (17) in the domain $[a_x,b_x]\times[t^n,t^{n+1}]$. The Euler tracking algorithm can be used for solving the characteristic equation (16) and (17) to obtain the approximation $X(\theta;\bar{x},\bar{t})=\bar{x}+(\theta-\bar{t})u_x(\bar{x},\bar{t})$. Normally, let $X(\theta;x,t^n)$ denote the characteristic originated at point (x,t^n) and $X(\theta;a_x,t)$ denote the characteristic originated at point (a_x,t).

Further, we introduce some notations: $(\widetilde{x},t^{n+1})$ denotes the point that (x,t^n) tracks forward to at time t^{n+1}, and (x^*,t^n) denotes the point that tracks forward to (x,t^{n+1}); Particularly, $(b_x^*(t),t^n)$ denotes the point that tracks forward to (b_x,t^{n+1}) (or to outflow boundary at $(b_x,\widetilde{t})$) and $(a_x,t^*(x))$ denotes the point on the inflow boundary which is tracked backward from (x,t^{n+1}) for $x\in[a_x,b_x]$. For any $x\in[a_x,b_x]$ at time t^{n+1}, we extend to define $t^*(x)=t^n$ if the characteristic does not track backward to the in-flow space–time boundary. For the outflow boundary, we introduce a fine partition. The nodes are defined as: $t_{n,k}=t^{n+1}-\frac{k\Delta t_f}{IC_x}$, $k=0,1,\ldots,IC_x$, where $\Delta t_f=\frac{\Delta t}{IC_x}$ denotes the small time size on the outflow boundary and IC_x is an integer number (see Fig. 1).

Now, at time level $t=t^{n+1}$, we choose the test function to satisfy:

$$w_i(x,t^{n+1})=\begin{cases}\frac{x-x_{i-1}}{\Delta x}, & x\in[x_{i-1},x_i],\\ \frac{x_{i+1}-x}{\Delta x}, & x\in(x_i,x_{i+1}],\\ 0, & \text{otherwise}\end{cases} \tag{18}$$

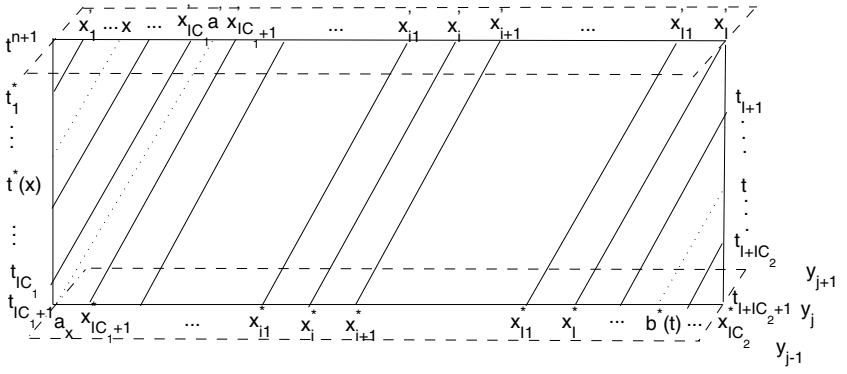


Fig. 1. The partition of x-directional domain $[a_x,b_x]\times(t^n,t^{n+1}]$ at $y=y_j$.

and at the outflow boundary,

$$w_{I+i}(b,t)=\begin{cases}\frac{t_{n,i-1}-t}{\Delta t_f}, & x\in[t_{n,i-1},t_{n,i}],\\ \frac{t-t_{n,i+1}}{\Delta t_f}, & t\in(t_{n,i},t_{n,i+1}],\\ 0, & \text{otherwise.}\end{cases} \tag{19}$$

Meanwhile, along the approximating characteristic, the test function $w_i(x,t)$ is constant in the domain $[a_x,b_x]\times(t_n,t_{n+1}]$, which is same as the value $w_i(\cdot,t^{n+1})$ at the level $t=t^{n+1}$. Let V_{hx}^{n+1} be the finite element space with base functions w_i, $i=1,2,\ldots,I+IC_x+1$.

Let $C^{(n+1)*}$ be the intermediate step approximation of c^{n+1}. Thus, we can define the ELLAM scheme from the reference equation (14) for the x-directional splitting equation (13) as follows: find $C^{(n+1)*}\in V_{hx}^{n+1}$ with $C^{(n+1)*}(a_x)=g_x^*(t^{n+1})$ such that

$$\begin{aligned}&\int_{a_x}^{b_x}C^{(n+1)*}(x,\bar y,t^{n+1})w(x,t^{n+1})\,dx+\int_{t^n}^{t^{n+1}}u_x(b_x,\bar y,t)C^{(n+1)*}(b_x,\bar y,t)w(b_x,t)\,dt\\&\quad+\int_{a_x}^{b_x}(t^{n+1}-t^*(x))D_x(x,\bar y,t^{n+1})\frac{\partial C^{(n+1)*}}{\partial x}(x,\bar y,t^{n+1})\frac{\partial w}{\partial x}(x,t^{n+1})\,dx\\&\quad-\int_{a_x}^{\widetilde a}\frac{dt^*(x)}{dx}D_x(x,\bar y,t^{n+1})\frac{\partial C^{(n+1)*}}{\partial x}(x,\bar y,t^{n+1})w(x,t^{n+1})\,dx\\&=\int_{a_x}^{b_x}C^n(x,\bar y)w(x,t_+^n)\,dx+\int_{t^n}^{t^{n+1}}u_x(a_x,\bar y,t)g_x^*(\bar y,t)w(a_x,t)\,dt-\int_{t^n}^{t^{n+1}}h_x(\bar y,t)w(b_x,t)\,dt\\&\quad-\int_{t^n}^{t^{n+1}}(t-t^n)h_x(\bar y,t)\frac{\partial w}{\partial t}(b_x,t)\,dt-\int_{t^n}^{t^{n+1}}(t-t^n)\frac{\partial X}{\partial t}(t;b^*(t),t^n)f(b_x,\bar y,t)w(b_x,t)\,dt\\&\quad+\int_{a_x}^{b_x}(t^{n+1}-t^*(x))f(x,\bar y,t^{n+1})w(x,t^{n+1})\,dx\end{aligned} \tag{20}$$

for any test function $w(x,t)=w_i(x,t)$, $i=1,2,\ldots,I+IC_x+1$. Where $g_x^*(\bar y,t)$ is the intermediate boundary condition function. Here, for the intermediate splitting equation, we do not use the original boundary condition function $g_x(\bar y,t)$ but instead, the modified intermediate boundary condition function is used. The more details of $g_x^*(\bar y,t)$ will be described in Section 3.3.

3.2. The y-directional ELLAM scheme

In each time interval $(t_n,t_{n+1}]$, after calculating the x-directional splitting equation for all $\bar y=y_j$, we sweep in y-direction to solve the y-directional splitting equation (10) on domain Ω_y for all $\bar x=x_i$. Similarly, we can construct the y-directional ELLAM Scheme. For a fixed $x=\bar x\in\Omega_x$, the y-directional splitting equation is: for $t\in(t_n,t_{n+1}]$

$$\frac{\partial c(\bar x,y,t)}{\partial t}+\frac{\partial(u_y c(\bar x,y,t))}{\partial y}-\frac{\partial}{\partial y}\left(D_y\frac{\partial c(\bar x,y,t)}{\partial y}\right)=0,\quad y\in\Omega_y, \tag{21}$$

and the boundary conditions are:

$$c(\bar x,a_y,t)=g_y(\bar x,t),\quad t\in(t^n,t^{n+1}],$$
$$D_y\frac{\partial c}{\partial y}(\bar x,b_y,t)=h_y(\bar x,t),\quad t\in(t^n,t^{n+1}].$$

Let $C^{(n+1)*}(\bar x,y)$ be the intermediate step approximation obtained in solving the x-directional splitting equation, then the initial condition is given as

$$c^n(\bar x,y)=C^{(n+1)*}(\bar x,y).$$

Further, we introduce that $y_j, j=0,1,\ldots,J$, denote the nodes at $t=t^{n+1}$; $(\widetilde y,t^{n+1})$ denotes the point that (y,t^n) tracks forward to at time t^{n+1}; (y^*,t^n) denotes the point that tracks forward to (y,t^{n+1}); $(b_y^*(t),t^n)$ denotes the point that tracks forward to (b_y,t^{n+1}) (or to outflow boundary at $(b_y,\widetilde t)$); and $(a_y,t^*(y))$ denotes the point

on the inflow boundary which tracks forward to (y, t^{n+1}). For any $y \in [a_y, b_y]$ at time t^{n+1}, we define $t^*(y) = t^n$ if the characteristic does not track backward to the in-flow space–time boundary. On the outflow boundary, the nodes are $t_{n,k} = t^{n+1} - \frac{k\Delta t_f}{IC_y}$, $k = 0, 1, \ldots, IC_y$, where $\Delta t_f = \frac{\Delta t}{IC_y}$ denotes the small time size on the outflow boundary and IC_y is an integer number.

Take the test functions at time $t = t^{n+1}$ as:

$$w_j(y, t^{n+1}) = \begin{cases} \frac{y - y_{j-1}}{\Delta y}, & y \in [y_{j-1}, y_j], \\ \frac{y_{j+1} - y}{\Delta y}, & y \in (y_j, y_{j+1}], \\ 0, & \text{otherwise} \end{cases} \tag{22}$$

and at the outflow boundary

$$w_{J+j}(b_y, t) = \begin{cases} \frac{t_{n,j-1} - t}{\Delta t_f}, & t \in [t_{n,j+1}, t_{n,j}], \\ \frac{t - t_{n,j+1}}{\Delta t_f}, & t \in (t_{n,j}, t_{n,j+1}], \\ 0, & \text{otherwise.} \end{cases} \tag{23}$$

In domain $\Omega_y \times (t^n, t^{n+1}]$, the test functions are defined as constant (with the same value of $w_j(\cdot, t^{n+1})$) along the characteristics. Let V_{hy}^{n+1} be the finite element space with base functions $\{w_j(y, t)\}$.

Thus, the y-directional ELLAM scheme for the y-directional splitting equation (21) is defined as: For all $\bar{x} = x_i$, find $C^{n+1}(\bar{x}, y) \in V_{hy}^{n+1}$ with $C^{n+1}(a_y) = g_y(a_y, t^{n+1})$ such that

$$\begin{aligned} &\int_{a_y}^{b_y} C^{n+1}(\bar{x}, y) w(y, t^{n+1})\,\mathrm{d}y + \int_{t^n}^{t^{n+1}} u_y(\bar{x}, b_y, t) C^{n+1}(\bar{x}, b_y, t) w(b_y, t)\,\mathrm{d}t \\ &\quad + \int_{a_y}^{b_y} (t^{n+1} - t^*(y)) D_y(\bar{x}, y, t^{n+1}) \frac{\partial C^{n+1}}{\partial y}(\bar{x}, y) \frac{\partial w}{\partial y}(y, t^{n+1})\,\mathrm{d}x - \int_{a_y}^{\widetilde{a_y}} \frac{\mathrm{d}t^*(y)}{dy} D_y(\bar{x}, y, t^{n+1}) \frac{\partial C^{n+1}}{\partial y}(\bar{x}, y) w(y, t^{n+1})\,\mathrm{d}y \\ &= \int_{a_y}^{b_y} C^{(n+1)*}(\bar{x}, y) w(y, t_+^n)\,\mathrm{d}y + \int_{t^n}^{t^{n+1}} u_y(\bar{x}, a_y, t) g_y(\bar{x}, t) w(a_y, t)\,\mathrm{d}t - \int_{t^n}^{t^{n+1}} h_y(\bar{x}, t) w(b_y, t)\,\mathrm{d}t \\ &\quad - \int_{t^n}^{t^{n+1}} (t - t^n) h_y(\bar{x}, t) \frac{\partial w}{\partial t}(b_y, t)\,\mathrm{d}t, \end{aligned} \tag{24}$$

for any test base function w_j, $j = 1, 2, \ldots, J + IC_y + 1$. Where $g_y(\bar{x}, t)$ and $h_y(\bar{x}, t)$ are the original boundary conditions.

3.3. Intermediate boundary conditions

In general, the boundary conditions for the middle step splitting equations do not simply copy the original boundary conditions, but are designed such that the approximation system has high accuracy both in the interior domain and near the boundaries (see, for example [17,23,24]).

We recall the fractional step scheme (8) and (10) with notations L_x and L_y defined in (6) and (7) in Section 2.2. Let c^* be the solution of (8) and let $c^{*(n+1)}(x, y)$ be $c^*(x, y, t^{n+1})$ at $t = t^{n+1}$. Let $c^{*(n+1)}(a_x, \bar{y})$ denote the intermediate boundary value on $x = a_x$. We can express $c^{*(n+1)}(a_x, \bar{y})$ in the terms of boundary values by using Taylor expansion at $t = t^n$ and Eq. (8),

$$\begin{aligned} c^{*(n+1)}(a_x, \bar{y}) &= c^*(a_x, \bar{y}, t^n) + \Delta t c_t^*(a_x, \bar{y}, t^n) + \frac{(\Delta t)^2}{2} c_{tt}^*(a_x, \bar{y}, t^n) + \mathrm{O}((\Delta t)^3) \\ &= c^*(a_x, \bar{y}, t^n) + \Delta t (L_x c^* + f)(a_x, \bar{y}, t^n) + \frac{(\Delta t)^2}{2} [L_x (L_x c^* + f_t) + f](a_x, \bar{y}, t^n) + \mathrm{O}((\Delta t)^3). \end{aligned} \tag{25}$$

Since $c^*(x,\bar{y},t^n) = c(x,\bar{y},t^n)$ for $x \in [a_x, b_x]$, it holds

$$\begin{aligned} c^{*(n+1)}(a_x,\bar{y}) &= c(a_x,\bar{y},t^n) + \Delta t(L_x c + f)(a_x,\bar{y},t^n) + \frac{(\Delta t)^2}{2}(L_x L_x c + L_x f_t + f)(a_x,\bar{y},t^n) + \mathrm{O}((\Delta t)^3) \\ &= c(a_x,\bar{y},t^n) + \Delta t(c_t - L_y c)(a_x,\bar{y},t^n) + \frac{(\Delta t)^2}{2}(L_x L_x c + L_x f_t + f)(a_x,\bar{y},t^n) + \mathrm{O}((\Delta t)^3), \end{aligned} \tag{26}$$

where we used Eq. (5) of the exact solution, which gives $L_x c + f = c_t - L_y c$ at $t = t^n$ in (26). Further, using the numerical integration

$$\begin{aligned} \Delta t L_y c(a_x,\bar{y},t^n) &= \int_{t^n}^{t^{n+1}} L_y c(a_x,\bar{y},\tau)\,\mathrm{d}\tau - \frac{(\Delta t)^2}{2} L_y c_t(a_x,\bar{y},t^n) + \mathrm{O}((\Delta t)^3) \\ &= \int_{t^n}^{t^{n+1}} L_y c(a_x,\bar{y},\tau)\,\mathrm{d}\tau - \frac{(\Delta t)^2}{2} L_y(L_x c + L_y c + f)(a_x,\bar{y},t^n) + \mathrm{O}((\Delta t)^3), \end{aligned} \tag{27}$$

and expanding $c(a_x,\bar{y},t^n)$ and $c_t(a_x,\bar{y},t^n))$ by Taylor expansion, from (26) we have

$$\begin{aligned} c^{*(n+1)}(a_x,\bar{y}) &= c(a_x,\bar{y},t^{n+1}) - \int_{t^n}^{t^{n+1}} L_y c(a_x,\bar{y},\tau)\,\mathrm{d}\tau \\ &\quad + \frac{(\Delta t)^2}{2}(L_x L_x c + L_x L_y c + L_y L_y c + L_x f_t + L_y f + f)(a_x,\bar{y},t^n) + \mathrm{O}((\Delta t)^3). \end{aligned} \tag{28}$$

Ignoring the high order terms of Δt in (28), $c^{*(n+1)}(a_x,\bar{y})$ just is the nth time level solution of the second splitting equation (10)

$$\frac{\partial c}{\partial t} = L_y c, \quad (\bar{y},t) \in (a_y,b_y) \times [t^n, t^{n+1}], \tag{29}$$

$$c^n(x,y) = c^{*(n+1)}, \quad \bar{y} \in (a_y,b_y), \tag{30}$$

on boundary $x = a_x$ by the direct integration from t^{n+1} back to t^n. Thus, the scheme of intermediate boundary values $c^{*(n+1)}(a_x)$ for the first (x-directional) splitting equation (8) can be defined by solving numerically the second (y-directional) splitting equation (10) on boundaries with the values at t^n as unknowns. The intermediate boundary scheme can be defined as follows: for $x = a_x$ and $y \in \Omega_y = [a_y, b_y]$, find $C^{(n+1)*}(a_x, y)$ such that

$$\begin{aligned} \int_{a_y}^{b_y} C^{(n+1)*}(a_x,y) w(y,t_+^n)\,\mathrm{d}y &= \int_{a_y}^{b_y} c(a_x,y,t^{n+1}) w(y,t^{n+1})\,\mathrm{d}y + \int_{t^n}^{t^{n+1}} (u_y c)(a_x,b_y,t) w(b_y,t)\,\mathrm{d}t \\ &\quad + \int_{\underset{a_y}{\sim}}^{b_y} (t^{n+1} - t^*(y)) D_y \frac{\partial c}{\partial y}(a_x,y,t^{n+1}) \frac{\partial w}{\partial y}(y,t^{n+1})\,\mathrm{d}y \\ &\quad + \int_{t^n}^{t^{n+1}} h_y(a_x,t) w(b_y,t)\,\mathrm{d}t + \int_{t^n}^{t^{n+1}} (t - t^n) h_y(a_x,t) \frac{\partial w}{\partial t}(b_y,t)\,\mathrm{d}t, \end{aligned} \tag{31}$$

i.e., the intermediate boundary values for the x-directional splitting scheme (20) are obtained by solving the y-directional scheme (24) backward from t^{n+1} to t^n on the boundaries.

3.4. The FS-ELLAM algorithm

Finally, we can propose our fractional step ELLAM approach (FS-ELLAM) based on the discussions above. Let C^n and C^{n+1} be the numerical solutions at time $t = t^n$ and $t = t^{n+1}$, respectively, and let $C^{(n+1)*}$ be the intermediate step approximation. The FS-ELLAM algorithm can be described as:

Step 1. *Initialization:* $C^0(x_i,y_j) = c_0(x_i,y_j)$, $0 \leqslant i \leqslant I$, $0 \leqslant j \leqslant J$.
Step 2. *For* $n = 0, 1, \ldots, N-1$, Do
  Step 3. *Calculate the intermediate boundary values* $C^{(n+1)*}(a_x,\cdot)$ *with the boundary scheme (31).*
  Step 4. *For all* $\bar{y} = y_j, 1 \leqslant j \leqslant J$, *calculate the intermediate step approximations* $C^{(n+1)*}(x_i,\bar{y})$, $1 \leqslant i \leqslant I$ *by solving the x-directional ELLAM scheme (20) with* $C^n(x_i,\bar{y})$ *being the initial values.*
  Step 5. *For all* $\bar{x} = x_i, 1 \leqslant i \leqslant I$, *Do*

Step 6. *Set the initial values for sweeping in y–direction*: $C^n(\bar{x}, y_j) = C^{(n+1)*}(\bar{x}, y_j)$, $1 \leqslant j \leqslant J$.
Step 7. *Calculate* $(n+1)$*th level approximating solution* $C^{n+1}(\bar{x}, y_j)$, $1 \leqslant j \leqslant J$, *by solving the y-directional ELLAM scheme* (24).
End Do
End Do
Step 8. *Output the approximating solution*: $C^N(x_i, y_j)$, $1 \leqslant i \leqslant I$, $1 \leqslant j \leqslant J$.

Remarks 3.1. In scheme (20), since $C^{(n+1)*} \in V_{hx}^{n+1}$ and $w(x,t) \in V_{hx}^{n+1}$ are standard piecewise linear functions at time t^{n+1}, all terms except the first one on the right side are standard integrals in one-dimensional finite element methods and can be evaluated in a fairly standard way such as using the Gaussian quadratures. Meanwhile, we mention that for $x \geqslant \tilde{a}$, $t^{n+1} - t^*(x) = \Delta t$, and for $a_x \leqslant x < \tilde{a}$, $t^{n+1} - t^*(x)$ is the length of characteristic line tracking backward from (x, t^{n+1}) to the space–time boundary, which can be calculated from the approximate characteristic line $a_x = X(t^*(x); x, t^{n+1}) = x - u_x(a_x, t^{n+1})(t^{n+1} - t^*(x))$, and $\frac{\mathrm{d}t^*(x)}{\mathrm{d}x}$ can be approximated by the relation of $X_x(t^*(x); x, t^{n+1}) = -u_x(a_x, t^*(x))\frac{\mathrm{d}t^*(x)}{\mathrm{d}x}$. We refer interested readers to [4,7,26] for more details of the evaluations of the integration terms.

However, in the first term on the right side, the test functions $w(x, t_+^n) := \lim_{t \to t_+^n} w(x,t) = w(\tilde{x}, t^{n+1})$, where $\tilde{x} = X(t^{n+1}; x, t^n)$ is the point at the head of the characteristic that corresponds to x at the foot. The most practical approach for evaluating this term is to use a forward tracking algorithm proposed by Russell and Trujillo [22]. In this algorithm, the Gaussian quadrature is enforced at time level t^n with respect to the fixed spatial grid on which $C^n(x, \bar{y})$ is defined, and the difficult evaluation is the test function $w(x, t_+^n)$. Rather than backtracking the geometry for valuing the test function, discrete quadrature points chosen on the fixed grids at the time level t^n can be tracked forward to time step level t^{n+1}, where the value of $w(x, t^{n+1})$ is given by the standard base function. The forward tracking algorithm used here does not suffer from the complication of distorted grids. Similarly, we can evaluate the integration terms of the scheme (24). For the intermediate boundary equation (31), the only unknown term on the left side is formed similarly to the first term on the right side of (20) but all other terms on the right side of (31) are evaluated from the given boundary functions by the Gaussian quadratures.

Remarks 3.2. Since the one-dimensional ELLAM method is used to solve the splitting equations, the FS-ELLAM solution technique inherits the advantages of the ELLAM method for convection-dominated diffusion problems. By using a characteristic tracking, the one-dimensional ELLAM schemes (20) and (24) reduce temporal errors for the splitting equations. It generates accurate numerical solutions even through large step sizes are used in computation. The method effectively treats boundary conditions, which is more involved because both the inflow and outflow boundary data can be incorporated into the general formulation. We do not use the exact inflow boundary values at t^{n+1} level for sweeping in x-direction for solving the x-directional splitting equation. Instead, we solve intermediate boundary values $C^{(n+1)*}(a_x, \cdot)$ with the boundary scheme (31). This treatment leads the fractional step ELLAM scheme to be very compatible and to have the same accuracy of temporal errors near the boundaries as that in the interior domain.

Remarks 3.3. Meanwhile, the fractional step technique is applied in the FS-ELLAM approach, which reduces multi-dimensional problems into a series of one-dimensional subproblems. In each time step interval, the one-dimensional ELLAM method is applied to solve the x-directional splitting equations for fixed $\bar{y} = y_j$, $1 \leqslant j \leqslant J$ in Step 4, and the y-directional one-dimensional splitting equations for fixed $\bar{x} = x_i$, $1 \leqslant i \leqslant I$ in Step 5. The corresponding algebraic equation systems for the x-directional splitting equations and the y-directional splitting equations are only symmetric and tridiagonal systems which can be easily solved by the Thomas algorithm. The approach reduces complexities of computation, requirements of large memory, and long computation durations.

4. Numerical experiments

In this section we present numerical experiments for several different convection–diffusion problems. Firstly, in Section 4.1, we investigate the FS-ELLAM for convection–diffusion problems with non-zero source

terms and time-dependent boundary conditions. In Section 4.2, we then simulate numerically the moving of a Gaussian hump in two-dimensions. We will focus on the effects of the methods, the diffusion coefficients and velocity components on the accuracy, the ratio of convergence, and the shape of the approximate moving Gaussian hump. In particular, we show the CPU time comparison by our fractional step ELLAM method (FS-ELLAM) with the two-dimensional ELLAM method (without splitting), the upstream difference method (USDM), the central difference method (CDM), the fractional step USDM method (FS-USDM) and the fractional step CDM method (FS-CDM). Finally, the third experiment in Section 4.3 discusses the problems of moving sharp fronts. The numerical results are compared in detail to show the excellent performance of our FS-ELLAM approach.

Let $c(x, y, t^n)$ be the exact solution of the problem and C^n be the approximate solution, then we calculate errors in L_2-norm and L_∞-norm as:

$$E^n_{\infty,\Delta t} = \max_{i,j}(e^n_{\Delta x,\Delta y,\Delta t}(i,j)) = \max_{i,j}\{|c(x_i,y_j,t^n) - C^n(x_i,y_j)|\}, \tag{32}$$

$$E^n_{2,\Delta t} = \|e^n_{\Delta x,\Delta y,\Delta t}\|_2 = \sqrt{\sum_{i,j}\Delta x\Delta y(c(x_i,y_j,t^n) - C^n(x_i,y_j))^2}, \tag{33}$$

where Δx and Δy are the step sizes in space and Δt is the step size in time. The ratios of convergence in time are calculated by

$$\log\left(\frac{E_{l,\Delta t_1}}{E_{l,\Delta t_2}}\right)\left[\log\left(\frac{\Delta t_1}{\Delta t_2}\right)\right]^{-1}, \quad l = 2, \infty, \tag{34}$$

when very small spatial step sizes Δx and Δy are taken in computation. Similarly, the ratios of convergence in space can be calculated. When the problem in experiment is with unknown exact solution, we will replace $c(x_i, y_j, t^n)$ in the above formulas by the approximate analytical solution $\bar{c}(x_i, y_j, t^n)$ obtained numerically by the two-dimensional ELLAM method (without splitting) (see [26]) with fine mesh.

4.1. Non-zero source problems

In this part, we will consider some convection–diffusion problems with non-zero source terms. The problems have small diffusion coefficients while the velocities involved are relatively large and satisfy Dirichlet in-flow and Neumann out-flow boundary conditions.

We will investigate numerically the problems with our FS-ELLAM proposed last section, the FS-CDM and the FS-USDM. The FS-USDM used here is the fractional step upstream central difference method, i.e., in each time interval $t \in (t^n, t^{n+1}]$, the Euler backward upstream central difference scheme (implicit scheme) is alternately used to solve the x-directional splitting equation and the y-directional splitting equation, in which the upstream difference is used to treat the convection terms of the one-dimensional splitting convection diffusion equations along x-direction and y-direction, respectively. The FS-CDM used here is the fractional step central difference method (Euler backward scheme), in which the central difference is used to discrete the convection terms of the splitting equations. Meanwhile, for the fractional step upstream central difference method (FS-USDM) and the fractional step CDM method (FS-CDM), the intermediate boundary conditions are also designed for the intermediate step splitting schemes in procedures of computation, which are similar to that discussed in Section 3.3 but are solved by the one-dimensional upstream central difference scheme and the one-dimensional central difference scheme along the boundaries, respectively.

Example 1. Firstly, we consider the following convection–diffusion problem with a non-zero source on the spatial region $\Omega = [0, 1]\times[0, 1]$.

$$\frac{\partial c}{\partial t} + \nabla(\vec{u}c) - \nabla(D\nabla c) = f_1(x,y,t), \quad (x,y,t) \in \Omega \times (0,T], \tag{35}$$

where

$$f_1(x,y,t)=\begin{cases} x(0.81-x^2)y(0.81-y^2), & (x,y)\in[0,0.9]\times[0,0.9],\\ 0, & \text{otherwise}\end{cases} \tag{36}$$

with the initial and boundary conditions:

$$\begin{aligned} &c(0,y,t)=1,\quad y\in[0,1],\quad c(x,0,t)=1,\quad x\in[0,1],\quad t\in(0,T],\\ &\frac{\partial c}{\partial x}(1,y,t)=0,\quad y\in[0,1],\quad \frac{\partial c}{\partial y}(x,1,t)=0,\quad x\in[0,1],\quad t\in(0,T],\\ &c(x,y,0)=0,\quad (x,y)\in\Omega. \end{aligned} \tag{37}$$

In this test, the velocity field is taken as $\vec{u}=(3,3)^{\tau}$ and the diffusion coefficient is chosen as $D=1$, 10^{-4} and 10^{-7}. We compute the problem by using our FS-ELLAM as well as the FS-CDM and the FS-USDM. To obtain the ratios in time, we solve the problem with very small spatial step sizes $\Delta x=\Delta y=1/200$, and vary the time step sizes as $\Delta t=1/10$, $1/20$, $1/30$, $1/40$, $1/50$ and $1/60$. Since we do not have the exact solution of this problem, we use the reference analytical solution obtained numerically by the two-dimensional ELLAM method (without splitting) (2D ELLAM) on the refined mesh $\Delta t=1/200$ to calculate errors in L_∞-norm and L_2-norm. In Table 1, we compare the errors and ratios in time for the three numerical methods for the different diffusion coefficients $D=1$, 10^{-4} and 10^{-7} at time $t=1.0$. For the large diffusion coefficient $D=1$, though all of these three numerical methods can obtain first order ratio in time, the errors of the FS-ELLAM are smaller than those of the other two methods. For the convection-dominated diffusion problem with very small diffusion coefficients $D=10^{-4}$ and $D=10^{-7}$, it is clearly shown that the error ratios in time by the FS-ELLAM are still of first order. For example, the ratios in L_∞-norm are 1.3817 and 1.0375 at time step size $\Delta t=1/60$ when the diffusion coefficients are 10^{-4} and $D=10^{-7}$, respectively. When the diffusion coefficients are $D=10^{-4}$ and $D=10^{-7}$, the numerical solutions of the FS-CDM scheme with $\Delta t=1/60$ oscillates. For the FS-USDM scheme, the ratios in time are only less than 0.3-order in L_2-norm and less than 0.1-order in L_∞-norm. Meanwhile, from the table, it is clear that for small diffusion coefficients, the errors in L_∞-norm and L_2-norm are very small even when using large time step sizes. For example, for $D=10^{-7}$, the error in L_∞-norm with a large step size $\Delta t=1/10$ is 7.4676×10^{-3}, which is similar in accuracy with that of using a small step size $\Delta t=1/60$, i.e. 1.2284×10^{-3}. The FS-CDM and the FS-USDM perform very poorly in accuracy for small diffusion coefficients. Numerical results indicate that both the FS-CDM and the FS-USDM cannot handle the convection-dominated diffusion problems well. We can see the FS-ELLAM has first-order ratio in time in both L_∞-norm and L_2-norm for the problems with small diffusions. Thus, we can clearly conclude that the FS-ELLAM scheme is much better than the FS-CDM scheme and FS-USDM scheme for the convection–diffusion problems.

Example 2. Next, we will solve the convection–diffusion problem with a more complicated time-dependent source term and boundary conditions. The source term is

$$f_2(x,y,t)=\cos(t)x(1-x^2)y(1-y^2),\quad (x,y,t)\in[0,1]\times[0,1]\times(0,T] \tag{38}$$

with the in-flow boundary conditions:

$$c(0,y,t)=1+t\sin(t),\quad y\in[0,1],\quad c(x,0,t)=1+t\sin(t),\quad x\in[0,1] \tag{39}$$

and the out-flow boundary and initial conditions are the same as those in Example 1. The velocity is also taken as $\vec{u}=(3,3)^{\tau}$. The diffusion coefficients are taken as $D=1$, 10^{-4} and 10^{-7}. The spatial and time step sizes are the same as those used in Example 1.

Note that the source term $f_2(x, y, t)$ and in-flow boundary conditions depend on time, while in Example 1, they are independent of time. From the computation, we see that, although with more complicated source and in-flow boundary functions, our FS-ELLAM still works very well and is not much affected by the dependence on time. Table 2 shows the errors and ratios in time of these three methods for different diffusion coefficients $D=1$, 10^{-4} and 10^{-7} at time $t=1.0$. It can be seen clearly that the FS-ELLAM is much more accurate than

Table 1
The errors and ratios in time for the convection–diffusion problem with non-zero source $f_1(x, y, t)$ and velocity $\vec{u} = (3, 3)^\tau$

	D	Δt	1/10	1/20	1/30	1/40	1/50	1/60
FS-ELLAM	1	L_∞-error	5.3292E−2	3.0780E−2	2.0718E−2	1.5001E−2	1.1309E−2	8.7301E−3
		Ratio	–	0.7919	0.9764	1.1224	1.2660	1.4196
		L_2-error	2.0704E−2	1.1886E−2	7.9823E−3	5.7757E−3	4.3555E−3	3.3660E−3
		Ratio	–	0.8007	0.9818	1.1248	1.2647	1.4135
	10^{-4}	L_∞-error	3.9000E−2	1.9175E−2	1.2478E−2	8.9213E−3	6.7190E−3	5.2228E−3
		Ratio	–	1.0242	1.0596	1.1664	1.2705	1.3817
		L_2-error	4.0138E−3	2.0581E−3	1.3587E−3	1.0015E−3	7.8536E−4	6.4114E−4
		Ratio	–	0.9637	1.0241	1.0605	1.0893	1.1128
	10^{-7}	L_∞-error	7.4676E−3	3.8522E−3	2.5280E−3	1.8723E−3	1.4842E−3	1.2284E−3
		Ratio	–	0.9550	1.0388	1.0438	1.0410	1.0375
		L_2-error	2.6817E−3	1.4363E−3	9.7893E−4	7.4356E−4	6.0059E−4	5.0479E−4
		Ratio	–	0.9008	0.9456	0.9560	0.9570	0.9531
FS-CDM	1	L_∞-error	9.9081E−2	5.7324E−2	3.9702E−2	2.9695E−2	2.3223E−2	1.8690E−2
		Ratio	–	0.7895	0.9059	1.0095	1.1018	1.1909
		L_2-error	6.1068E−2	3.1616E−2	2.0906E−2	1.5340E−2	1.1926E−2	9.6173E−3
		Ratio	–	0.9498	1.0201	1.0761	1.1283	1.1801
	10^{-4}	L_∞-error	6.2111E−1	6.4708E−1	6.5586E−1	6.5921E−1	6.6024E−1	6.6010E−1
		Ratio	–	–	–	–	–	–
		L_2-error	2.5316E−1	2.2051E−1	2.0193E−1	1.8878E−1	1.7867E−1	1.7055E−1
		Ratio	–	0.1992	0.2172	0.2341	0.2465	0.2553
	10^{-7}	L_∞-error	7.0215E−1	6.8820E−1	6.8737E−1	6.8980E−1	6.9256E−1	6.9491E−1
		Ratio	–	–	–	–	–	–
		L_2-error	2.6307E−1	2.2821E−1	2.0916E−1	1.9608E−1	1.8620E−1	1.7833E−1
		Ratio	–	0.2051	0.2150	0.2245	0.2317	0.2369
FS-USDM	1	L_∞-error	9.7691E−2	5.5517E−2	3.7833E−2	2.7799E−2	2.1313E−2	1.6776E−2
		Ratio	–	0.8153	0.9459	1.0713	1.1906	1.3131
		L_2-error	6.0432E−2	3.0993E−2	2.0294E−2	1.4737E−2	1.1329E−2	9.0266E−3
		Ratio	–	0.9634	1.0443	1.1124	1.1784	1.2462
	10^{-4}	L_∞-error	7.2036E−1	6.6380E−1	6.3825E−1	6.2094E−1	6.0722E−1	5.9657E−1
		Ratio	–	0.1180	0.0968	0.0956	0.1001	0.0971
		L_2-error	3.1674E−1	2.4211E−1	2.1423E−1	1.9788E−1	1.8651E−1	1.7790E−1
		Ratio	–	0.3877	0.3016	0.2760	0.2652	0.2592
	10^{-7}	L_∞-error	7.2431E−1	6.7505E−1	6.5496E−1	6.4206E−1	6.3207E−1	6.2384E−1
		Ratio	–	0.1016	0.0745	0.0691	0.0703	0.0719
		L_2-error	3.2106E−1	2.4769E−1	2.2049E−1	2.0461E−1	1.9362E−1	1.8531E−1
		Ratio	–	0.3743	0.2869	0.2598	0.2476	0.2404

the FS-CDM and the FS-USDM for problems with both small and large diffusion coefficients. For the large diffusion coefficient $D = 1$, the FS-ELLAM has better order of convergence than the FS-CDM and the FS-USDM, and the corresponding ratios in L_∞-norm for these three methods with $\Delta t = 1/60$ are 1.4475, 1.1971, and 1.3345, respectively. For the cases with small diffusion coefficients $D = 10^{-4}$ and $D = 10^{-7}$, numerical results of the FS-CDM and the FS-USDM get much worse. For example, with $\Delta t = 1/60$, the ratios of convergence in L_∞-norm of the FS-USDM are only 0.0945 and 0.0464 for $D = 10^{-4}$ and $D = 10^{-7}$, respectively. There is no ratio in L_∞-norm for the FS-CDM due to oscillation. However, the FS-ELLAM still keeps first-order ratios in time as 1.1640 and 0.9813 in L_∞-norm as well as 1.0217 and 0.9354 in L_2-norm for $D = 10^{-4}$ and $D = 10^{-7}$, respectively.

Example 3. Thirdly, we will consider the problem with a large non-zero source on the spatial region $\Omega = [0, 1] \times [0, 1]$ as:

$$f_3(x, y, t) = 4x(1 - x^2)y(1 - y^2), \quad (x, y, t) \in \Omega \times (0, T], \tag{40}$$

with the in-flow boundary conditions

$$c(0, y, t) = 1.2, \quad y \in [0, 1], \quad c(x, 0, t) = 1.2, \quad x \in [0, 1], \tag{41}$$

Table 2
The errors and ratios in time for the convection–diffusion problem with non-zero source $f_2(x, y, t)$, inflow boundary $g_{in}(t) = 1 + t\sin(t)$ and velocity $\vec{u} = (3, 3)^{\tau}$

	D	Δt	1/10	1/20	1/30	1/40	1/50	1/60
FS-ELLAM	1	L_∞-error	4.9021E−2	2.8569E−2	1.9190E−2	1.3839E−2	1.0387E−2	7.9780E−3
		Ratio	–	0.7790	0.9814	1.1363	1.2857	1.4475
		L_2-error	1.9072E−2	1.1044E−2	7.4129E−3	5.3501E−3	4.0227E−3	3.0978E−3
		Ratio	–	0.7882	0.9832	1.1335	1.2780	1.4329
	10^{-4}	L_∞-error	4.0889E−2	1.9807E−2	1.2872E−2	9.2366E−3	7.2153E−3	5.8356E−3
		Ratio	–	1.0457	1.0629	1.1537	1.1068	1.1640
		L_2-error	5.2444E−3	2.7329E−3	1.8302E−3	1.3692E−3	1.0908E−3	9.0540E−4
		Ratio	–	0.9404	0.9889	1.0088	1.0186	1.0217
	10^{-7}	L_∞-error	1.3808E−2	7.9417E−3	5.5053E−3	4.1988E−3	3.3845E−3	2.8300E−3
		Ratio	–	0.7980	0.9037	0.9416	0.9662	0.9813
		L_2-error	4.3161E−3	2.3035E−3	1.5706E−3	1.1950E−3	9.6775E−4	8.1602E−4
		Ratio	–	0.9059	0.9445	0.9499	0.9454	0.9354
FS-CDM	1	L_∞-error	9.0574E−2	5.2939E−2	3.6802E−2	2.7554E−2	2.1542E−2	1.7318E−2
		Ratio	–	0.7748	0.8967	1.0060	1.1032	1.1971
		L_2-error	5.5108E−2	2.8471E−2	1.8797E−2	1.3763E−2	1.0671E−3	8.5791E−3
		Ratio	–	0.9528	1.0240	1.0836	1.1401	1.1969
	10^{-4}	L_∞-error	6.3162E−1	6.5208E−1	6.5902E−1	6.6143E−1	6.6190E−1	6.6138E−1
		Ratio	–	–	–	–	–	–
		L_2-error	2.4991E−1	2.1901E−1	2.0095E−1	1.8805E−1	1.7810E−1	1.7007E−1
		Ratio	–	0.1904	0.2123	0.2306	0.2437	0.2529
	10^{-7}	L_∞-error	7.2041E−1	6.9953E−1	6.9632E−1	6.9751E−1	6.9948E−1	7.0128E−1
		Ratio	–	–	–	–	–	–
		L_2-error	2.6080E−1	2.2722E−1	2.0854E−1	1.9564E−1	1.8586E−1	1.7807E−1
		Ratio	–	0.1989	0.2115	0.2220	0.2297	0.2351
FS-USDM	1	L_∞-error	8.9152E−2	5.1084E−2	3.4886E−2	2.5612E−2	1.9587E−2	1.5357E−3
		Ratio	–	0.8034	0.9406	1.0742	1.2018	1.3345
		L_2-error	5.4442E−1	2.7813E−2	1.8148E−2	1.3122E−3	1.0037E−3	7.9504E−3
		Ratio	–	0.9690	1.0529	1.1273	1.2011	1.2783
	10^{-4}	L_∞-error	7.1524E−1	6.6068E−1	6.3587E−1	6.1897E−1	6.0552E−1	5.9518E−1
		Ratio	–	0.1145	0.0944	0.0936	0.0984	0.0945
		L_2-error	3.1488E−1	2.4081E−1	2.1329E−1	1.9714E−1	1.8591E−1	1.7739E−1
		Ratio	–	0.3869	0.2993	0.2736	0.2629	0.2572
	10^{-7}	L_∞-error	7.1933E−1	6.7215E−1	6.5279E−1	6.4029E−1	6.3315E−1	6.2782E−1
		Ratio	–	0.0979	0.0721	0.0672	0.0502	0.0464
		L_2-error	3.1922E−1	2.4641E−1	2.1957E−1	2.0390E−1	1.9303E−1	1.8482E−1
		Ratio	–	0.3735	0.2845	0.2574	0.2454	0.2384

and the initial and out-flow boundary conditions same as those in Example 1. The spatial and time step sizes are the same as Example 1 as well.

In this case, the source function $f_3(x, y, t)$ does not vanish near the out-flow boundary, which is different with the first source function $f_1(x, y, t)$ in Example 1. The maximum value of $f_3(x, y, t)$ is larger than that of $f_1(x, y, t)$. The in-flow boundary values are also a little bit bigger than those in Example 1. In Table 3, it has been shown clearly that our FS-ELLAM can treat these types of source terms and boundary conditions well due to the advantages of the ELLAM discussed above. Comparing the numerical results at $t = 1.0$ in Table 3 with those in Tables 1 and 2, we can see clearly that for the small diffusion coefficients, the source term and boundary values affect numerical results of the FS-CDM and the FS-USDM more than those of the FS-ELLAM.

In this subsection, we have compared these numerical methods in terms of errors and ratios in L_∞-norm and L_2-norm for convection diffusion problems with non-zero sources and non-zero inflow boundary conditions. The numerical results show that all of these three numerical methods can obtain first order ratio in time for the large diffusion coefficient $D = 1$. However, for the convection-dominated problems with the small dif-

Table 3
The errors and ratios in time for the convection–diffusion problem with non-zero source $f_3(x, y, t)$ and velocity $\vec{u} = (3, 3)^\tau$

	D	Δt	1/10	1/20	1/30	1/40	1/50	1/60
FS-ELLAM	1	L_∞-error	6.8952E−2	3.7020E−2	2.4169E−2	1.7156E−2	1.2731E−2	9.6814E−3
		Ratio	–	0.8973	1.0516	1.1913	1.3370	1.5019
		L_2-error	2.9185E−2	1.5587E−2	1.0201E−2	7.2898E−3	5.4656E−3	4.2166E−3
		Ratio	–	0.9049	1.0456	1.1679	1.2906	1.4230
	10^{-4}	L_∞-error	8.2437E−2	4.5835E−2	3.0732E−2	2.2903E−2	1.8176E−2	1.4957E−2
		Ratio	–	0.8468	0.9859	1.0221	1.0359	1.0692
		L_2-error	1.7767E−2	9.4586E−3	6.4328E−3	4.8824E−3	3.9443E−3	3.3182E−3
		Ratio	–	0.9095	0.9508	0.9586	0.9562	0.9481
	10^{-7}	L_∞-error	5.5541E−2	3.1947E−2	2.2146E−2	1.6894E−2	1.3617E−2	1.1386E−2
		Ratio	–	0.7979	0.9037	0.9408	0.9664	0.9814
		L_2-error	1.7420E−2	9.2984E−3	6.3371E−3	4.8192E−3	3.9004E−3	3.2868E−3
		Ratio	–	0.9057	0.9456	0.9518	0.9480	0.9387
FS-CDM	1	L_∞-error	1.2899E−1	7.5229E−2	5.2317E−2	3.9309E−2	3.0893E−2	2.4997E−2
		Ratio	–	0.7779	0.8958	0.9937	1.0797	1.1616
		L_2-error	7.9185E−2	4.1286E−2	2.7411E−2	2.0182E−2	1.5743E−2	1.2739E−2
		Ratio	–	0.9396	1.0102	1.0642	1.1133	1.1612
	10^{-4}	L_∞-error	7.2406E−1	7.6080E−1	7.7481E−1	7.8083E−1	7.8339E−1	7.8416E−1
		Ratio	–	–	–	–	–	–
		L_2-error	3.0215E−1	2.6297E−1	2.4101E−1	2.2551E−1	2.1360E−1	2.0401E−1
		Ratio	–	0.2004	0.2150	0.2311	0.2432	0.2520
	10^{-7}	L_∞-error	8.5730E−1	8.4284E−1	8.4210E−1	8.4482E−1	8.4784E−1	8.5039E−1
		Ratio	–	–	–	–	–	–
		L_2-error	3.1811E−1	2.7556E−1	2.5251E−1	2.3679E−1	2.2499E−1	2.1563E−1
		Ratio	–	0.2071	0.2155	0.2234	0.2292	0.2330
FS-USDM	1	L_∞-error	3.0751E−1	7.3148E−2	5.0166E−2	3.7129E−2	2.8699E−2	2.2796E−2
		Ratio	–	2.0718	0.9301	1.0461	1.1541	1.2631
		L_2-error	8.0043E−2	4.0592E−2	2.6732E−2	1.9515E−2	1.5085E−2	1.2089E−2
		Ratio	–	0.9796	1.0302	1.0938	1.1539	1.2143
	10^{-4}	L_∞-error	8.5453E−1	7.8948E−1	7.6033E−1	7.4053E−1	7.2476E−1	7.1249E−1
		Ratio	–	0.1142	0.0928	0.0917	0.0965	0.0937
		L_2-error	3.7469E−1	2.8716E−1	2.5473E−1	2.3568E−1	2.2241E−1	2.1233E−1
		Ratio	–	0.3838	0.2956	0.2701	0.2598	0.2544
	10^{-7}	L_∞-error	8.6074E−1	8.0415E−1	7.8129E−1	7.6664E−1	7.5524E−1	7.4618E−1
		Ratio	–	0.0981	0.0711	0.0658	0.0672	0.0662
		L_2-error	3.7993E−1	2.9393E−1	2.6230E−1	2.4381E−001	2.3098E−1	2.2127E−1
		Ratio	–	0.3703	0.2808	0.2540	0.2423	0.2357

fusion coefficients $D = 10^{-4}$ and $D = 10^{-7}$, we come into conclusion that the FS-ELLAM treats these problems very accurately and has first-order ratio in time in both L_2-norm and L_∞-norm, while the FS-CDM and FS-USDM are impractical and have very low ratios in time (less than 0.3-order in L_2-norm and less than 0.1-order in L_∞ norm).

4.2. The moving Gaussian hump

In this subsection, we consider the moving of the Gaussian pulse of the convection diffusion problem (1)–(4) over a spatial domain $\Omega = [a_x, b_x] \times [a_y, b_y]$ with velocity $\vec{u} = (V_x, V_y)^\tau$ and small diffusion $D > 0$. The initial configuration of the Gaussian pulse is given by

$$c(x, y, 0) = \exp\left(-\frac{(x - x_0)^2 + (y - y_0)^2}{2\sigma_0^2}\right), \quad (x, y) \in \Omega, \tag{42}$$

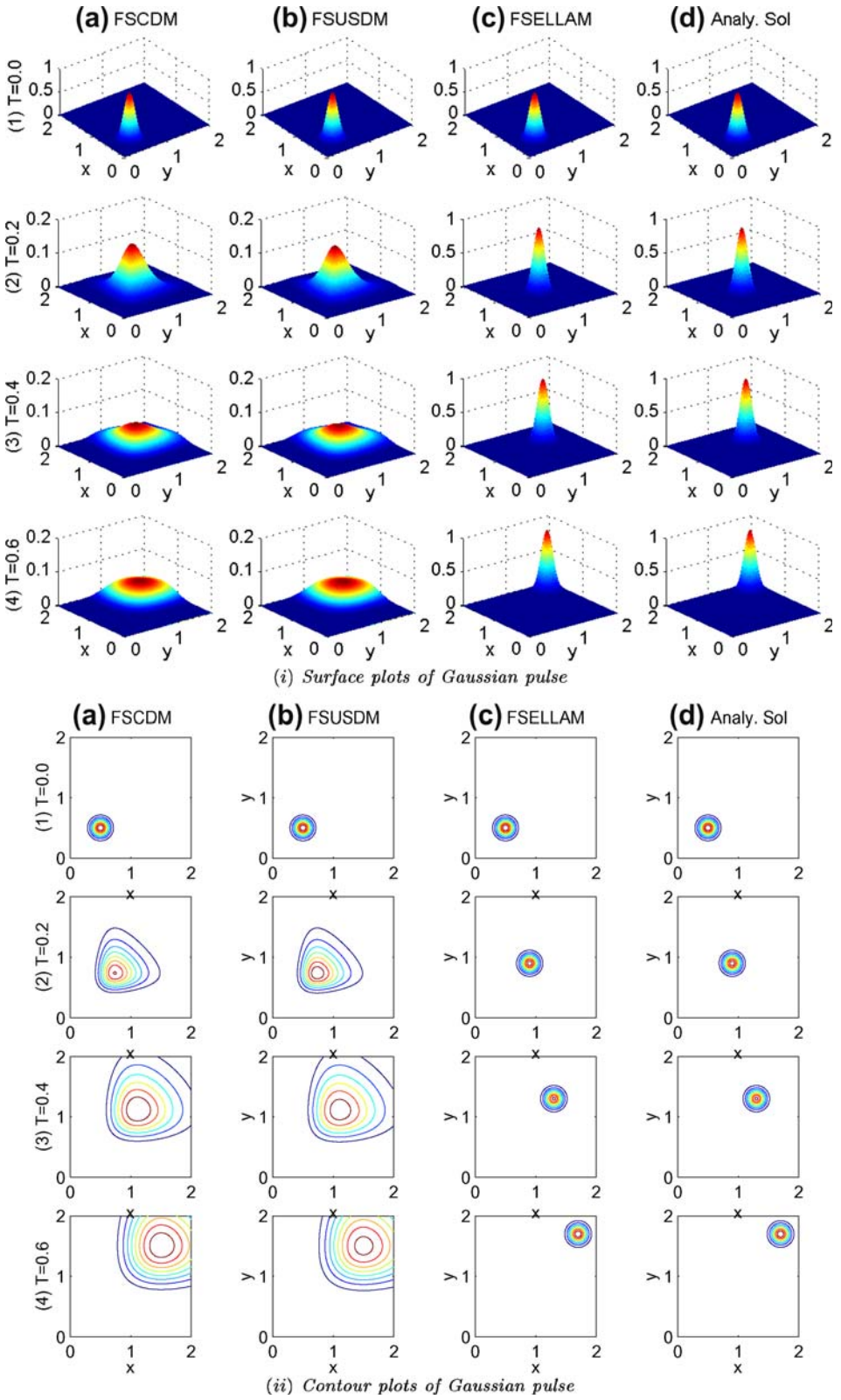


Fig. 2. The moving Gaussian hump with diffusion $D = 0.001$ and velocity $\vec{u} = (2, 2)^{\tau}$.

where $\sigma_0 > 0$ is the initial standard deviation of the Gaussian pulse, and (x_0, y_0) is the initial location of the centre of mass. Let $\Gamma_{\text{in}} := [a_x, b_x]\times\{y = a_y\} \cup [a_y, b_y]\times\{x = a_x\}$ be the in-flow boundary, and let $\Gamma_{\text{out}} := [a_x, b_x]\times\{y = b_y\} \cup [a_y, b_y]\times\{x = b_x\}$ be the out-flow boundary. The boundary conditions are $c(x, y, t) = g(x, y, t), (x, y) \in \Gamma_{\text{in}}, t \in (0, T]$ and $\frac{\partial c(x,y,t)}{\partial n} = h(x, y, t),\ (x, y) \in \Gamma_{\text{out}},\ t \in (0, T]$. Further, $g(x, y, t)$, $h(x, y, t)$ and $f(x, y, t)$ are computed according to the analytical solution of the problem, which is given by:

$$c(x, y, t) = \frac{\sigma_0^2}{\sigma^2(t)} \exp\left(-\frac{(x - x_0 - V_x t)^2 + (y - y_0 - V_y t)^2}{2\sigma^2(t)}\right) \tag{43}$$

with $\sigma(t) = (\sigma_0^2 + 2Dt)^{\frac{1}{2}}$. For this problem, the Gaussian distribution hump centred initially at (x_0, y_0) will propagate in the velocity field and diffuse due to the effect of the small diffusion. There are serious difficulties in simulating the moving procedure of the Gaussian hump numerically (see, for example [14]). In this experiment, we apply our FS-ELLAM scheme to compute the moving Gaussian pulse problem numerically.

Firstly, we carry out numerical computations of the moving Gaussian hump. The velocity field is chosen as $V_x = V_y = 2$ and the diffusion coefficient is taken as $D = 0.1$ or $D = 0.001$. The spatial domain is $\Omega = [0, 2]\times[0, 2]$. The center of the initial Gaussian pulse is specified as $(x_0, y_0) = (0.5, 0.5)$. Take the spatial step sizes $\Delta x = \Delta y = 1/100$ and the time step size $\Delta t = 1/10$. A group of surfaces and contour plots at different times, obtained by the FS-ELLAM, the FS-CDM, and the FS-USDM, display the propagation of the moving Gaussian hump as shown in Fig. 2. The diffusion coefficient is $D = 0.001$. The deviation of the initial Gaussian pulse is $\sigma_0 = 0.1$. The surfaces and contour plots are at four different times $t = 0.0$, 0.2, 0.4 and 0.6. From the surface figures, we can see clearly that the FS-ELLAM (in Fig. 2(i)c) simulates the surface of the Gaussian hump very well while the FS-CDM (in Fig. 2(i)a) and the FS-USDM (in Fig. 2(i)b) smear the hump badly. From the contour plots in Fig. 2(ii)a–d, both the FS-CDM and the FS-USDM schemes deform the shape of the Gaussian hump and change the location of the centre of the Gaussian hump. Our FS-ELLAM perfectly simulates the propagation of the Gaussian pulse and the moving Gaussian pulse (in Fig. 2(ii)c) is in excellent agreement with the pulse of the exact solution (in Fig. 2(ii)d). The errors of the computed peaks for the cases with $D = 0.1$ and $D = 0.001$ are shown in Table 4. For example, with the diffusion coefficient $D = 0.001$ and at time $t = 0.4$, where the exact value of the peak is 0.9259, the computed peak of the FS-ELLAM is 0.9267 while FS-CDM's and FS-USDMs are only 0.0695 and 0.0665, respectively. The error of the peak is about 0.1% for

Table 4
The maximum and minimum of the moving Gaussian pulse with velocity $\vec{u} = (2, 2)^{\tau}$

Diffusion coefficient	Final time	Exact solution		FS-ELLAM		FS-CDM		FS-USDM	
	T	Max.	Min.	Max.	Min.	Max.	Min.	Max.	Min.
$D = 0.1$	0.2	0.2000	0.0	0.2499	0.0	0.1176	0.0	0.1142	0.0
	0.4	0.1111	0.0	0.1289	0.0	0.0494	0.0	0.0479	0.0
	0.6	0.0769	0.0	0.0859	0.0	0.0311	0.0	0.0301	0.0
$D = 0.001$	0.2	0.9615	0.0	0.9619	0.0	0.1608	0.0	0.1548	0.0
	0.4	0.9259	0.0	0.9267	0.0	0.0695	0.0	0.0665	0.0
	0.6	0.8929	0.0	0.8939	0.0	0.0439	0.0	0.0420	0.0

Table 5
The errors of the moving Gaussian pulse with velocity $\vec{u} = (2, 2)^{\tau}$

Diffusion coefficient	Final time	FS-ELLAM		FS-CDM		FS-USDM	
	T	L_∞-error	L_2-error	L_∞-error	L_2-error	L_∞-error	L_2-error
$D = 0.1$	0.2	4.9860E−2	1.0839E−2	1.2670E−1	4.2802E−2	1.2857E−1	4.3498E−2
	0.4	1.0787E−2	5.3690E−3	7.2594E−2	3.2558E−2	7.3658E−2	3.3158E−2
	0.6	8.9961E−3	3.2084E−3	5.0645E−2	2.4578E−2	5.1397E−2	2.5028E−2
$D = 0.001$	0.2	4.3254E−4	4.9118E−5	8.5363E−1	1.4700E−1	8.5826E−1	1.4813E−1
	0.4	7.7407E−4	8.9542E−5	8.7061E−1	1.5561E−1	8.7301E−1	1.5621E−1
	0.6	1.0257E−3	6.7770 E−4	8.5529E−1	1.5580E−1	8.5693E−1	1.5625E−1

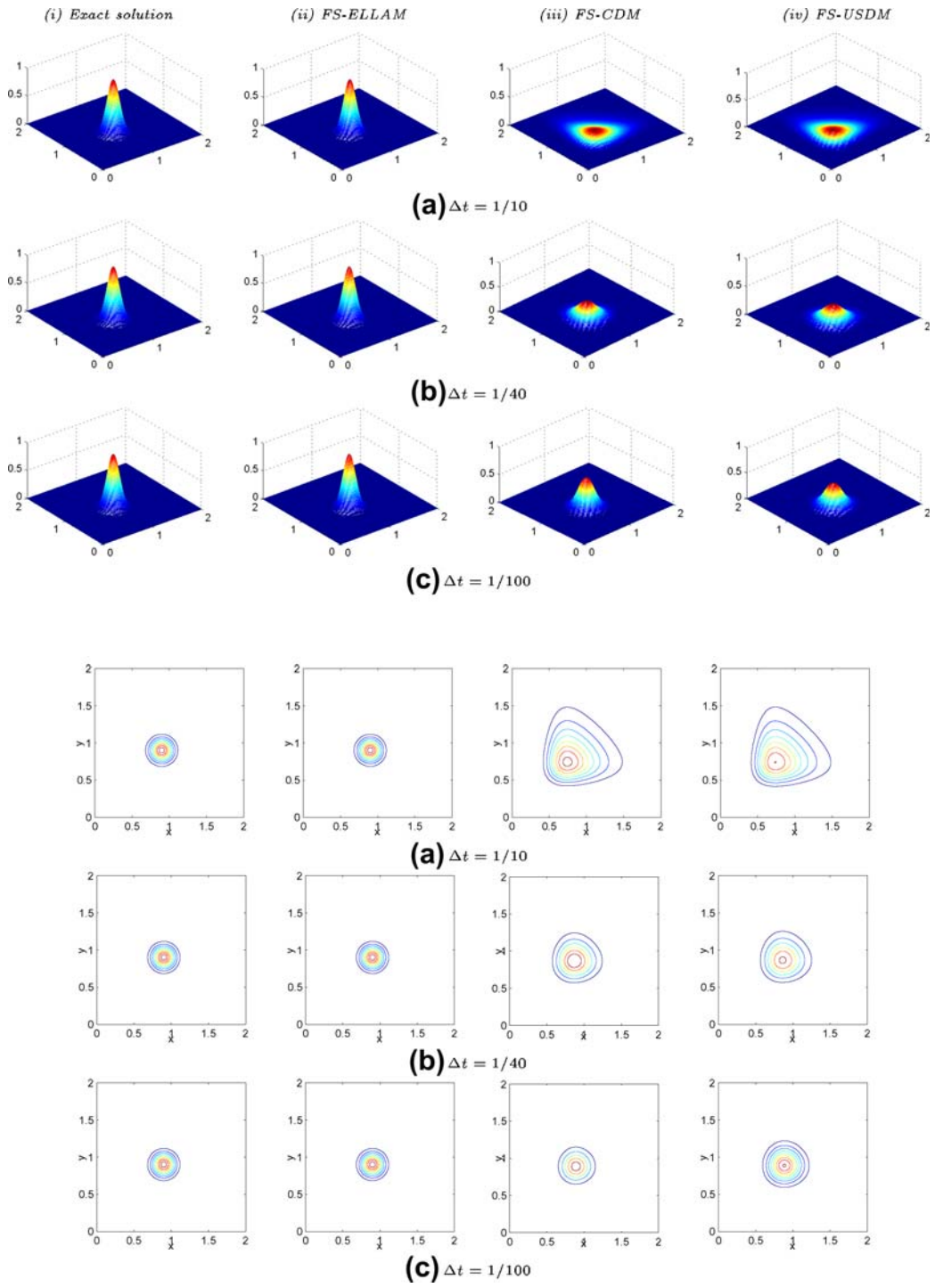


Fig. 3. The surfaces and contour plots of the moving Gaussian pulse with different time step sizes.

the FS-ELLAM but 92% for the FS-CDM and 93% for the FS-USDM. Moreover, numerical errors in L_∞-norm and L_2-norm of these numerical methods are listed in Table 5 for the cases with $D = 0.1$ and $D = 0.001$. For example, for $D = 0.001$ at $t = 0.6$ the errors of the FS-ELLAM in L_∞-norm and L_2-norm

Table 6
The maximum and minimum of the moving Gaussian pulse with velocity $\vec{u} = (2, 2)^{\tau}$

Time step	Exact solution		FS-ELLAM		FS-CDM		FS-USDM	
	Max.	Min.	Max.	Min.	Max.	Min.	Max.	Min.
$\Delta t = \frac{1}{10}$	0.9615	0.0	0.9619	0.0	0.1608	0.0	0.1548	0.0
$\Delta t = \frac{1}{40}$	0.9615	0.0	0.9615	0.0	0.3493	0.0	0.3101	0.0
$\Delta t = \frac{1}{100}$	0.9615	0.0	0.9615	0.0	0.5511	0.0	0.4540	0.0

are 1.0257×10^{-3} and 6.7770×10^{-4} but the FS-CDM and the FS-USDM only obtain accuracies of 8.5529×10^{-1} and 8.5693×10^{-1} in L_∞-norm and of 1.5580×10^{-1} and 1.5625×10^{-1} in L_2-norm, respectively.

Another advantage of our FS-ELLAM is that it can obtain high accuracy even when using large time step sizes while the FS-CDM and the FS-USDM strongly depend on small time step sizes. We show this conclusion by propagating the Gaussian hump to time $t = 0.2$ with different time step sizes $\Delta t = 1/10$,1/40 and 1/100. The deviation of the initial Gaussian pulse is chosen as $\sigma_0 = 0.1$. The numerical results for $D = 0.001$ are shown in the surfaces and contour plots in Fig. 3. Meanwhile, we also compare the computed peak values of these numerical methods in Table 6. The FS-ELLAM has excellent approximations even with large time step sizes. For example, with $\Delta t = 1/10$, the computed maximum of the FS-ELLAM approximation is 0.9619 where the maximum of exact solution is 0.9615, but the maximum of the FS-CDM approximation is only 0.1608 and the FS-USDM approximation only reaches to 0.1548. Both the FS-CDM and the FS-USDM lose most of the peak as shown in Table 6.

Next, we consider the velocity field with different components $V_x = 5$ and $V_y = 0.5$ and with the ratio $|V_x|/|V_y|$ of 10. We show the surfaces and contour plots of the three numerical methods and the exact solutions at three time levels $t = 0.0$, 0.1 and 0.2. In this experiment, the initial deviation is chosen as $\sigma_0 = 0.1$ and the diffusion coefficient is chosen as $D = 0.001$. The spatial step sizes are taken as $\Delta x = \Delta y = 1/100$ and the time step size is taken as $\Delta t = 1/10$. In Fig. 4, figures (ii) are the surfaces and contour plots of our FS-ELLAM, figures (iii) and figures (iv) are ones of the FS-CDM and the FS-USDM while figures (i) are from the exact solution. Due to non-equal velocity components in x-direction and y-direction, the moving Gaussian hump does not propagate along the diagonal but moves much faster along x-direction than along y-direction. From these figures, it is clear that along x-direction where the velocity is much larger, the moving Gaussian humps of the FS-CDM and the FS-USDM badly deform the shape at time $t = 0.1$ and 0.2 while along y-direction, due to the small velocity, the moving Gaussian humps of these two methods do not distort much. However, the moving Gaussian humps of the FS-ELLAM (see figures (ii)) do not perform deformation of the shape, which are same as those of the exact Gaussian humps (see figures (i)). Thus, our FS-ELLAM is robust for the convection-dominated diffusion problem of the moving Gaussian hump. Moreover, the maximum and minimum values of the case with two non-equal velocity components are shown in Table 7, in which similar results are obtained as those in Tables 4 and 6 for the case with two equal velocity components.

In the following part of this subsection, we will show the comparison of errors and ratios of convergence in both time and space of the FS-ELLAM, the FS-CDM and the FS-USDM for solving the moving Gaussian pulse problem. Table 8 shows the errors and ratios of convergence in time. The velocity field is $\vec{u} = (6, 6)^{\tau}$, the diffusion coefficient is chosen as $D = 0.001$. The initial location and deviation are $(x_0, y_0) = (0.3, 0.3)$ and $\sigma_0 = \sqrt{2}/20$, respectively. We simulate the problem on the spatial domain $\Omega = [0, 2] \times [0, 2]$ and the time period $T = 0.2$. To obtain the ratio of convergence in time, we take the small spatial step sizes $\Delta x = \Delta y = 1/200$ and the time step sizes $\Delta t = 1/10$, 1/20, 1/30, 1/40, 1/50 and 1/60. In Table 8, it is clearly shown that the FS-ELLAM has more than first-order ratio of convergence in time while both the FS-CDM and the FS-USDM have very low ratios of less than 0.2. Moreover, we can see that the FS-ELLAM has very high accuracy even with very large time step sizes. For example, when using $\Delta t = 1/10$, the FS-ELLAM only produces an error of 1.7341×10^{-3} in L_∞-norm and an error of 1.4250×10^{-4} in L_2-norm while the errors of the FS-CDM are 9.0041×10^{-1} in L_∞-norm and 1.1623×10^{-1} in L_2-norm, and the errors of the FS-USD are 9.0084×10^{-1} in L_∞-norm and 1.1628×10^{-1} in L_2-norm, respectively.

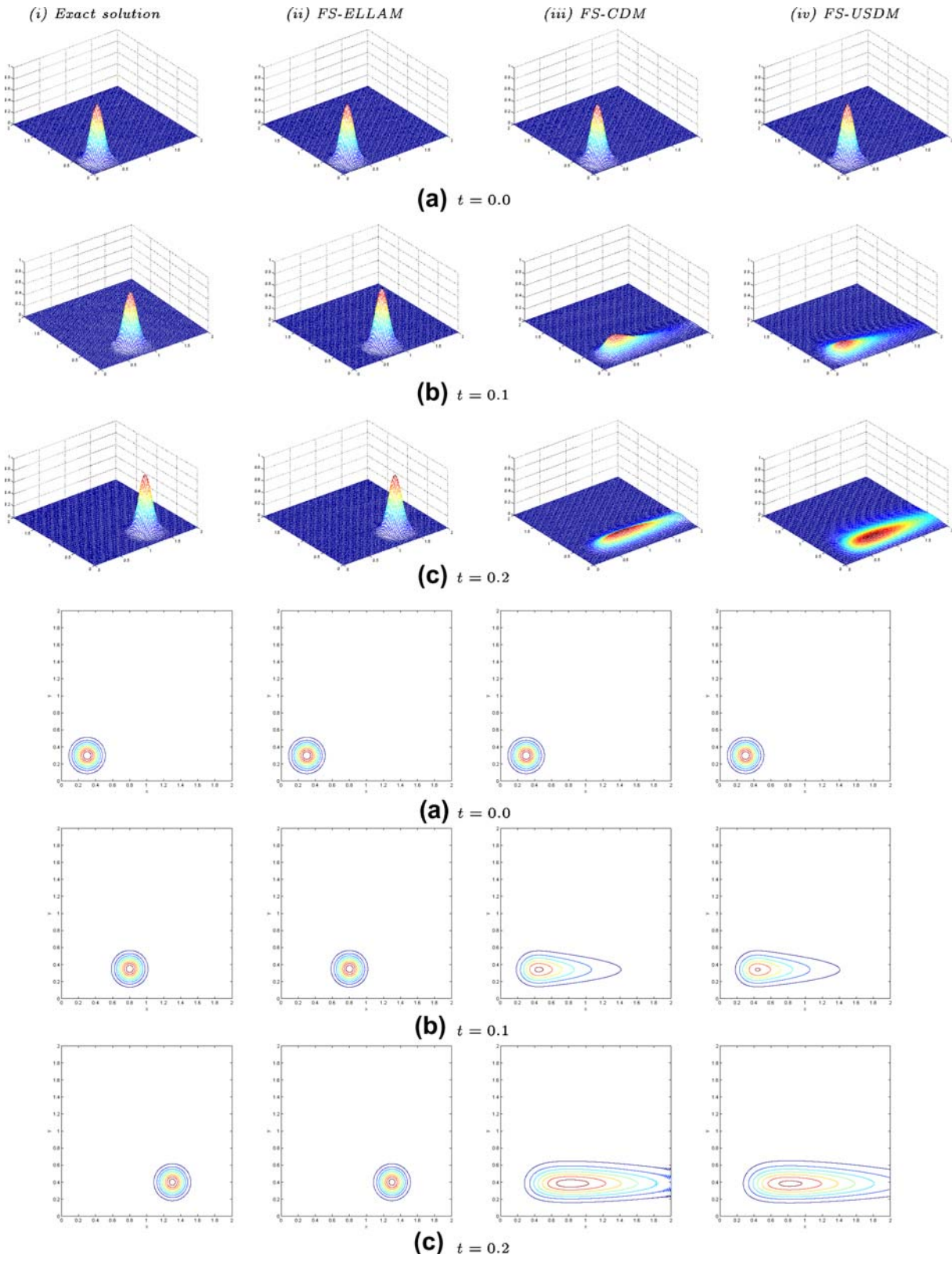


Fig. 4. The surfaces and contour plots of the moving Gaussian pulse with diffusion $D = 0.001$ and velocity $\vec{u} = (5, 0.5)^{\tau}$.

Table 7
The maximum and minimum of the moving Gaussian pulse with velocity $\vec{u} = (5, 0.5)^{\tau}$

Time step	Exact solution		FS-ELLAM		FS-CDM		FS-USDM	
T	Max.	Min.	Max.	Min.	Max.	Min.	Max.	Min.
0.1	0.9804	0.0	0.9806	0.0	0.3103	0.0	0.3038	0.0
0.2	0.9615	0.0	0.9619	0.0	0.1502	0.0	0.1450	0.0

Table 8
The errors and ratios in time of the Gaussian pulse with velocity $\vec{u} = (6, 6)^{\tau}$

Δt	1/10	1/20	1/30	1/40	1/50	1/60
FS-ELLAM						
L_∞-error	1.7341E−3	8.5107E−4	5.4607E−4	3.9207E−4	2.9907E−4	2.3607E−4
Ratio	–	1.0268	1.0944	1.1516	1.2133	1.2974
L_2-error	1.4250E−4	6.9798E−5	4.4842E−5	3.2246E−5	2.4664E−5	1.9632E−5
Ratio	–	1.0297	1.0913	1.1458	1.2015	1.2518
FS-CDM						
L_∞-error	9.0041E−1	8.7441E−1	8.4987E−1	8.2671E−1	8.0481E−1	7.8408E−1
Ratio	–	0.0423	0.0702	0.0960	0.1203	0.1431
L_2-error	1.1623E−1	1.1108E−1	1.0650E−1	1.0234E−1	9.8528E−2	9.5016E−2
Ratio	–	0.0654	0.1038	0.1385	0.1701	0.1990
FS-USDM						
L_∞-error	9.0084E−1	8.7601E−1	8.5325E−1	8.3235E−1	8.1309E−1	7.9526E−1
Ratio	–	0.0403	0.0649	0.0862	0.1049	0.1216
L_2-error	1.1628E−1	1.1136E−1	1.0710E−1	1.0332E−1	9.9936E−2	9.6882E−2
Ratio	–	0.0624	0.0968	0.1249	0.1492	0.1702

In order to compare the errors and ratios of convergence in space for the three method, we use a small time step size $\Delta t = 1/60$ for the FS-ELLAM and even a much smaller time step size $\Delta t = 1/10000$ for both the FS-CDM and the FS-USDM. The spatial step sizes are taken as $\Delta x = \Delta y = 1/10$, 1/20, 1/30, 1/40 and 1/50. Table 9 shows the results of these three methods for the moving Gaussian pulse problem with diffusion coefficient $D = 0.001$. From the table, we can see clearly that the FS-ELLAM has a ratio that is larger than second order in space while the FS-CDM has only a ratio of first order and the FS-USDM has a much lower order (about 0.2). For example, with the spatial step sizes $\Delta x = \Delta y = 1/50$, the ratio of the

Table 9
The errors and ratios in space of Gaussian pulse with velocity $\vec{u} = (6, 6)$

$\Delta x = \Delta y$	1/10	1/20	1/30	1/40	1/50
FS-ELLAM					
L_∞-error	1.2592E−2	7.5119E−3	3.2159E−3	1.6639E−3	9.4792E−4
Ratio	–	0.7452	2.0923	2.2904	2.5215
L_2-error	1.3389E−3	6.1372E−4	2.6044E−4	1.3510E−4	7.7250E−5
Ratio	–	1.1253	2.1140	2.2817	2.5049
FS-CDM					
L_∞-error	8.0702E−1	6.4143E−1	4.8282E−1	3.9162E−1	3.1793E−1
Ratio	–	0.3313	0.7006	0.7277	0.9343
L_2-error	1.4101E−1	1.0630E−1	7.5912E−2	5.3398E−2	3.8446E−2
Ratio	–	0.4076	0.8303	1.2228	1.4721
FS-USDM					
L_∞-error	8.8672E−1	8.5065E−1	8.1778E−1	7.8770E−1	7.6008E−1
Ratio	–	0.0599	0.0972	0.1302	0.1599
L_2-error	1.0898E−1	1.0398E−1	9.9156E−2	9.4532E−2	9.0276E−2
Ratio	–	0.0678	0.1171	0.1659	0.2064

FS-ELLAM in L_∞-norm reaches 2.5215 while the ratios of the FS-CDM and the FS-USDM in L_∞-norm are only 0.9343 and 0.1599, respectively. For the moving Gaussian hump problem, the FS-CDM and the FS-USDM cannot obtain good ratios even when using the very small time step size $\Delta t = 1/10000$. We can also see that even with the larger spatial step sizes $\Delta x = \Delta y = 1/30$, the FS-ELLAM still obtains a second order ratio. Thus, the FS-ELLAM is much more suitable for the convection-dominated diffusion problem of the moving Gaussian hump.

Finally, in the last part of this subsection, we compare the accuracies and CPU times of our FS-ELLAM, the two-dimensional ELLAM method (without splitting) (2D ELLAM), the two-dimensional central difference method with Euler backward time difference (2D CDM), the two-dimensional upstream difference method with Euler backward time difference (2D USDM), the FS-CDM, and the FS-USDM. The FS-ELLAM, the FS-USDM and the FS-CDM lead to a series of tridiagonal systems along x-direction and y-direction, which are solved by the Thomas algorithm. The 2D ELLAM, the 2D USDM and the 2D CDM yield relatively large algebraic equation systems on the two-dimensional domain, which are solved by the preconditioned conjugate gradient square algorithm (PCGS) in experiments. All computations and measurements are taken on a DELL poweredge server 2800 with Dual 3.2G Xeon CPU and 2G memory.

We consider the moving Gaussian hump problem on domain $\Omega = [0, 2] \times [0, 2]$. The initial Gauss hump is located at $(x_0, y_0) = (0.5, 0.5)$. The velocity field is taken as $\vec{u} = (1, 1)^\tau$. The diffusion coefficient is $D = 2.5 \times 10^{-4}$, and the initial deviation is $\sigma_0 = 0.02$. Table 10 shows the errors of numerical solutions (at $t = 1.0$) obtained by the different methods with different step sizes. From the table, we can see that for the moving Gauss hump with the small diffusion coefficient $D = 2.5 \times 10^{-4}$, the 2D ELLAM and the FS-ELLAM have similar high accuracies when using the step sizes $\Delta t = 1/30$ and $\Delta x = \Delta y = 1/90$. The errors of the 2D ELLAM are 3.6520×10^{-3} in L_∞-norm and 1.3600×10^{-4} in L_2-norm and the errors of the FS-ELLAM are 4.7607×10^{-3} in L_∞-norm and 1.6168×10^{-4} in L_2-norm. Moreover, when using the large step sizes $\Delta t = 1/10$ and $\Delta x = \Delta y = 1/60$, the FS-ELLAM still has high accuracy like the 2D ELLAM. The errors of the FS-ELLAM are 7.4309×10^{-3} in L_∞-norm and 2.5837×10^{-4} in L_2-norm and those of the 2D ELLAM are 4.2224×10^{-3} in L_∞-norm and 1.9560×10^{-4} in L_2-norm. As shown, the FS-ELLAM and the 2D ELLAM solutions have similar small errors even with large step sizes. The FS-ELLAM is very accurate, and keeps the advantage of the 2D ELLAM.

In Table 10, we also show the results of the 2D CDM, the 2D USDM, the FS-CDM and the FS-USDM. Even using a very small time step size $\Delta t = 1/1000$, the 2D CDM and the 2D USDM have errors of 2.7962×10^{-1} and 4.1365×10^{-1} in L_∞-norm, respectively, while the FS-ELLAM with a large time step size $\Delta t = 1/10$ still has a much smaller error of 7.4309×10^{-3} in L_∞-norm. Also, the losses of solutions of the 2D CDM and the 2D USDM at the peaks are more than 62% and 90%, respectively (see maximum values in the table). Although the 2D CDM solutions lose less at the peak that those of the 2D USDM, they still suffer oscillations (e.g., the minimum value of 2D CDM solution with $\Delta t = 1/1000$ becomes -0.0480 due to oscillation). The 2D USDM solutions avoid oscillation but smear the hump badly. The errors of the FS-CDM and the FS-USDM are larger than those of the 2D CDM and the 2D USDM, respectively.

Now we compare the CPU time for these methods. We consider a much smaller diffusion coefficient $D = 10^{-4}$. The domain is $\Omega = [0, 3] \times [0, 3]$, the velocity field is $\vec{u} = (1, 1)^\tau$, the initial deviation is $\sigma_0 = 0.02$, and the initial Gauss hump is at $(x_0, y_0) = (0.5, 0.5)$. In this experiment, we use spatial step sizes $\Delta x = \Delta y = 1/60$ and a large time step size $\Delta t = 1/10$ for the 2D ELLAM and the FS-ELLAM. We vary the spatial step sizes from $\Delta x = \Delta y = 1/60$ to $\Delta x = \Delta y = 1/100$ and the time step sizes from $\Delta t = 1/100$ to $\Delta t = 1/3000$ for the 2D CDM, the 2D USDM, the FS-CDM and the FS-USDM in order to get suitable approximation results for comparison. In Table 11, we present the minimum and maximum values of the exact solution, the minimum and maximum values of numerical solutions obtained by numerical methods, the CPU time used per time step by each method, and the overall CPU time consumed by each method at time $t = 2.0$.

In Table 11, the 2D ELLAM and the FS-ELLAM obtain maximum values of 0.4863 and 0.4849, respectively, which are good approximations to the exact maximum value of 0.5 when using a large time step size $\Delta t = 1/10$. In comparison of CPU times, the 2D ELLAM uses a CPU time of 0.373 s per time step and an overall CPU time of 7.83 s, however, our FS-ELLAM reduces the CPU time to only 0.049 s per time step and only consumes an overall CPU time of 1.03 s. The FS-ELLAM is much faster than the 2D ELLAM while still obtaining a similar high accuracy of approximation.

Table 10
Comparison of errors of different methods with $D = 2.5 \times 10^{-4}$ at $t = 1.0$

	Δt	$\Delta x = \Delta y$	L_∞-error	L_2-error	Max. value	Min. value
Exact solution	N/A	N/A	N/A	N/A	$0.444\bar{4}$	0.0
FS-ELLAM	$\frac{1}{10}$	$\frac{1}{60}$	7.4309E − 3	2.5837E − 4	0.4370	0.0
	$\frac{1}{30}$	$\frac{1}{90}$	4.7607E − 3	1.6168E − 4	0.4397	0.0
2D ELLAM	$\frac{1}{10}$	$\frac{1}{60}$	4.2224E − 3	1.9560E − 4	0.4402	0.0
	$\frac{1}{30}$	$\frac{1}{90}$	3.6520E − 3	1.3600E − 4	0.4408	0.0
2D CDM	$\frac{1}{100}$	$\frac{1}{60}$	3.7446E − 1	1.9839E − 2	0.0768	−0.0166
	$\frac{1}{200}$	$\frac{1}{60}$	3.5798E − 1	1.9091E − 2	0.0977	−0.0256
	$\frac{1}{400}$	$\frac{1}{60}$	3.4293E − 1	1.8814E − 2	0.1216	−0.0390
	$\frac{1}{1000}$	$\frac{1}{60}$	3.3162E − 1	1.9752E − 2	0.1522	−0.0613
	$\frac{1}{100}$	$\frac{1}{90}$	3.6115E − 1	1.9089E − 2	0.0869	−0.0066
	$\frac{1}{200}$	$\frac{1}{90}$	3.3608E − 1	1.7587E − 2	0.1150	−0.0138
	$\frac{1}{400}$	$\frac{1}{90}$	3.0925E − 1	1.6107E − 2	0.1483	−0.0252
	$\frac{1}{1000}$	$\frac{1}{90}$	2.7962E − 1	1.5190E − 2	0.1949	−0.0480
2D USDM	$\frac{1}{100}$	$\frac{1}{60}$	4.2892E − 1	2.2423E − 2	0.0156	0.0
	$\frac{1}{200}$	$\frac{1}{60}$	4.2631E − 1	2.2223E − 2	0.0182	0.0
	$\frac{1}{400}$	$\frac{1}{60}$	4.2440E − 1	2.2077E − 2	0.0201	0.0
	$\frac{1}{1000}$	$\frac{1}{60}$	4.2292E − 1	2.1964E − 2	0.0215	0.0
	$\frac{1}{100}$	$\frac{1}{90}$	4.2409E − 1	2.2062E − 2	0.0204	0.0
	$\frac{1}{200}$	$\frac{1}{90}$	4.1988E − 1	2.1741E − 2	0.0246	0.0
	$\frac{1}{400}$	$\frac{1}{90}$	4.1650E − 1	2.1485E − 2	0.0280	0.0
	$\frac{1}{1000}$	$\frac{1}{90}$	4.1365E − 1	2.1272E − 2	0.0307	0.0
FS-CDM	$\frac{1}{100}$	$\frac{1}{60}$	4.0909E − 1	2.0971E − 2	0.0368	−1.2E − 4
	$\frac{1}{200}$	$\frac{1}{60}$	3.8475E − 1	1.9370E − 2	0.0652	−0.0034
	$\frac{1}{400}$	$\frac{1}{60}$	3.6052E − 1	1.8341E − 2	0.1031	−0.0190
	$\frac{1}{1000}$	$\frac{1}{60}$	3.4166E − 1	1.9295E − 2	0.1467	−0.0531
	$\frac{1}{100}$	$\frac{1}{90}$	4.0819E − 1	2.0885E − 2	0.0370	−1.8E − 7
	$\frac{1}{200}$	$\frac{1}{90}$	3.7917E − 1	1.8866E − 2	0.0675	−2.0E − 4
	$\frac{1}{400}$	$\frac{1}{90}$	3.4045E − 1	1.6485E − 2	0.1130	−0.0054
	$\frac{1}{1000}$	$\frac{1}{90}$	2.9644E − 1	1.4961E − 2	0.1810	−0.0336
FS-USDM	$\frac{1}{100}$	$\frac{1}{60}$	4.2997E − 1	2.2503E − 2	0.0146	0.0
	$\frac{1}{200}$	$\frac{1}{60}$	4.2676E − 1	2.2257E − 2	0.0178	0.0
	$\frac{1}{400}$	$\frac{1}{60}$	4.2455E − 1	2.2088E − 2	0.0200	0.0
	$\frac{1}{1000}$	$\frac{1}{60}$	4.2295E − 1	2.1967E − 2	0.0215	0.0
	$\frac{1}{100}$	$\frac{1}{90}$	4.2631E − 1	2.2223E − 2	0.0183	0.0
	$\frac{1}{200}$	$\frac{1}{90}$	4.2096E − 1	2.1818E − 2	0.0236	0.0
	$\frac{1}{400}$	$\frac{1}{90}$	4.1691E − 1	2.1515E − 2	0.0276	0.0
	$\frac{1}{1000}$	$\frac{1}{90}$	4.1374E − 1	2.1279E − 2	0.0307	0.0

The FS-CDM, the FS-USDM, the 2D CDM and the 2D USDM work very badly in the aspect of CPU times. For example, with a small time step $\Delta t = 1/500$ and small spatial steps $\Delta x = \Delta y = 1/60$, the 2D CDM and the FS-CDM solutions have maximum values of only 0.0856 and 0.0695 and negative minimum values of -0.0334 and -0.0149, respectively, which are excessively overdamped. Meanwhile, the 2D CDM solution needs a much longer overall CPU time of 84.55 s and the FS-CDM solution needs an overall CPU time of 24.10 s. Both are much longer than those used by the FS-ELLAM and the 2D ELLAM. Even with a very small time step size $\Delta t = 1/1000$, the 2D CDM and the FS-CDM solutions have only maximum values of 0.1029 and 0.0963 and negative minimum values of -0.0744 and -0.0665, respectively. The maximum values are still very diffusive comparing to the exact maximum value of 0.5.

For observing the improvement of numerical solutions, we also reduce the spatial step sizes to $\Delta x = \Delta y = 1/100$ and the time step size to $\Delta t = 1/3000$. Both the 2D CDM and the FS-CDM have essentially no improvement in accuracy, however, the overall CPU times have been significantly increased to 1007.16 s and 404.80 s, respectively. Due to the upstream technique, both the 2D USDM and the FS-USDM are

Table 11
Comparison of CPU times of different methods with $D = 10^{-4}$ at $t = 2.0$

	Δt	$\Delta x = \Delta y$	Max. value	Min. value	CPU (s)/each step	Overall CPU
Exact solution	N/A	N/A	0.5000	0.0	N/A	N/A
FS-ELLAM	$\frac{1}{10}$	$\frac{1}{60}$	0.4849	0.0	0.049	1.03
2D ELLAM	$\frac{1}{10}$	$\frac{1}{60}$	0.4863	0.0	0.373	7.83
2D CDM	$\frac{1}{100}$	$\frac{1}{60}$	0.0513	−0.0144	0.210	42.30
	$\frac{1}{200}$	$\frac{1}{60}$	0.0646	−0.0213	0.115	46.09
	$\frac{1}{500}$	$\frac{1}{60}$	0.0856	−0.0334	0.084	84.55
	$\frac{1}{1000}$	$\frac{1}{60}$	0.1029	−0.0467	0.071	143.31
	$\frac{1}{1000}$	$\frac{1}{100}$	0.1466	−0.0425	0.208	417.82
	$\frac{1}{2000}$	$\frac{1}{100}$	0.1754	−0.0620	0.179	717.54
	$\frac{1}{3000}$	$\frac{1}{100}$	0.1906	−0.0744	0.168	1007.16
2D USDM	$\frac{1}{200}$	$\frac{1}{60}$	0.0093	0.0	0.119	47.54
	$\frac{1}{500}$	$\frac{1}{60}$	0.0106	0.0	0.086	86.34
	$\frac{1}{1000}$	$\frac{1}{60}$	0.0112	0.0	0.078	156.62
	$\frac{1}{1000}$	$\frac{1}{100}$	0.0176	0.0	0.241	483.99
	$\frac{1}{2000}$	$\frac{1}{100}$	0.0184	0.0	0.207	830.94
	$\frac{1}{3000}$	$\frac{1}{100}$	0.0186	0.0	0.142	854.13
FS-CDM	$\frac{1}{100}$	$\frac{1}{60}$	0.0192	−1.7559E − 5	0.024	4.84
	$\frac{1}{200}$	$\frac{1}{60}$	0.0361	−0.0010	0.024	9.66
	$\frac{1}{500}$	$\frac{1}{60}$	0.0695	−0.0149	0.024	24.10
	$\frac{1}{1000}$	$\frac{1}{60}$	0.0963	−0.0362	0.024	48.33
	$\frac{1}{1000}$	$\frac{1}{100}$	0.1246	−0.0204	0.069	137.93
	$\frac{1}{2000}$	$\frac{1}{100}$	0.1664	−0.0493	0.066	263.19
	$\frac{1}{3000}$	$\frac{1}{100}$	0.1848	−0.0665	0.067	404.80
FS-USDM	$\frac{1}{200}$	$\frac{1}{60}$	0.0091	0.0	0.023	9.10
	$\frac{1}{500}$	$\frac{1}{60}$	0.0105	0.0	0.023	22.85
	$\frac{1}{1000}$	$\frac{1}{60}$	0.0111	0.0	0.023	45.69
	$\frac{1}{1000}$	$\frac{1}{100}$	0.0175	0.0	0.063	126.37
	$\frac{1}{2000}$	$\frac{1}{100}$	0.0183	0.0	0.063	252.93
	$\frac{1}{3000}$	$\frac{1}{100}$	0.0186	0.0	0.065	399.08

oscillation free but both solutions are much more diffusive. With the small time step size $\Delta t = 1/500$, the 2D USDM and the FS-USDM have only maximum values of 0.0106 and 0.0105, respectively. The 2D USDM and the FS-USDM still need much more overall CPU times of 86.34 s and 22.85 s, respectively. Thus, from Table 11, we can see that the FS-ELLAM requires much less CPU time compared to the ELLAM while still obtaining the same accuracy of approximation. The other methods, the 2D CDM, the FS-CDM, the 2D USDM and the FS-USDM, are not comparable with the FS-ELLAM in accuracy and CPU time.

4.3. The moving sharp front

In the last subsection, we compute the moving steep front problem with our FS-ELLAM. The numerical results show that the FS-ELLAM simulates the steep front very well even if the time step size taken is very large. The FS-CDM and the FS-USDM give either a too wide representation of the steep front or obvious oscillations near the front.

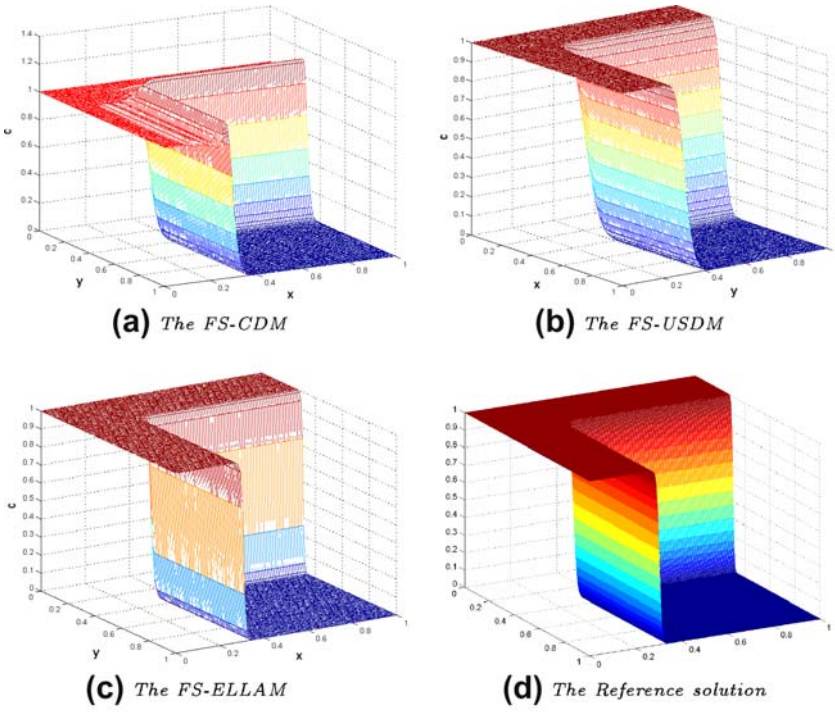
Example 4. We consider the following convection–diffusion problem on the spatial domain $\Omega = [0, 1] \times [0, 1]$:

$$\frac{\partial c}{\partial t} + \nabla(\vec{u}c) - \nabla(D\nabla c) = 0, \quad (x, y, t) \in \Omega \times (0, T], \tag{44}$$

$$c(x, y, 0) = c_0(x, y), \quad (x, y) \in \Omega, \tag{45}$$

$$c(0, y, t) = 1, \quad y \in [0, 1], \quad c(x, 0, t) = 1, \quad x \in [0, 1], \quad t \in (0, T], \tag{46}$$

$$\frac{\partial c(1, y, t)}{\partial x} = 0, \quad y \in [0, 1], \quad \frac{\partial c(x, 1, t)}{\partial y} = 0, \quad x \in [0, 1], \quad t \in (0, T], \tag{47}$$



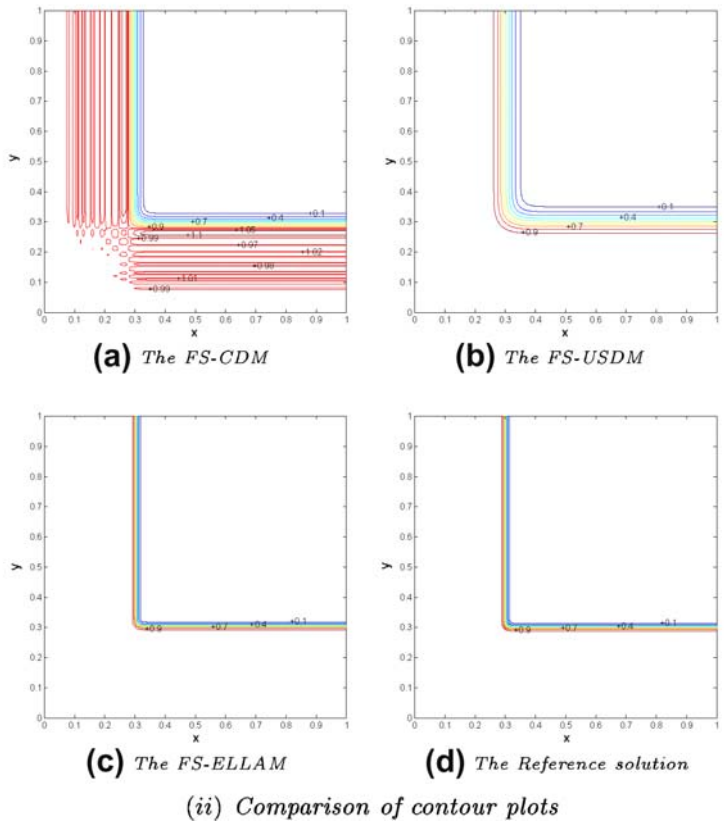


Fig. 5. The surfaces and contour plots of the moving sharp front by different numerical methods.

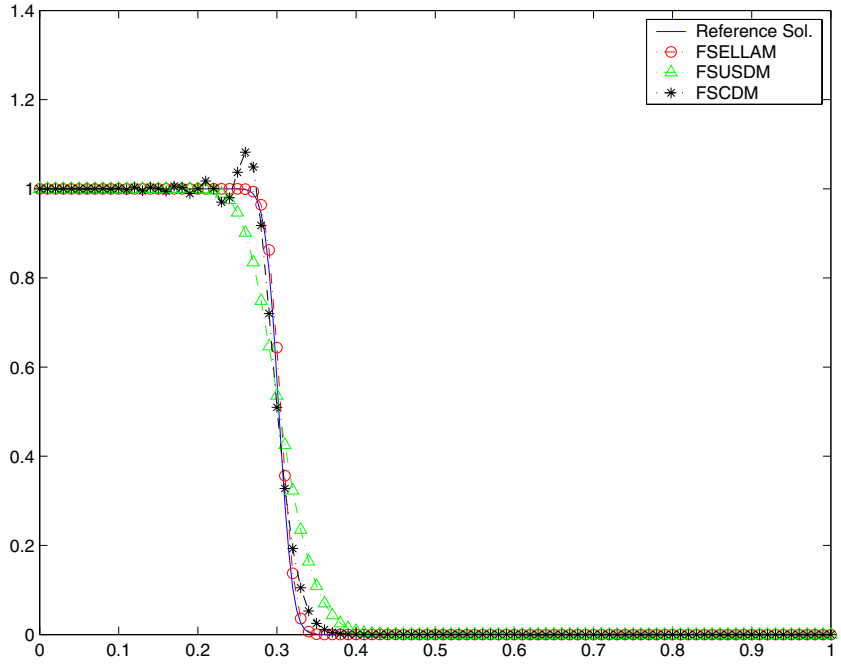


Fig. 6. The comparison of solution curves of different numerical methods on a sectional plane.

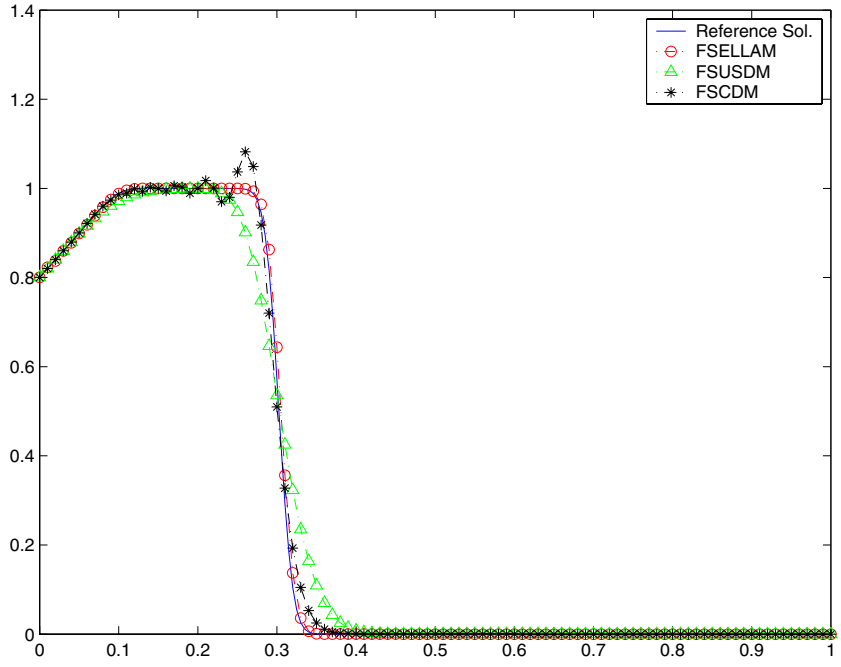


Fig. 7. The comparison of the moving sharp front with inflow boundary condition $g(t) = 1 - 2\sin t$ of different numerical methods on a sectional plane.

where the initial value $c_0(x, y)$ is given as

$$c_0(x,y) = \begin{cases} 1, & 0 \leqslant x,\ y \leqslant 0.2, \\ 0, & \text{otherwise} \end{cases} \tag{48}$$

and the diffusion coefficient and the velocity are taken as $D = 0.001$ and $\vec{u} = (1, 1)^{\tau}$. Since there is a jump in the initial function, a moving sharp front will occur at the jump as the time increases. This makes it difficult to simulate numerically the solution by the traditional numerical methods.

The surfaces and contour plots of the approximate solutions at time $t = 0.1$ computed by the FS-ELLAM, the FS-CDM, the FS-USDM and the reference analytical solution are presented in Fig. 5. The spatial step sizes are taken as $\Delta x = \Delta y = 1/100$. The approximation solution of our FS-ELLAM (Fig. 5c) is obtained with a time step size $\Delta t = 1/50$ while the solutions of the FS-CDM and the FS-USDM (Fig. 5a and b) are both obtained with a smaller time step size $\Delta t = 1/1000$. The reference analytical solution, in Fig. 5d, is computed by the two-dimensional ELLAM (without splitting) with small step sizes $\Delta x = \Delta y = 1/200$ and $\Delta t = 1/100$. From Fig. 5, it is clear that the FS-CDM performs oscillation near the front and the FS-USDM overcomes oscillation with the upstream technique but smears the front even when using the smaller time step size. Particularly, from the contour plots in Fig. 5(ii), the numerical solution of the FS-CDM does not decay monotonically near the sharp front, instead, it oscillates between 1.05 and 0.97 before the front and the width of the front is very large. The FS-USDM yields a solution with excessive numerical diffusion, and its front width is larger than that of the reference solution. Our FS-ELLAM approximates the moving sharp front much more accurately without oscillation and smearing in Fig. 5c, which is in excellent agreement with the reference analytical solution in Fig. 5d. Furthermore, the detailed comparison of the solution curves of these three numerical solutions and the reference analytical solution on a sectional plane along x-direction at $y = 1$ is shown in Fig. 6. The results show clearly that the FS-ELLAM scheme simulates the front correctly and accurately while the FS-CDM and the FS-USDM perform oscillation or smear the front.

Example 5. Furthermore, we consider the moving sharp front problem with more complicated boundary conditions: $c(0, y, t) = 1 - 2\sin(t)$, $y \in [0, 1]$, $c(x, 0, t) = 1 - 2\sin(t)$, $x \in [0, 1]$, $t \in (0, T]$, and $\frac{\partial c(1,y,t)}{\partial x} = 0$, $y \in [0, 1]$, $\frac{\partial c(x,1,t)}{\partial y} = 0$, $x \in [0, 1]$, $t \in (0, T]$. The diffusion coefficient is $D = 0.001$ and the velocity is $\vec{u} = (1, 1)^{\tau}$. The initial value is same as Example 4.

In Fig. 7 the comparison of the numerical solution curves of these numerical methods at $t = 0.1$ is shown along a sectional plane of x-direction at $y = 1$. The spatial step sizes are $\Delta x = \Delta y = 1/100$. The FS-ELLAM solution is obtained with a time step size $\Delta t = 1/50$, and the FS-CDM and the FS-USDM solutions are both obtained with a smaller time step size $\Delta t = 1/1000$. The reference analytical solution is computed by the two-dimensional ELLAM (without splitting) on the fine mesh $\Delta x = \Delta y = 1/200$ and $\Delta t = 1/100$. The FS-CDM performs oscillation and the FS-USDM overcomes oscillation by the upstream technique but smear the front. It can been seen clearly that the FS-ELLAM approximates the sharp front much more accurately without oscillation and smearing.

5. Conclusion

We developed a fractional step ELLAM algorithm (FS-ELLAM) for two-dimensional convection–diffusion problems in this paper. The method reduces the high-dimensional problems to a series of uncoupled one-dimensional problems in each time step interval, in which one-dimensional Eulerian–Lagrangian localized adjoint method (ELLAM) is used to solve the one-dimensional splitting equations.

The developed FS-ELLAM approach takes the attractive advantages of both the ELLAM method and the fractional step technique. It reduces computational complexities, large memory requirements, and long computation durations due to the application of the splitting technique. By inheriting the feature of the ELLAM method, it significantly reduces temporal truncation errors and generates accurate numerical solutions even when large time and coarse spatial step sizes are used in computation. It effectively eliminates non-physical oscillation or excessive numerical dispersion and treats boundary conditions in a natural way. The numerical experiments included convection–diffusion problems with source terms, moving Gaussian hump problems and moving sharp front problems. At each time level, the FS-ELLAM approach applied the one-dimensional ELLAM method to solve the x-directional and y-directional splitting problems. The corresponding algebraic equation systems are symmetric and tridiagonal systems which can be easily solved.

Numerical results and comparisons indicate that the proposed FS-ELLAM is very efficient and suitable to solve the convection-dominated diffusion problems. The ELLAM scheme greatly eliminates numerical oscillation by using the tracking technique along the characteristics. We would like to mention that a monotonic ELLAM scheme has recently been studied by Neubauer and Bastian in [20].

The algorithm of FS-ELLAM developed in this paper can be easily extended to three- and higher-dimensional convection–diffusion problems with more general boundary conditions. The procedure can be solved by a parallel computing system.

Acknowledgements

The authors thank the referees and the editor for their invaluable comments and suggestions which have helped to improve the paper greatly. This work was supported by Natural Sciences and Engineering Research Council of Canada.

References

[1] K. Aziz, A. Settari, Petroleum Reservoir Simulation, Applied Science Publisher Ltd., London, 1979.
[2] J. Bear, Hydraulics of Groundwater, McGraw-Hill, New York, 1978.
[3] P. Binning, M.A. Celia, A finite volume Eulerian–Lagrangian localized adjoint method for solution of the contaminant transport equations in two-dimensional multi-phase flow systems, Water Resour. Res. 32 (1996) 103–114.
[4] M.A. Celia, T.F. Russell, I. Herrera, R.E. Ewing, An Eulerian–Lagrangian localized adjoint method for the advection–diffusion equation, Adv. Water Resour. 4 (1990) 187–206.
[5] C. Clavero, J.C. Jorge, F. Lisbona, G.I. Shishkin, A fractional step method on a special mesh for the resolution of multidimensional evolutionary convection–diffusion problems, Appl. Numer. Math. 27 (1998) 211–231.
[6] M.G. Crandall, A. Majda, The method of fractional steps for conservation laws, Numer. Math. 34 (1980) 285–314.
[7] H.K. Dahle, R.E. Ewing, T.F. Russell, Eulerian–Lagrangian localized adjoint methods for a nonlinear advection–diffusion equation, Comput. Methods Appl. Mech. Eng. 122 (1995) 223–250.
[8] W. Dai, N. Raja, A new ADI scheme for solving three-dimensional parabolic equations with first-order derivatives and variable coefficients, J. Comput. Anal. Appl. 2 (2000) 125–146.
[9] J. Douglas Jr., H.H. Rachford, On the numerical solution of heat conduction problems in two and three space variables, Trans. Am. Math. Soc. 82 (1956) 421–439.
[10] J. Douglas Jr., S. Kim, Improved accuracy for locally one-dimensional methods for parabolic equations, Math. Models Methods Appl. Sci. 11 (2001) 1563–1579.
[11] R.E. Ewing (Ed.), The Mathematics of Reservoir Simulation, Frontiers in Applied Mathematics, vol. 1, 1988, pp. 294–318.
[12] S.K. Godunov, Finite difference methods for numerical computation of discontinuous solutions of the equations of fluid dynamics, Mat. Sbornik 47 (1959) 271–295.
[13] R.W. Healy, T.F. Russell, Solution of the advection–diffusion equation in two dimension by a finite-volume Eulerian–Lagrangian localized adjoint method, Adv. Water Resour. 21 (1998) 11–26.
[14] K.H. Karlsen, K.-A. Lie, An unconditionally stable splitting scheme for a class of nonlinear parabolic equations, IMA J. Numer. Anal. 19 (1999) 1–28.
[15] R.B. Kellogg, S. Shih, Asymptotic analysis of a singular perturbation problem, SIAM J. Math. Anal. 18 (1987) 1467–1511.
[16] L.A. Khan, P.L.-F. Liu, Numerical analyses of operator splitting algorithm for the two-dimensional advection–diffusion equation, Comput. Methods Appl. Mech. Eng. 152 (1998) 337–359.
[17] R.J. LeVeque, Intermediate boundary conditions for time split methods applied to hyperbolic partial differential equations, Math. Comput. 47 (1986) 37–54.
[18] D. Liang, W. Zhao, The high-order upwind scheme for the convection–diffusion problem, Comput. Methods Appl. Mech. Eng. 147 (1997) 105–115.
[19] G.I. Marchuk, Splitting and Alternating Direction MethodsHandbook of Numerical Analysis, vol. 1, North-Holland, Amsterdam, 1990.
[20] T. Neubauer, P. Bastian, On a monotonicity preserving Eulerian–Lagrangian localized adjoint method for advection–diffusion equations, Adv. Water Resour. 28 (2005) 1292–1309.
[21] M. Reeves, R.M. Cranwell, User's Manual for The Sandia Waste-Isolation Flow and Transport Model (SWIFT) Release 4.81, Sandia Report Nureg/CR-2324, SAND 81-2516, GR, Albuquerque, NM, 1981.
[22] T.F. Russell, R.V. Trujillo, Eulerian–Lagrangian localized adjoint methods with variable coefficients in multiple dimensions, in: G. Gambolati et al. (Eds.), Computational Methods in Surface Hydrology, Springer, Berlin, 1990, pp. 357–363.
[23] B.P. Sommeijer, P.J. van der Houwen, J.G. Vewer, On the treatment of time-dependent boundary conditions in splitting methods for parabolic differential equations, Int. J. Numer. Methods Eng. 17 (1981) 335–346.
[24] N. Yanenko, The Method of Fractional Steps, Springer, Berlin/Heidelberg/New York, 1971.

[25] G. Strang, On the construction and comparison of difference schemes, SIAM J. Numer. Anal. 5 (1968) 506–517.
[26] H. Wang, H.K. Dahle, R.E. Ewing, M.S. Espedal, R.C. Sharpley, S. Man, An Eulerian–Lagrangian localized adjoint method for two-dimensional advection–diffusion equations and its comparison to other schemes, SIAM J. Sci. Comput. 20 (1999) 2160–2194.
[27] H. Wang, D. Liang, R.E. Ewing, S.L. Lyons, G. Qin, An approximation to miscible fluid flows in porous media with point sources and sinks by an Eulerian–Lagrangian localized adjoint method and mixed finite element methods, SIAM J. Sci. Comput. 22 (2000) 561–581.
[28] H. Wang, D. Liang, R.E. Ewing, S.L. Lyons, G. Qin, An ELLAM approximation for highly compressible multicomponent flows in porous media, Comput. Geosci. 6 (2002) 227–251.