Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp



An iteration free backward semi-Lagrangian scheme for solving incompressible Navier–Stokes equations *



Xiangfan Piao^a, Sunyoung Bu^b, Soyoon Bak^b, Philsu Kim^{b,*}

^a Department of Mathematics, Hannam University, Daejeon 306-791, Republic of Korea

^b Department of Mathematics, Kyungpook National University, Daegu 702-701, Republic of Korea

ARTICLE INFO

Article history: Received 27 March 2014 Received in revised form 7 November 2014 Accepted 15 November 2014 Available online 5 December 2014

Keywords: Semi-Lagrangian method Error correction method Projection method Fourth-order finite difference method Navier–Stokes equations

ABSTRACT

A backward semi-Lagrangian method based on the error correction method is designed to solve incompressible Navier–Stokes equations. The time derivative of the Stokes equation is discretized with the second order backward differentiation formula. For the induced steady Stokes equation, a projection method is used to split it into velocity and pressure. Fourth-order finite differences for partial derivatives are used to the boundary value problems for the velocity and the pressure. Also, finite linear systems for Poisson equations and Helmholtz equations are solved with a matrix-diagonalization technique. For characteristic curves satisfying highly nonlinear self-consistent initial value problems, the departure points are solved with an error correction strategy having a temporal convergence of order two. The constructed algorithm turns out to be completely iteration free. In particular, the suggested algorithm possesses a good behavior of the total energy conservation compared to existing methods. To assess the effectiveness of the method, two-dimensional lid-driven cavity problems with large different Reynolds numbers are solved. The doubly periodic shear layer flows are also used to assess the efficiency of the algorithm.

© 2014 Elsevier Inc. All rights reserved.

1. Introduction

The model problems we consider are the incompressible Navier–Stokes equations on a bounded domain $\Omega \subset \mathbb{R}^2$ which are given by

$$\begin{cases} \partial_t \mathbf{u} + (\mathbf{u} \cdot \nabla) \mathbf{u} + \nabla p = \nu \Delta \mathbf{u}, \\ \nabla \cdot \mathbf{u} = \mathbf{0}, \\ \mathbf{u}|_{\Gamma} = \mathbf{g} := (g_1, g_2)^T, \end{cases}$$
(1)

where $\mathbf{u} = (u, v)^T$, p, v and Γ denote the velocity field, pressure, kinematic viscosity and the boundary of Ω , respectively. Here, the boundary conditions for velocity fields are considered slip or periodic cases.

* Corresponding author.

http://dx.doi.org/10.1016/j.jcp.2014.11.040



 $^{^{*}}$ This work was supported by basic science research program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (grant number 2011-0029013). The first author (Piao) was supported by the 2014 Hannam University Research Fund. Also, the corresponding author (Kim) was partially supported by basic science research program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (grant number 2011-0009825).

E-mail addresses: piaoxf76@hanmail.net (X. Piao), syboo@knu.ac.kr (S. Bu), jiya525@naver.com (S. Bak), kimps@knu.ac.kr (P. Kim).

 $^{0021\}mathchar`-9991/\mathchar` 2014$ Elsevier Inc. All rights reserved.

Characteristic-based methods are popular among numerical techniques for solving time-dependent advection-dominated partial differential equations (see [10]). One such method, the backward semi-Lagrangian (BSL) method was designed by Robert [28] to solve meteorological equations in the beginning of the eighties. BSL methods have various significant advantages. (i) They allow a large time step size without damaging the accuracy of the solution. (ii) Unlike pure Lagrangian methods, they do not suffer from mesh-deformation, so that no remeshing is needed. If free boundaries are present, a new mesh should be used at each time step. (iii) They yield algebraic symmetric systems of equations to be solved. Because of these advantages, BSL methods have been extensively used in the numerical simulation of models for fluid dynamics (for examples, see [2–4,9,25,29,34,35] and the references therein).

Most existing BSL methods focus on the development of the interpolation scheme of the solution and spatial discretization schemes. Despite the importance of accuracy in numerical schemes for finding departure points of fluid particles arriving at an Eulerian grid point, it has been received little attention. One reason for this is that the characteristic curves of the particles are described by highly nonlinear ordinary differential equations (ODE) which must be coupled with the solution of the original problems. This is a so-called self-consistency problem and the reason why a high-order time integration scheme is practically hard to implement. It is a known drawback of the BSL methods.

Methods for solving the characteristic equation have a particularly sensitive effect on the accuracy of BSL methods. Traditionally, two main strategies have been proposed and implemented to solve highly nonlinear initial value problems (IVPs) and to find the departure points of the fluid particles. One is an implicit approach requiring iteration [1,32,34]. The other is a substepping method, which is an explicit type [34]. These methods have both second-order convergence accuracy, but it is well known that for a stiff problem, the implicit method achieves a slightly more accurate result compared to the explicit method. Furthermore, when the Reynolds number is large, the implicit method gets a more accurate solution than the explicit method. In addition, the explicit method may work ineffectively in some special cases. The conventional second-order backward integration schemes require an iteration process such as fixed point or Newton iteration when the velocity changes with time (see [23]). At each time and for every spatial point, this iteration process requires the interpolation of solutions which need considerable computational costs. Sometimes, it is prone to accumulate errors during a long-time simulation.

The primary aim of this paper is to develop a BSL method that retains the advantages of conventional second order BSL methods but that does not require the ineffective iteration steps for solving the self-consistent nonlinear problem of the characteristic equations. To accomplish these, we discretize the time derivative of the Stokes equations with the second order backward differentiation formula (BDF2) and apply a projection method in order to split the steady state Stokes equation into velocity and pressure. Secondly, fourth-order finite differences for partial derivatives are used to discretize the Poisson equation for the pressure and the Helmholtz equation for the velocity. Also, the finite discrete linear systems for both the Poisson equation and the Helmholtz equation are solved with a matrix-diagonalization technique. Finally, we apply the error correction techniques, which originated in our recent articles (see [20-22,27]), to solve the highly nonlinear initial value problem of finding the departure points of fluid particles. The error correction method (ECM) is based on the Euler's polygon on each time integration step. To maintain the advantages of the ECM, we suggest a modified Euler's polygon and apply the A-stable midpoint rule for the time integration of the initial value problems. As an interpolation scheme for the solution, the Hermite cubic interpolation technique discussed by Kim et al. [19] is used. The resulting algorithm turns out to be completely iteration free. In particular, it exhibits a good behavior of the total energy conservation compared to existing methods. To assess the effectiveness of the method, two-dimensional lid-driven cavity problems with large different Reynolds numbers are solved. The doubly periodic shear layer flows are also used to assess the efficiency of the proposed method. Throughout these numerical tests, it is shown that the proposed method is quite efficient compared to existing methods.

This paper is organized as follows. In Section 2, we include a brief review of the backward semi-Lagrangian method together with the projection method to deal with steady state Stokes equations. Also, we review the fourth-order finite difference schemes for approximating partial derivatives and the Hermite cubic interpolation theory required for the spatial discretization. Section 3 describes the error correction scheme for solving the self-consistent nonlinear initial value problem for the characteristic curves. In Section 4, we review the matrix-diagonalization technique for solving the finite systems obtained from the discretization based on the finite difference method of the Poisson equation and the Helmholtz equation. Several test problems are performed in Section 5 to exhibit the accuracy and superiority of the proposed method. Finally, in Section 6, a summary for the method, and some discussion of further work are given.

2. Preliminary

The aim of this section is to review a backward semi-Lagrangian scheme for solving the model problem (1) based on the characteristic curve and also to introduce a projection method for a steady Stokes' equation [14,16,24,34]. The splitting scheme is referred to as the rotational form of the velocity-correction scheme in [16] and is also used with semi-Lagrangian schemes in [14,34]. Let $\pi(s, \mathbf{x}; t) := (\pi_1(t), \pi_2(t))^T$ be the characteristic curves satisfying the initial value problem given by

$$\frac{d\boldsymbol{\pi}(s, \mathbf{x}; t)}{dt} = \mathbf{u}(t, \boldsymbol{\pi}(s, \mathbf{x}; t)), \quad t < s,$$

$$\boldsymbol{\pi}(s, \mathbf{x}; s) = \mathbf{x}$$
(2)

where **u** is the solution of the model problem (1) and $\mathbf{x} = (x_1, x_2)^T$ denotes the arbitrary spatial variable. Then, by combining (2) with (1), one may check that the total derivative of $\mathbf{u}(t, \boldsymbol{\pi}(s, \mathbf{x}; t))$ along the characteristic curve $\boldsymbol{\pi}(s, \mathbf{x}; t)$ satisfies the following Stokes equation:

$$\frac{d}{dt}\mathbf{u}(t,\boldsymbol{\pi}(s,\mathbf{x};t)) + \nabla p(t,\boldsymbol{\pi}(s,\mathbf{x};t)) = \nu \Delta \mathbf{u}(t,\boldsymbol{\pi}(s,\mathbf{x};t)).$$
(3)

For further discussion, we introduce a uniform discretization of the temporal variable as follows:

$$t_m = mh, \quad 0 \le m \le M, \tag{4}$$

where *M* is the number of time steps and *h* is a time step size. Also, we introduce notations $\mathbf{u}^k(\mathbf{x}) := \mathbf{u}(t_k, \mathbf{x})$ and $p^k(\mathbf{x}) := p(t_k, \mathbf{x})$. After evaluating (3) at time $t = t_{m+1}$ by setting $s = t_{m+1}$ and then applying the second order backward differentiation formula to approximate the total time derivative, one gets the following steady state Stokes equation with an asymptotic term $\mathcal{O}(h^2)$.

$$\frac{3}{2h}\mathbf{u}^{m+1}(\mathbf{x}) - \nu \Delta \mathbf{u}^{m+1}(\mathbf{x}) + \nabla p^{m+1}(\mathbf{x}) = \mathbf{f}^{m+1}(\mathbf{x}) + \mathcal{O}(h^2),$$
(5)

where

$$\mathbf{f}^{m+1}(\mathbf{x}) = \frac{4\mathbf{u}^m(\boldsymbol{\pi}(t_{m+1}, \mathbf{x}; t_m)) - \mathbf{u}^{m-1}(\boldsymbol{\pi}(t_{m+1}, \mathbf{x}; t_{m-1}))}{2h}.$$
(6)

As a strategy for treating the Stokes equation (5), we will split it into pressure and velocity using a projection scheme as follows. By taking the divergence of both sides of the Stokes equation (5) and using the divergence free condition $\nabla \cdot \mathbf{u}(t, \mathbf{x}) = 0$ in (1), we get a Poisson equation for the pressure given by

$$\Delta p^{m+1}(\mathbf{x}) = \nabla \cdot \mathbf{f}^{m+1}(\mathbf{x}) + \mathcal{O}(h^2).$$
(7)

To get a suitable boundary condition for the pressure $p^{m+1}(\mathbf{x})$, we first multiply an outer normal vector **n** into both sides of Eq. (5) and then take a limit of the spatial variable **x** to the boundary. Then, by using the identity $\Delta \mathbf{u} = \nabla(\nabla \cdot \mathbf{u}) - \nabla \times \nabla \times \mathbf{u}$ and the boundary condition of (1), we find

$$\frac{\partial}{\partial \mathbf{n}} p^{m+1}(\mathbf{x}) \bigg|_{\Gamma} = \mathbf{n} \cdot \left(\mathbf{f}^{m+1}(\mathbf{x}) - \frac{3}{2h} \mathbf{u}^{m+1}(\mathbf{x}) - \nu \nabla \times \nabla \times \mathbf{u}^{m+1}(\mathbf{x}) \right) \bigg|_{\Gamma} + \mathcal{O}(h^2)$$
$$= \mathbf{n} \cdot \left(\mathbf{f}^{m+1}(\mathbf{x}) - \frac{3}{2h} \mathbf{g}^{m+1}(\mathbf{x}) - \nu \nabla \times \nabla \times \mathbf{u}^{m+1}(\mathbf{x}) \right) \bigg|_{\Gamma} + \mathcal{O}(h^2). \tag{8}$$

Notice that the last term in the parentheses of the right hand side of (8) is unknown and hence a suitable approximation is required. Substituting the expansion $\mathbf{u}^{m+1}(\mathbf{x}) = 2\mathbf{u}^m(\mathbf{x}) - \mathbf{u}^{m-1}(\mathbf{x}) + \mathcal{O}(h^2)$ into (8), one gets a known Neumann boundary condition for the pressure with an asymptotic term $\mathcal{O}(h^2)$ given by

$$\frac{\partial}{\partial \mathbf{n}} p^{m+1}(\mathbf{x}) \bigg|_{\Gamma} = \mathbf{n} \cdot \left(\mathbf{f}^{m+1}(\mathbf{x}) - \frac{3}{2h} \mathbf{g}^{m+1}(\mathbf{x}) - \nu \nabla \times \nabla \times \left(2\mathbf{u}^m(\mathbf{x}) - \mathbf{u}^{m-1}(\mathbf{x}) \right) \right) \bigg|_{\Gamma} + \mathcal{O}(h^2).$$
(9)

Finally, combining Eqs. (7) and (9) leads to a Neumann boundary value problem for the pressure at time t_{m+1} with an asymptotic term $\mathcal{O}(h^2)$ described by

$$\begin{cases} \Delta p^{m+1}(\mathbf{x}) = \nabla \cdot \mathbf{f}^{m+1}(\mathbf{x}) + \mathcal{O}(h^2), \\ \frac{\partial}{\partial \mathbf{n}} p^{m+1}(\mathbf{x}) \Big|_{\Gamma} = \mathbf{n} \cdot \left(\mathbf{f}^{m+1}(\mathbf{x}) - \frac{3}{2h} \mathbf{g}^{m+1}(\mathbf{x}) - \nu \nabla \times \nabla \times \left(2\mathbf{u}^m(\mathbf{x}) - \mathbf{u}^{m-1}(\mathbf{x}) \right) \right) \Big|_{\Gamma} + \mathcal{O}(h^2). \end{cases}$$
(10)

Notice that given the velocities at the times t_m and t_{m-1} , one can get an approximate value for the pressure p^{m+1} at time t_{m+1} by truncating the asymptotic term, and then solving the boundary value problem (10). Further, after solving (10) and then using Eq. (5) together with the Dirichlet boundary condition for the velocity given in (1), the velocity field \mathbf{u}^{m+1} at time t_{m+1} can be obtained by solving the Helmholtz equation described by

$$\begin{cases} \frac{3}{2h} \mathbf{u}^{m+1}(\mathbf{x}) - \nu \Delta \mathbf{u}^{m+1}(\mathbf{x}) = \mathbf{f}^{m+1}(\mathbf{x}) - \nabla p^{m+1}(\mathbf{x}) + \mathcal{O}(h^2), \\ \mathbf{u}^{m+1}(\mathbf{x}) \Big|_{\Gamma} = \mathbf{g}^{m+1}(\mathbf{x}). \end{cases}$$
(11)

Notice that the original problem (1) is split into one nonlinear initial value problem (2) and two linear boundary value problems (10) and (11). This is a backward semi-Lagrangian method (BSLM). Note that both boundary value problems (10) and (11) require the function values of \mathbf{f}^{n+1} defined by (6). To find this, one has to solve the initial value problem (2) to get departure points $\pi(t_{m+1}, \mathbf{x}; t_m)$ and $\pi(t_{m+1}, \mathbf{x}; t_{m-1})$ reaching the same point $\mathbf{x} = \pi(t_{m+1}, \mathbf{x}; t_{m+1})$. In fact, the slope function of the characteristic curve $\pi(s, \mathbf{x}; t)$ is the unknown velocity field \mathbf{u} on time interval $(t_m, t_{m+1}]$. Thus, one may

conclude that the solutions of the problems (2), (10) and (11) are self-consistent and hence the problem we have to solve has a highly nonlinear self consistency, which is comparable with the nonlinearity of the original problem (1). Also, from the boundary value problems (10) and (11), one can get approximate values for the velocity and the pressure only at the grid points. Usually, however, the departure points $\pi(t_{m+1}, \mathbf{x}; t_m)$ and $\pi(t_{m+1}, \mathbf{x}; t_{m-1})$ do not coincide with grid points. Hence, a suitable interpolation scheme for the solutions is needed.

In the following subsequent sections, we will develop numerical schemes to solve these self-consistent problems. In particular, we focus on the development of an iteration free scheme for solving the initial value problem (2). To solve the boundary value problems (10) and (11), we will use a finite difference scheme. Notice that the numerical solutions for the velocity field and the pressure are known only on the grid points and hence an interpolation scheme has to be adopted to solve the boundary value problems. Before closing this section, we introduce a fourth-order finite difference scheme for approximating the first and second order partial derivatives and introduce the Hermite cubic interpolation technique. For simplicity, we assume that the computational domain Ω is a rectangle given by $\Omega := [x_{1,\min}, x_{1,\max}] \times [x_{2,\min}, x_{2,\max}] \subset \mathbb{R}^2$ and it is uniformly divided as follows:

$$x_{k,\min} = x_{k,0} < x_{k,1} < \dots < x_{k,N_k} = x_{k,\max}, \quad x_{k,i} = x_{k,0} + i\Delta x_k, \tag{12}$$

where $\Delta x_k := (x_{k,\max} - x_{k,\min})/N_k$ (k = 1, 2) are given uniform spatial mesh sizes in x_k directions. For a function f, we let $f_{i,j} := f(\mathbf{x}_{i,j}), \mathbf{x}_{i,j} := (x_{1,i}, x_{2,j})^T$. The first and second derivatives $\frac{\partial}{\partial x_1} f_{i,j}$ and $\frac{\partial^2}{\partial x_1^2} f_{i,j}$, which are required in our calculation, are approximated by the fourth-order finite difference schemes (for example, see [13]) defined by

$$\frac{\partial}{\partial x_1} f_{i,j} = \frac{f_{i-2,j} - 8f_{i-1,j} + 8f_{i+1,j} - f_{i+2,j}}{12\Delta x_1} + \mathcal{O}(\Delta x_1^4), \quad i = 2, 3, \cdots, N_1 - 2,$$

$$\frac{\partial}{\partial x_1} f_{i,j} = \pm \frac{-22f_{i,j} + 36f_{i\pm 1,j} - 18f_{i\pm 2,j} + 4f_{i\pm 3,j}}{12\Delta x_1} + \mathcal{O}(\Delta x_1^3), \quad i = 0, N_1,$$

$$\frac{\partial}{\partial x_1} f_{i,j} = \frac{6f_{i+1,j} - 6f_{i-1,j}}{12\Delta x_1} + \mathcal{O}(\Delta x_1^2), \quad i = 1, N_1 - 1$$
(13)

and

$$\frac{\partial^2}{\partial x_1^2} f_{i,j} = \frac{-f_{i-2,j} + 16f_{i-1,j} - 30f_{i,j} + 16f_{i+1,j} - f_{i+2,j}}{12\Delta x_1^2} + \mathcal{O}(\Delta x_1^4), \quad i = 2, 3, \cdots, N_1 - 2,$$

$$\frac{\partial^2}{\partial x_1^2} f_{i,j} = \frac{45f_{i,j} - 154f_{i\pm 1,j} + 214f_{i\pm 2,j} - 156f_{i\pm 3,j} + 61f_{i\pm 4,j} - 10f_{i\pm 5,j}}{12\Delta x_1^2} + \mathcal{O}(\Delta x_1^4), \quad i = 0, N_1,$$

$$\frac{\partial^2}{\partial x_1^2} f_{i,j} = \frac{10f_{i\mp 1,j} - 15f_{i,j} - 4f_{i\pm 1,j} + 14f_{i\pm 2,j} - 6f_{i\pm 3,j} + f_{i\pm 4,j}}{12\Delta x_1^2} + \mathcal{O}(\Delta x_1^4), \quad i = 1, N_1 - 1,$$
(14)

respectively, where the notations $f_{i\pm k}$ and $f_{i\mp 1}$ are chosen so that the corresponding points $x_{i\pm k}$ and $x_{i\mp 1}$ are included in the computational domain Ω . Similarly, the approximations for the derivatives $\frac{\partial}{\partial x_2} f_{i,j}$ and $\frac{\partial^2}{\partial x_2^2} f_{i,j}$ are defined. With these approximations, we introduce the Hermite cubic interpolation $\mathcal{H}F$ [19], which is defined as follows: for $\mathbf{x} := (x_1, x_2)^T \in [x_{1,i}, x_{1,i+1}] \times [x_{2,j}, x_{2,j+1}]$,

$$\mathcal{H}F(\mathbf{x}) := \mathcal{H}_{i,j}f(\mathbf{x}) = \sum_{0 \le k+l \le 3} C_{k,l}(\mathbf{x} - \mathbf{x}_{i,j})^{k,l} + C_{3,1}(\mathbf{x} - \mathbf{x}_{i,j})^{3,1} + C_{1,3}(\mathbf{x} - \mathbf{x}_{i,j})^{1,3},$$
(15)

where $(\mathbf{x} - \mathbf{x}_{i,j})^{k,l} = (x_1 - x_{1,i})^k (x_2 - x_{2,j})^l$ and the coefficients $C_{k,l}$ are uniquely defined as follows.

$$C_{0,0} = f(\mathbf{x}_{i,j}), \qquad C_{1,0} = \frac{\partial}{\partial x_1} f(\mathbf{x}_{i,j}), \qquad C_{0,1} = \frac{\partial}{\partial x_2} f(\mathbf{x}_{i,j}), \\ C_{2,0} = 3(f(\mathbf{x}_{i+1,j}) - f(\mathbf{x}_{i,j})) - \frac{\partial}{\partial x_1} f(\mathbf{x}_{i+1,j}) - 2\frac{\partial}{\partial x_1} f(\mathbf{x}_{i,j}), \\ C_{0,2} = 3(f(\mathbf{x}_{i,j+1}) - f(\mathbf{x}_{i,j})) - \frac{\partial}{\partial x_2} f(\mathbf{x}_{i,j+1}) - 2\frac{\partial}{\partial x_2} f(\mathbf{x}_{i,j}), \\ C_{3,0} = -2(f(\mathbf{x}_{i+1,j}) - f(\mathbf{x}_{i,j})) + \frac{\partial}{\partial x_1} f(\mathbf{x}_{i+1,j}) + \frac{\partial}{\partial x_1} f(\mathbf{x}_{i,j}), \\ C_{0,3} = -2(f(\mathbf{x}_{i,j+1}) - f(\mathbf{x}_{i,j})) + \frac{\partial}{\partial x_2} f(\mathbf{x}_{i,j+1}) + \frac{\partial}{\partial x_2} f(\mathbf{x}_{i,j}), \\ C_{2,1} = 3f(\mathbf{x}_{i+1,j+1}) - 2\frac{\partial}{\partial x_1} f(\mathbf{x}_{i,j+1}) - \frac{\partial}{\partial x_1} f(\mathbf{x}_{i+1,j+1}) - 3\sum_{k=0}^{3} C_{0,k} - C_{2,0}, \end{cases}$$

$$C_{3,1} = -2f(\mathbf{x}_{i+1,j+1}) + \frac{\partial}{\partial x_1} f(\mathbf{x}_{i,j+1}) + \frac{\partial}{\partial x_1} f(\mathbf{x}_{i+1,j+1}) + 2\sum_{k=0}^{3} C_{0,k} - C_{3,0},$$

$$C_{1,2} = 3f(\mathbf{x}_{i+1,j+1}) - 2\frac{\partial}{\partial x_2} f(\mathbf{x}_{i+1,j}) - \frac{\partial}{\partial x_2} f(\mathbf{x}_{i+1,j+1}) - 3\sum_{k=0}^{3} C_{k,0} - C_{0,2},$$

$$C_{1,3} = -2f(\mathbf{x}_{i+1,j+1}) + \frac{\partial}{\partial x_2} f(\mathbf{x}_{i+1,j}) + \frac{\partial}{\partial x_2} f(\mathbf{x}_{i+1,j+1}) + 2\sum_{k=0}^{3} C_{k,0} - C_{0,3},$$

$$C_{1,1} = \frac{\partial}{\partial x_1} f(\mathbf{x}_{i,j+1}) - C_{1,3} - C_{1,2} - C_{1,0},$$
(16)

where all the partial derivatives are approximated with the mentioned finite difference scheme. For vector valued functions, the hermite interpolation scheme can be extended as follows: for $\mathbf{F} := (F_1, F_2)^T$, we have

$$\mathcal{H}\mathbf{F}(\mathbf{x}) = \left(\mathcal{H}F_1(\mathbf{x}), \mathcal{H}F_2(\mathbf{x})\right)^T.$$

For a simple expression of the first and second partial derivatives (13) and (14), we introduce the differentiation matrices D_{1,N_k} and D_{2,N_k} of size $(N_k + 1) \times (N_k + 1)$ as follows.

$$D_{1,N_k} = \frac{1}{12\Delta x_k} \begin{bmatrix} -22 & 36 & -18 & 4 & & \\ -6 & 0 & 6 & 0 & & \\ 1 & -8 & 0 & 8 & -1 & & \\ & \ddots & \ddots & \ddots & \ddots & \ddots & \\ & & 1 & -8 & 0 & 8 & -1 \\ & & & 0 & -6 & 0 & 6 \\ & & & & -4 & 18 & -36 & 22 \end{bmatrix}$$
(17)

and

$$D_{2,N_k} = \frac{1}{12\Delta x_k^2} \begin{bmatrix} 45 & -154 & 214 & -156 & 61 & -10 \\ 10 & -15 & -4 & 14 & -6 & 1 \\ -1 & 16 & -30 & 16 & -1 \\ & \ddots & \ddots & \ddots & \ddots & \ddots \\ & & -1 & 16 & -30 & 16 & -1 \\ & 1 & -6 & 14 & -4 & -15 & 10 \\ & & -10 & 61 & -156 & 214 & -154 & 45 \end{bmatrix}.$$
(18)

Then, the first and second derivatives $\frac{\partial}{\partial x_k} f_{i,j}$, being $2 \le i \le N_1 - 2$ if k = 1, and $2 \le j \le N_2 - 2$ if k = 2 and $\frac{\partial^2}{\partial x_k^2} f_{i,j}$ are respectively expressed with the described matrices as follows.

$$\frac{\partial}{\partial x_1} f_{i,j} = (D_{1,N_1} F)_{i,j} + \mathcal{O}(\Delta x_1^4), \qquad \frac{\partial}{\partial x_2} f_{i,j} = (FD_{1,N_2}^T)_{i,j} + \mathcal{O}(\Delta x_2^4),$$

$$\frac{\partial^2}{\partial x_1^2} f_{i,j} = (D_{2,N_1} F)_{i,j} + \mathcal{O}(\Delta x_1^4), \qquad \frac{\partial^2}{\partial x_2^2} f_{i,j} = (FD_{2,N_2}^T)_{i,j} + \mathcal{O}(\Delta x_2^4),$$
(19)

where $F = (f_{i,j})_{(N_1+1)\times(N_2+1)}$. In a similar way, the last two equations of (13) can be simplified with the above differentiation matrices.

3. Error correction method for solving (2)

This section aims to develop an iteration free numerical scheme for finding the approximate values of π ($t_{m+1}, \mathbf{x}_{i,j}; t_{m-k}$), k = 0, 1, where $\mathbf{x}_{i,j}$ denotes an arbitrary grid point. We will follow the technique of the error correction method recently developed by the authors (for examples, see [20,21,27]). In particular, we will derive an integration scheme with the rate of convergence order 2. Assume that the velocity vector $\mathbf{u}(t, \mathbf{x})$ for time $t_{m-k}, k = 0, 1$, is already calculated at all grid points, whose approximations are denoted by

$$\mathbf{U}^{m-k} := \{ U^{m-k}, V^{m-k} \}, \quad (U^{m-k})_{i,j} = u^{m-k}_{i,j}, \quad (V^{m-k})_{i,j} = v^{m-k}_{i,j},$$

where $u_{i,j}^{m-k}$ and $v_{i,j}^{m-k}$ are approximated solutions at time t_{m-k} and grid points $\mathbf{x}_{i,j}$ which are obtained either from the initial conditions or by using the scheme described later. For simplicity, let $\boldsymbol{\pi}_{i,j}(t) := \boldsymbol{\pi}(t_{m+1}, \mathbf{x}_{i,j}; t)$ be the solution of the self-consistent nonlinear initial value problem given by

$$\begin{cases} \frac{d\pi_{i,j}(t)}{dt} = \mathbf{u}(t, \pi_{i,j}(t)), & t \in [t_{m-1}, t_{m+1}), \\ \pi_{i,j}(t_{m+1}) = \mathbf{x}_{i,j}, \end{cases}$$
(20)

where **u** is the solution of the model problem (1). Notice that the Taylor's expansion of $\pi_{i,j}(t)$ about t_{m-1} together with the initial condition $\pi_{i,j}(t_{m+1}) = \mathbf{x}_{i,j}$ can be written as

$$\pi_{i,j}(t) = \pi_{i,j}(t_{m-1}) + (t - t_{m-1})\mathbf{u}(t_{m-1}, \pi_{i,j}(t_{m-1})) + \frac{(t - t_{m-1})^2}{4h^2} (\mathbf{x}_{i,j} - \pi_{i,j}(t_{m-1}) - 2h\mathbf{u}(t_{m-1}, \pi_{i,j}(t_{m-1}))) + \mathcal{O}(h^3), \quad t \in [t_{m-1}, t_{m+1}].$$
(21)

Thus, a relation between $\pi_{i,j}(t_m)$ and $\pi_{i,j}(t_{m-1})$ using the interpolation scheme (15) can be obtained by evaluating equation (21) at time t_m as follows.

$$\boldsymbol{\pi}_{i,j}(t_m) \approx \frac{1}{4} \big(\mathbf{x}_{i,j} + 3\boldsymbol{\pi}_{i,j}(t_{m-1}) + 2h\mathcal{H}\mathbf{U}^{m-1} \big(\boldsymbol{\pi}_{i,j}(t_{m-1}) \big) \big).$$
(22)

Eq. (22) gives a suitable approximation of $\pi_{i,j}(t_m)$, provided an approximation of $\pi_{i,j}(t_{m-1})$. Hence, we will focus on finding an approximation of $\pi_{i,j}(t_{m-1})$. First, we introduce existing explicit and implicit second-order schemes for comparison. As stated in [24,34], an approximation $\pi_{i,j}^{m-1}$ of $\pi_{i,j}(t_{m-1})$ can be obtained using either the second order explicit mid-point rule

$$\widehat{\boldsymbol{\pi}} = \mathbf{x}_{i,j} - h \mathbf{U}_{i,j}^{m}, \qquad \boldsymbol{\pi}_{i,j}^{m-1} = \mathbf{x}_{i,j} - 2h \mathcal{H} \mathbf{U}^{m}(\widehat{\boldsymbol{\pi}}),$$
(23)

or the second order implicit mid-point rule

$$\boldsymbol{\pi}_{i,j}^{m-1} = \mathbf{x}_{i,j} - 2\boldsymbol{\alpha}, \quad \boldsymbol{\alpha} \approx h \mathcal{H} \mathbf{U}^m(\mathbf{x}_{i,j} - \boldsymbol{\alpha}).$$
(24)

Note that the solution α of the last equation in (24) is usually obtained by iterative methods such as the fixed point iteration or the Newton iteration method.

Our development of a new iteration free second-order scheme for (20) begins with a discussion using Euler's polygon in the error correction strategy [20,21]. Since $\pi_{i,j}(t)$ is the position of the trajectory of the fluid particles over the solution surface, the characteristic curves $\pi_{i,j}(t) = \pi(t_{m+1}, \mathbf{x}_{i,j}; t)$ can be assumed to be sufficiently smooth with respect to the time variable *t*. Thus, for the characteristic curve $\pi_{i,j}(t)$ satisfying (20), taking its Taylor's expansion about t_{m+1} , and the Taylor's expansion of $\mathbf{u}(t_{m+1}, \pi_{i,j}(t_{m+1}))$ about t_m , we get the following expansion

$$\pi_{i,j}(t) = \mathbf{x}_{i,j} + (t - t_{m+1})\mathbf{u}(t_{m+1}, \pi_{i,j}(t_{m+1})) + \mathcal{O}(h^2)$$

= $\mathbf{x}_{i,j} + (t - t_{m+1})\mathbf{u}(t_m, \mathbf{x}_{i,j}) + \mathcal{O}(h^2)$
 $\approx \mathbf{x}_{i,j} + (t - t_{m+1})\mathbf{U}_{i,j}^m, \quad t \in [t_{m-1}, t_{m+1}].$ (25)

Following (25), we introduce an Euler's polygon $\mathbf{y}_{i,j}(t)$ defined by

$$\mathbf{y}_{i,j}(t) := \mathbf{x}_{i,j} + (t - t_{m+1})\mathbf{U}_{i,j}^{m}, \quad t \in [t_{m-1}, t_{m+1}]$$
(26)

and define

. .

$$\boldsymbol{\psi}_{i,j}(t) := \boldsymbol{\pi}_{i,j}(t) - \mathbf{y}_{i,j}(t), \quad t \in [t_{m-1}, t_{m+1}].$$
(27)

By differentiating both sides of (27) and applying the Taylor's expansion into the result together with (20), one gets an asymptotically first-order ODE given by

$$\Psi_{i,j}^{\prime}(t) = \mathbf{u}(t, \Psi_{i,j}(t) + \mathbf{y}_{i,j}(t)) - \mathbf{y}_{i,j}^{\prime}(t)$$

= $\mathbf{u}(t, \mathbf{y}_{i,j}(t)) + \mathbf{u}_{\mathbf{x}}(t, \mathbf{y}_{i,j}(t)) \Psi_{i,j}(t) - \mathbf{U}_{i,j}^{m} + \mathcal{O}(\Psi_{i,j}(t)^{2})$
 $\approx \mathbf{U}_{\mathbf{x}}(t_{m}, \mathbf{y}_{i,j}(t_{m})) \Psi_{i,j}(t) + \mathbf{u}(t, \mathbf{y}_{i,j}(t)) - \mathbf{U}_{i,j}^{m}, \quad t \in [t_{m-1}, t_{m+1}],$ (28)

where the Jacobian matrices $\mathbf{u}_{\mathbf{x}}(t, \mathbf{y}_{i,i}(t))$ and $\mathbf{U}_{\mathbf{x}}(t_m, \mathbf{y}_{i,i}(t_m))$ are given by

$$\mathbf{u}_{\mathbf{X}}(t, \mathbf{y}_{i,j}(t)) = \begin{bmatrix} \frac{\partial}{\partial x_1} u(t, \mathbf{y}_{i,j}(t)) & \frac{\partial}{\partial x_2} u(t, \mathbf{y}_{i,j}(t)) \\ \frac{\partial}{\partial x_1} v(t, \mathbf{y}_{i,j}(t)) & \frac{\partial}{\partial x_2} v(t, \mathbf{y}_{i,j}(t)) \end{bmatrix}, \quad t \in [t_{m-1}, t_{m+1}]$$

and

$$\mathbf{U}_{\mathbf{x}}(t_m, \mathbf{y}_{i,j}(t_m)) = \begin{bmatrix} \frac{\partial}{\partial x_1} \mathcal{H} U^m(\mathbf{y}_{i,j}(t_m)) & \frac{\partial}{\partial x_2} \mathcal{H} U^m(\mathbf{y}_{i,j}(t_m)) \\ \frac{\partial}{\partial x_1} \mathcal{H} V^m(\mathbf{y}_{i,j}(t_m)) & \frac{\partial}{\partial x_2} \mathcal{H} V^m(\mathbf{y}_{i,j}(t_m)) \end{bmatrix},$$
(29)

respectively. Note that the first equation of (28) is known as the deferred differential equation for (20) and one can use it together with (27) to find the approximate value $\pi_{i,j}^{m-1}$. However, there may be some computational difficulties in solving the equation due to the highly nonlinearity of the self-consistent constraint for the velocity field **u**. On the other hand, the last equation of (28) is of linear, while being self-consistent on the right hand side. Since the solution of Eq. (28) corresponds to the error for the Euler's polygon, we would like to say the proposed method with the so-called error correction method.

To solve Eq. (28), we apply the mid-point integration method, which is known as an A-stable implicit method. From (26) and (27), it can be seen that $\psi_{i,j}(t_{m+1}) = \mathbf{0}$. Hence, integrating both sides of (28) over $[t_{m-1}, t_{m+1}]$ using the mid-point integration rule, one gets an asymptotic formula given by

$$-\boldsymbol{\psi}_{i,j}(t_{m-1}) \approx 2h \big(\mathbf{U}_{\mathbf{x}}(t_m, \mathbf{y}_{i,j}(t_m)) \boldsymbol{\psi}_{i,j}(t_m) + \mathcal{H} \mathbf{U}^m \big(\mathbf{y}_{i,j}(t_m) \big) - \mathbf{U}_{i,j}^m \big).$$
(30)

Since $\psi_{i,j}(t_m) = \frac{1}{2}(\psi_{i,j}(t_{m+1}) + \psi_{i,j}(t_{m-1})) + O(h^2)$ which can be proved by the Taylor's expansion theorem, Eq. (30) can be rewritten as

$$\left(\mathcal{I} + h\mathbf{U}_{\mathbf{x}}(t_m, \mathbf{y}_{i,j}(t_m))\right) \boldsymbol{\psi}_{i,j}(t_{m-1}) \approx 2h\left(\mathbf{U}_{i,j}^m - \mathcal{H}\mathbf{U}^m(\mathbf{y}_{i,j}(t_m))\right),\tag{31}$$

where \mathcal{I} is an identity matrix. Finally, combining the solution of (31) with (26) and (27), one gets an approximate formula $\pi_{i,i}^{m-1}$ defined by

$$\boldsymbol{\pi}_{i,j}^{m-1} := \mathbf{y}_{i,j}(t_{m-1}) + \boldsymbol{\psi}_{i,j}^{m-1}, \tag{32}$$

where $\psi_{i,j}^{m-1}$ is the solution of Eq. (31). From the approximation $\pi_{i,j}^{m-1}$ defined by (32), identity (22) gives an approximation of $\pi_{i,j}(t_m)$.

4. Finite difference methods for Poisson equation and Helmholtz equation

In this section, we present implementation details for solving the Poisson equation (10) and the Helmholtz equation (11) based on the finite difference scheme discussed in Section 2. Using the differentiation matrices defined in (17) and (18), the spatial discretization formula for both Eqs. (10) and (11) can be obtained and it can be solved with matrix-diagonalization techniques ([18,30]). For reader's convenience, we review the matrix-diagonalization solvers of (10) and (11).

4.1. Matrix-diagonalization method to Poisson equation (10)

Assume the approximations $\pi_{i,j}^{m-1}$ and $\pi_{i,j}^m$ are obtained using the scheme developed in the previous section. Then, the Poisson equation (10) using the differentiation matrices (17) and (18) is discretized as follows.

$$D_{2,N_1}P^{m+1} + P^{m+1}D_{2,N_2}^T \approx D_{1,N_1}F_1^{m+1} + F_2^{m+1}D_{1,N_2}^T,$$
(33)

where P^{m+1} is the matrix of size $(N_1 + 1) \times (N_2 + 1)$ for the pressure and F_k^{m+1} (k = 1, 2) are matrices of the same size, whose elements are defined by

$$(F_1^{m+1})_{i,j} := \frac{1}{2h} (4\mathcal{H}U^m(\boldsymbol{\pi}_{i,j}^m) - \mathcal{H}U^{m-1}(\boldsymbol{\pi}_{i,j}^{m-1})), \qquad (F_2^{m+1})_{i,j} := \frac{1}{2h} (4\mathcal{H}V^m(\boldsymbol{\pi}_{i,j}^m) - \mathcal{H}V^{m-1}(\boldsymbol{\pi}_{i,j}^{m-1})). \tag{34}$$

The x_k -component of the right-hand side of the boundary equation in (10) can be discretized by

$$BC_{1} \approx -\nu \left(D_{1,N_{1}} \left(2V^{m} - V^{m-1} \right) - \left(2U^{m} - U^{m-1} \right) D_{1,N_{2}}^{T} \right) D_{1,N_{2}}^{T} + F_{1}^{m+1} - \frac{3}{2h} G_{1}^{m+1},$$

$$BC_{2} \approx -\nu D_{1,N_{1}} \left(D_{1,N_{1}} \left(2V^{m} - V^{m-1} \right) - \left(2U^{m} - U^{m-1} \right) D_{1,N_{2}}^{T} \right) + F_{2}^{m+1} - \frac{3}{2h} G_{2}^{m+1}.$$
(35)

Here, U^k and V^k , k = m, m - 1, are the approximated matrices for the velocity vector $\mathbf{u}(t_k, \mathbf{x})$ defined by

$$U^{k} := (u_{i,j}^{k})_{(N_{1}+1)\times(N_{2}+1)}, \qquad V^{k} := (v_{i,j}^{k})_{(N_{1}+1)\times(N_{2}+1)}$$

and the matrix G_k^{m+1} is defined by

$$G_k^{m+1} := \left(g_k(t_{m+1}, \mathbf{x}_{i,j})\right)_{(N_1+1)\times(N_2+1)}$$

where g_k is the *k*th component of **g**. Then, the boundary conditions of (10) along $x_k = x_{k,\min}$ and $x_k = x_{k,\max}$ can be discretized by

$$(P^{m+1})_{k,j} \approx \left((BC_1)_{k,j} - \sum_{l=1}^{N_1 - 1} (D_{1,N_1})_{k,l} (P^{m+1})_{l,j} \right) / (D_{1,N_1})_{k,k}, \quad k = 0, N_1,$$

$$(P^{m+1})_{i,l} \approx \left((BC_2)_{i,l} - \sum_{k=1}^{N_2 - 1} (P^{m+1})_{i,k} (D_{1,N_2}^T)_{k,l} \right) / (D_{1,N_2})_{l,l}, \quad l = 0, N_2,$$

$$(36)$$

where $i = 1, \dots, N_1 - 1$, $j = 1, \dots, N_2 - 1$. Combining (33) with (36) leads to a linear system for $\tilde{P} = P^{m+1}(1:N_1 - 1, 1:N_2 - 1)$ given by

$$\overline{D}_{2,N_1}\widetilde{P} + \widetilde{P}\overline{D}_{2,N_2}^T \approx \overline{H},\tag{37}$$

where \overline{D}_{2,N_k} are the matrices constructed from D_{2,N_k} by taking the interior elements $(i, j) = (1 : N_1 - 1, 1 : N_2 - 1)$ and \overline{H} is from the right-hand side of (33) together with some manipulations of (36). Since \overline{D}_{2,N_k} are diagonalizable, Eq. (37) can be solved using the matrix-diagonalization procedure as follows. Let $\overline{D}_{2,N_k} = R_k \Sigma_k R_k^{-1}$ and define $\widehat{P} := R_1^{-1} \widetilde{P}(R_2^T)^{-1}$ and $\widehat{H} := R_1^{-1} \overline{H}(R_2^T)^{-1}$, where $\Sigma_k = \text{diag}\{\sigma_{k,1}, \dots, \sigma_{k,N_k-1}\}$. Recall that \overline{D}_{2,N_k} is singular with one dimensional null space [26]. Thus, we take $\sigma_{k,1} = 0$. Then, a particular solution of (37) is given by

$$\widehat{P}_{i,j} \approx \begin{cases} 0 & i = j = 1, \\ \frac{\widehat{H}_{i,j}}{\sigma_{1,i} + \sigma_{2,j}}, & \text{otherwise.} \end{cases}$$
(38)

Hence, the pressure can be calculated with the formula $P^{m+1}(1:N_1-1,1:N_2-1) = \tilde{P} = R_1 \hat{P} R_2^T$ and the boundary values (without four corners) of the pressure P^{m+1} are determined by (36). Finally, since (36) holds for $i = 0, N_1, j = 0, N_2$, the values of P^{m+1} at the four corner points can be also recovered by (36).

4.2. Matrix-diagonalization method to Helmholtz equation (11)

Discretizing the Helmholtz equation (11) using (17) and (18) results in finite difference systems for U^{m+1} and V^{m+1} , respectively. For U^{m+1} , the finite system is described by

$$\frac{3}{2h}U^{m+1} - \nu \left(D_{2,N_1}U^{m+1} + U^{m+1}D_{2,N_2}^T \right) = G :\approx F_1^{m+1} - D_{1,N_1}P^{m+1}.$$
(39)

As the case of the discretization for pressure, let $\tilde{U} = U^{m+1}(1:N_1-1,1:N_2-1)$ and $\tilde{D}_{2,N_k} = D_{2,N_k}(1:N_1-1,1:N_2-1)$. Then, (39) leads to

$$\frac{3}{2h}\widetilde{U} - \nu \left(\widetilde{D}_{2,N_1}\widetilde{U} + \widetilde{U}\widetilde{D}_{2,N_2}^T\right) = \overline{G},\tag{40}$$

where \overline{G} is defined from G in a similar way to how \overline{H} was defined from H in the previous subsection. Since \widetilde{D}_{2,N_k} are also diagonalizable, let $\widetilde{D}_{2,N_k} = Q_k \Lambda_k Q_k^{-1}$ and define $\widehat{U} := Q_1^{-1} \widetilde{U} (Q_2^T)^{-1}$ and $\widehat{G} := Q_1^{-1} \overline{G} (Q_2^T)^{-1}$, where $\Lambda_k = \text{diag}\{\lambda_{k,1}, \dots, \lambda_{k,N_k-1}\}$. Then, the solution of (40) can be directly calculated by the following formula

$$\widehat{U}_{i,j} \approx \frac{\widehat{G}_{i,j}}{\frac{3}{2h} - \nu(\lambda_{1,i} + \lambda_{2,j})}, \quad i = 1, 2, \cdots, N_1 - 1, \ j = 1, 2, \cdots, N_2 - 1$$
(41)

and $\tilde{U} = Q_1 \hat{U} Q_2^T$. This completes the computational scheme of U^{m+1} . The approximation of V^{m+1} is conducted with a similar way.

5. Numerical examples

In this section, we perform numerical experiments to illustrate the accuracy and effectiveness of the proposed numerical algorithm. The numerical results of three test problems are given. For simplicity, the proposed scheme discussed in the previous sections is simply denoted by ECM2. The other notations ERK2 and Fixed2 are the corresponding numerical methods, for which the approximation $\pi_{i,j}^{m-1}$ is obtained by using (23) and (24), respectively. Here, the stopping criterion of the fixed point iteration is taken so that the following inequality is satisfied

$$\|\boldsymbol{\alpha}^{(n+1)} - \boldsymbol{\alpha}^{(n)}\|_2 \leq h^2,$$

where $\boldsymbol{\alpha}^{(n)}$ is the *n*th iterative solution and the initial approximation $\boldsymbol{\alpha}^{(0)} = \mathbf{0}$. The velocity $[U^1, V^1]$ at time *h* in all numerical examples is computed with the first order backward difference formula and the backward Euler scheme instead of the second order backward difference formula and mid-point rules used in ECM2.

Table 1	
The spatial convergence rate on [0, 2] for $u, v, p, \nabla \cdot \mathbf{u}$ and $\nabla \cdot \mathbf{f}$ ($v = 10^{-2}, k$	$k = 2, h = 2^{-9}$.

$N_1 = N_2$	<i>u</i> -error	Rate	v-error	Rate	<i>p</i> -error	Rate	$ abla \cdot \mathbf{u}$	Rate	$\nabla \cdot \mathbf{f}$ -error	Rate
20	$1.12 imes 10^{-2}$	-	$1.12 imes 10^{-2}$	-	$1.66 imes 10^{+0}$	-	$8.40 imes10^{-2}$	-	$6.33 imes10^{+1}$	-
40	$1.57 imes 10^{-3}$	2.83	1.57×10^{-3}	2.83	1.52×10^{-1}	3.45	$1.38 imes 10^{-2}$	2.60	$1.00 imes 10^{+1}$	2.66
80	$1.83 imes 10^{-4}$	3.10	$1.83 imes 10^{-4}$	3.10	$1.37 imes 10^{-2}$	3.47	$1.46 imes 10^{-3}$	3.24	$1.09 imes10^{+0}$	3.20
160	$1.78 imes 10^{-5}$	3.36	$1.78 imes 10^{-5}$	3.36	$1.42 imes 10^{-3}$	3.27	$9.82 imes 10^{-5}$	3.89	$7.49 imes 10^{-2}$	3.86
320	$3.27 imes10^{-6}$	2.44	$3.27 imes10^{-6}$	2.44	$1.75 imes 10^{-4}$	3.03	$5.32 imes 10^{-6}$	4.21	$4.73 imes 10^{-3}$	3.99

Table 2 The spatial convergence rate on [0, 2] for $u, v, p, \nabla \cdot \mathbf{u}$ and $\nabla \cdot \mathbf{f}$ ($v = 10^{-3}, k = 2, h = 2^{-9}$).

$N_1 = N_2$	<i>u</i> -error	Rate	v-error	Rate	<i>p</i> -error	Rate	$ abla \cdot \mathbf{u}$	Rate	$\nabla \cdot \mathbf{f}$ -error	Rate
20	$1.70 imes 10^{-2}$	-	$1.70 imes 10^{-2}$	-	$1.67 imes10^{+0}$	-	$1.39 imes 10^{-1}$	-	$1.07 imes 10^{+2}$	-
40	$2.46 imes10^{-3}$	2.79	$2.46 imes 10^{-3}$	2.79	1.52×10^{-1}	3.45	$2.80 imes 10^{-2}$	2.31	$2.06 imes 10^{+1}$	2.38
80	$3.07 imes10^{-4}$	3.00	$3.07 imes 10^{-4}$	3.00	$1.38 imes 10^{-2}$	3.47	$3.76 imes 10^{-3}$	2.90	$2.80 imes10^{+0}$	2.88
160	$3.50 imes10^{-5}$	3.13	$3.50 imes 10^{-5}$	3.13	$1.43 imes 10^{-3}$	3.26	$4.28 imes 10^{-4}$	3.13	$3.19 imes 10^{-1}$	3.13
320	4.41×10^{-6}	2.99	4.41×10^{-6}	2.99	$1.78 imes 10^{-4}$	3.01	$3.06 imes 10^{-5}$	3.81	2.25×10^{-2}	3.83

The temporal convergence rate of ECM2 on $t \in [0, 2]$ for $u, v, p, \nabla \cdot \mathbf{u}$ and $\nabla \cdot \mathbf{f}$ ($v = 10^{-2}, k = 4, N_1 = N_2 = 512$).

Table 3

h	<i>u</i> -error	Rate	v-error	Rate	<i>p</i> -error	Rate	$ abla \cdot \mathbf{u}$	Rate	$\nabla \cdot \mathbf{f}$ -error	Rate
2 ⁻²	$2.58 imes 10^{-1}$	-	2.58×10^{-1}	-	$8.30 imes 10^{-1}$	-	$1.78 imes 10^{-1}$	-	$3.26 imes 10^{+1}$	-
2^{-3}	$3.46 imes 10^{-2}$	2.90	$3.46 imes 10^{-2}$	2.90	$1.19 imes 10^{-1}$	2.81	$3.31 imes 10^{-2}$	2.43	$4.95\times10^{+0}$	2.72
2^{-4}	$1.07 imes10^{-2}$	1.69	$1.07 imes 10^{-2}$	1.69	$2.81 imes 10^{-2}$	2.08	$9.03 imes10^{-3}$	1.87	$1.53 imes10^{+0}$	1.70
2^{-5}	$3.07 imes 10^{-3}$	1.80	$3.07 imes 10^{-3}$	1.80	$9.56 imes 10^{-3}$	1.55	2.05×10^{-3}	2.14	4.81×10^{-1}	1.67
2^{-6}	$8.25 imes 10^{-4}$	1.90	$8.25 imes 10^{-4}$	1.90	3.23×10^{-3}	1.57	$5.58 imes10^{-4}$	1.88	$1.88 imes 10^{-1}$	1.36
2 ⁻⁷	2.16×10^{-4}	1.93	2.16×10^{-4}	1.93	$1.26 imes 10^{-3}$	1.35	$9.50 imes10^{-5}$	2.55	$8.22 imes 10^{-2}$	1.19

Example 1. To check the order of convergence of the proposed scheme, we test the problem (1) whose exact solution is given by (for example, see [8,33])

$$u(t, x_1, x_2) = -\cos(kx_1)\sin(kx_2)\exp(-2k^2\nu t),$$

$$v(t, x_1, x_2) = \sin(kx_1)\cos(kx_2)\exp(-2k^2\nu t),$$

$$p(t, x_1, x_2) = -\frac{1}{4}(\cos(2kx_1) + \cos(2kx_2))\exp(-4k^2\nu t),$$
(42)

where ν is a kinetic viscosity and *k* is a nonzero constant. The initial and boundary conditions are taken from the exact solution (42) and the computational domain is $[0, 2\pi]^2$.

To measure the computational error for the proposed scheme, we use the root mean square (RMS) error which is defined by

$$Err_{RMS} := \max_{0 \le m \le M} \sqrt{\frac{1}{(N_1 + 1)(N_2 + 1)} \sum_{i=0, j=0}^{N_1, N_2} (\phi_{i, j}^m - \phi(t_m, x_{1, i}, x_{2, j}))^2},$$
(43)

where $\phi(t_m, x_{1,i}, x_{2,j})$ is the exact solution at grid points $(x_{1,i}, x_{2,j})$, at time $t = t_m$ and $\phi_{i,j}^m$ is its approximation. Here, ϕ stands for u, v, p and $\nabla \cdot \mathbf{f}$.

The spatial convergence rate is calculated on the time interval [0, 2] with the fixed value k = 2 and the fixed time step size $h = 2^{-9}$ for two different Reynolds numbers $v = 10^{-2}$ and $v = 10^{-3}$. The numerical results are listed in Tables 1 and 2. From the tables, it is clear that the rate of spatial convergence for both pressure and velocity is almost 3. That is, the present algorithm has numerically third-order spatial convergence. Besides the spatial convergence analysis for pressure and velocity, we calculate the spatial convergence order for $\nabla \cdot \mathbf{f}$. Further, we list the values of $\nabla \cdot \mathbf{u}$ which are shown to be zero as Δx goes to zero.

In order to investigate the temporal convergence, we simulate the problem with the fixed value k = 4 and the fixed number of spatial grids $N_1 = N_2 = 512$ for different Reynolds numbers $v = 10^{-2}$ (Tables 3–5) and $v = 10^{-3}$ (Tables 6–8). The simulation is carried out by ranging the time step size from 2^{-2} to 2^{-7} and the convergence rate is calculated on the time interval [0, 2]. The numerical results are listed in Tables 3–8, where Tables 3 and 6 are for ECM2, Tables 4 and 7 are for ERK2 and Tables 5 and 8 are for Fixed2. The results show that all schemes have second order temporal convergence numerically. Also, in the sense of the numerical accuracy, the three methods have similar accuracy.

Table	e 4
-------	-----

Table 7

The temporal convergence rate of ERK2 on $t \in [0, 2]$ for $u, v, p, \nabla \cdot \mathbf{u}$ and $\nabla \cdot \mathbf{f}$ ($v = 10^{-2}$, k = 4, $N_1 = N_2 = 512$).

h	<i>u</i> -error	Rate	v-error	Rate	p-error	Rate	$ abla \cdot \mathbf{u}$	Rate	$\nabla \cdot \mathbf{f}$ -error	Rate
2 ⁻²	$2.51 imes 10^{-1}$	-	$2.51 imes 10^{-1}$	-	$8.26 imes 10^{-1}$	-	1.81×10^{-1}	-	$2.14\times10^{+1}$	-
2^{-3}	$3.36 imes 10^{-2}$	2.90	$3.36 imes 10^{-2}$	2.90	$1.78 imes 10^{-1}$	2.22	$3.53 imes 10^{-2}$	2.36	$5.32 imes10^{+0}$	2.01
2^{-4}	$1.07 imes 10^{-2}$	1.65	$1.07 imes 10^{-2}$	1.65	$4.54 imes 10^{-2}$	1.97	9.39×10^{-3}	1.91	$1.56 imes10^{+0}$	1.77
2^{-5}	$3.07 imes 10^{-3}$	1.80	3.07×10^{-3}	1.80	1.43×10^{-2}	1.66	2.07×10^{-3}	2.18	4.81×10^{-1}	1.70
2^{-6}	$8.25 imes 10^{-4}$	1.90	$8.25 imes 10^{-4}$	1.90	$4.49 imes 10^{-3}$	1.67	$5.60 imes 10^{-4}$	1.88	1.88×10^{-1}	1.36
2^{-7}	$2.16 imes 10^{-4}$	1.93	$2.16 imes 10^{-4}$	1.93	1.59×10^{-3}	1.50	9.51×10^{-5}	2.56	8.21×10^{-2}	1.19

Table 5			
The temporal converge	ence rate of Fixed2 on $t \in [0, 2]$ for $t \in [0, 2]$	$u, v, p, \nabla \cdot \mathbf{u}$ and $\nabla \cdot \mathbf{f} (v = \mathbf{f})$	10^{-2} , $k = 4$, $N_1 = N_2 =$

h	<i>u</i> -error	Rate	v-error	Rate	<i>p</i> -error	Rate	$ abla \cdot \mathbf{u}$	Rate	$\nabla \cdot \mathbf{f}$ -error	Rate
2 ⁻²	2.54×10^{-1}	-	2.54×10^{-1}	-	$5.88 imes 10^{-1}$	-	$1.92 imes 10^{-1}$	-	$4.45\times10^{+1}$	-
2^{-3}	$3.67 imes 10^{-2}$	2.79	$3.67 imes 10^{-2}$	2.79	$1.17 imes 10^{-1}$	2.33	3.21×10^{-2}	2.58	$6.69 imes10^{+0}$	2.73
2^{-4}	$1.10 imes 10^{-2}$	1.74	$1.10 imes 10^{-2}$	1.74	$3.27 imes 10^{-2}$	1.84	$8.76 imes10^{-3}$	1.87	$1.63 imes10^{+0}$	2.04
2^{-5}	3.11×10^{-3}	1.82	3.11×10^{-3}	1.82	$1.09 imes 10^{-2}$	1.59	$2.04 imes 10^{-3}$	2.11	$4.82 imes 10^{-1}$	1.75
2^{-6}	$8.29 imes 10^{-4}$	1.91	$8.29 imes 10^{-4}$	1.91	$3.58 imes 10^{-3}$	1.60	$5.58 imes10^{-4}$	1.87	$1.88 imes 10^{-1}$	1.36
2^{-7}	2.16×10^{-4}	1.94	$2.16 imes10^{-4}$	1.94	$1.35 imes 10^{-3}$	1.40	$9.54 imes10^{-5}$	2.55	$8.22 imes 10^{-2}$	1.19

512).

Table 6 The temporal convergence rate of ECM2 on $t \in [0, 2]$ for $u, v, p, \nabla \cdot \mathbf{u}$ and $\nabla \cdot \mathbf{f}$ ($v = 10^{-3}$, k = 4, $N_1 = N_2 = 512$).

h	<i>u</i> -error	Rate	v-error	Rate	<i>p</i> -error	Rate	$ abla \cdot \mathbf{u}$	Rate	$\nabla \cdot \mathbf{f}$ -error	Rate
2 ⁻²	$3.37 imes 10^{-1}$	-	3.38×10^{-1}	-	$1.17\times10^{+0}$	-	$2.43\times10^{+0}$	-	$1.57\times10^{+2}$	-
2^{-3}	$5.72 imes 10^{-2}$	2.56	$5.72 imes 10^{-2}$	2.56	$1.49 imes 10^{-1}$	2.98	$4.55 imes 10^{-2}$	5.74	$7.64 imes10^{+0}$	4.36
2^{-4}	$1.45 imes 10^{-2}$	1.98	$1.45 imes 10^{-2}$	1.98	$3.07 imes 10^{-2}$	2.28	1.61×10^{-2}	1.50	$2.27 imes10^{+0}$	1.75
2^{-5}	3.52×10^{-3}	2.04	3.52×10^{-3}	2.04	$1.10 imes 10^{-2}$	1.49	3.68×10^{-3}	2.13	$6.71 imes 10^{-1}$	1.76
2^{-6}	$9.03 imes 10^{-4}$	1.96	$9.03 imes10^{-4}$	1.96	4.02×10^{-3}	1.45	1.05×10^{-3}	1.80	2.06×10^{-1}	1.70
2 ⁻⁷	2.29×10^{-4}	1.98	2.29×10^{-4}	1.98	$1.50 imes 10^{-3}$	1.42	$3.91 imes 10^{-4}$	1.43	$7.81 imes 10^{-2}$	1.40

The temporal convergence rate of ERK2 on $t \in [0, 2]$ for $u, v, p, \nabla \cdot \mathbf{u}$ and $\nabla \cdot \mathbf{f}$ ($v = 10^{-3}, k = 4, N_1 = N_2 = 512$).

h	u-error	Rate	v-error	Rate	p-error	Rate	$ abla \cdot \mathbf{u}$	Rate	$\nabla \cdot \mathbf{f}$ -error	Rate
2 ⁻²	$3.09 imes 10^{-1}$	-	$3.09 imes10^{-1}$	-	$9.67 imes 10^{-1}$	-	$5.38 imes 10^{-1}$	-	$4.44\times10^{+1}$	-
2^{-3}	$1.94 imes 10^{-1}$	0.67	$1.94 imes 10^{-1}$	0.67	$1.97 imes 10^{-1}$	2.29	$1.09\times10^{+0}$	-1.02	$1.21 imes 10^{+2}$	-1.44
2^{-4}	$1.40 imes 10^{-2}$	3.80	$1.40 imes 10^{-2}$	3.80	$4.69 imes 10^{-2}$	2.07	$1.59 imes 10^{-2}$	6.10	$2.42\times10^{+0}$	5.64
2 ⁻⁵	3.52×10^{-3}	1.99	3.52×10^{-3}	1.99	1.61×10^{-2}	1.54	$3.79 imes 10^{-3}$	2.07	$6.84 imes10^{-1}$	1.82
2^{-6}	$9.03 imes10^{-4}$	1.96	$9.03 imes10^{-4}$	1.96	$5.34 imes 10^{-3}$	1.59	$1.05 imes 10^{-3}$	1.85	$2.07 imes10^{-1}$	1.72
2 ⁻⁷	2.29×10^{-4}	1.98	2.29×10^{-4}	1.98	1.84×10^{-3}	1.54	$3.91 imes 10^{-4}$	1.43	$7.80 imes 10^{-2}$	1.41

Table 8 The temporal convergence rate of Fixed2 on $t \in [0, 2]$ for $u, v, p, \nabla \cdot \mathbf{u}$ and $\nabla \cdot \mathbf{f}$ ($v = 10^{-3}, k = 4, N_1 = N_2 = 512$).

h	<i>u</i> -error	Rate	v-error	Rate	<i>p</i> -error	Rate	$ abla \cdot \mathbf{u}$	Rate	$\nabla \cdot \mathbf{f}$ -error	Rate
2-2	$3.54 imes10^{-1}$	-	$3.54 imes10^{-1}$	-	$1.10\times10^{+0}$	-	$1.55 imes10^{+0}$	-	$8.36 imes10^{+1}$	-
2^{-3}	$9.66 imes 10^{-2}$	1.88	$9.66 imes 10^{-2}$	1.88	$2.25 imes 10^{-1}$	2.28	$8.42 imes 10^{-2}$	4.21	$1.01 imes 10^{+1}$	3.04
2^{-4}	$2.45 imes 10^{-2}$	1.98	$2.45 imes 10^{-2}$	1.98	5.31×10^{-2}	2.08	$1.79 imes 10^{-2}$	2.24	$2.36\times10^{+0}$	2.11
2^{-5}	$4.90 imes 10^{-3}$	2.32	$4.90 imes 10^{-3}$	2.32	$1.23 imes 10^{-2}$	2.11	$3.88 imes 10^{-3}$	2.20	$6.72 imes 10^{-1}$	1.81
2^{-6}	$9.87 imes 10^{-4}$	2.31	$9.87 imes 10^{-4}$	2.31	4.38×10^{-3}	1.49	1.13×10^{-3}	1.78	2.06×10^{-1}	1.71
2^{-7}	$2.29 imes10^{-4}$	2.11	$2.29 imes10^{-4}$	2.11	$1.59 imes 10^{-3}$	1.46	$4.05 imes 10^{-4}$	1.48	$8.08 imes 10^{-2}$	1.35

In order to investigate the effect of viscosity ν for the proposed scheme when the time step sizes decrease, we simulate the problem with the fixed value k = 4 and the fixed number of spatial grids $N_1 = N_2 = 256$ for different Reynolds numbers $\nu = 1$ and $\nu = 10^{-4}$. The numerical results are listed in Table 9. For $\nu = 1$, the numerical errors decrease as h decreases. However, for small $\nu = 10^{-4}$, the numerical errors do not decrease even though h is decreasing.

Example 2. The next problem is the two-dimensional lid-driven cavity problem. This problem is often used to demonstrate the accuracy and efficiency of numerical schemes for incompressible flows. The problem has a great scientific interest because it displays almost all fluid mechanical phenomena for incompressible viscous flows in the simple geometric settings. The computational domain is the unit square $[0, 1]^2$ and the initial and boundary conditions are as follows

X. Piao et al. / Journal of Computational Physics 283 (2015) 189-204

Table 9	
The numerical error performances of ECM2 on $t \in [0, 2]$	for $u, v, p, \nabla \cdot \mathbf{u}$ and $\nabla \cdot \mathbf{f}$ ($k = 4, N_1 = N_2 = 256$).

ν	h	<i>u</i> -error	v-error	<i>p</i> -error	$ abla \cdot \mathbf{u}$	$\nabla \cdot \mathbf{f}$ -error
1	2 ⁻²	$8.43 imes 10^{-2}$	$8.43 imes 10^{-2}$	$1.00 imes 10^{+0}$	$5.19 imes10^{-1}$	$4.68\times10^{+0}$
	2-3	$6.88 imes 10^{-2}$	6.88×10^{-2}	9.46×10^{-1}	4.45×10^{-1}	$7.36 imes10^{+0}$
	2^{-4}	$4.80 imes 10^{-2}$	$4.80 imes 10^{-2}$	8.40×10^{-1}	$3.28 imes 10^{-1}$	$9.87 imes10^{+0}$
	2^{-5}	$3.18 imes 10^{-2}$	3.18×10^{-2}	6.55×10^{-1}	1.82×10^{-1}	$1.12 imes 10^{+1}$
	2 ⁻⁶	2.20×10^{-2}	2.20×10^{-2}	4.55×10^{-1}	7.01×10^{-2}	$9.55 imes10^{+0}$
	2 ⁻⁷	$9.85 imes 10^{-3}$	9.85×10^{-3}	2.82×10^{-1}	2.06×10^{-2}	$8.09 imes10^{+0}$
	2-8	3.37×10^{-3}	3.37×10^{-3}	1.62×10^{-1}	5.26×10^{-3}	$6.20 imes10^{+0}$
	2 ⁻⁹	9.94×10^{-4}	9.94×10^{-4}	8.88×10^{-2}	1.24×10^{-3}	$4.36\times10^{+0}$
10 ⁻⁴	2^{-2}	4.42×10^{-1}	$4.43 imes 10^{-1}$	$2.86 imes10^{+0}$	$9.10 imes10^{+0}$	$1.45\times10^{+2}$
	2-3	6.61×10^{-2}	6.61×10^{-2}	2.16×10^{-1}	1.76×10^{-1}	$8.09 imes10^{+0}$
	2^{-4}	1.75×10^{-2}	1.75×10^{-2}	4.42×10^{-2}	2.61×10^{-2}	$2.30 imes10^{+0}$
	2^{-5}	$4.35 imes 10^{-3}$	$4.35 imes 10^{-3}$	1.75×10^{-2}	9.07×10^{-3}	$6.77 imes 10^{-1}$
	2 ⁻⁶	1.07×10^{-3}	1.07×10^{-3}	8.31×10^{-3}	3.78×10^{-3}	3.72×10^{-1}
	2^{-7}	$2.60 imes 10^{-4}$	$2.60 imes 10^{-4}$	6.24×10^{-3}	1.48×10^{-3}	2.88×10^{-1}
	2 ⁻⁸	$8.25 imes 10^{-5}$	$8.25 imes 10^{-5}$	$5.75 imes 10^{-3}$	1.19×10^{-3}	$4.55 imes 10^{-1}$
	2^{-9}	7.40×10^{-5}	$7.40 imes10^{-5}$	$5.65 imes 10^{-3}$	$1.75 imes 10^{-3}$	$1.32\times10^{+0}$

$$\mathbf{u}(0,\mathbf{x}) = \begin{cases} (1,0)^T, & x_2 = 1 \& \mathbf{x} \in \Gamma, \\ (0,0)^T, & \text{otherwise}, \end{cases}, \quad \mathbf{u}(t,\mathbf{x}) = \begin{cases} (1,0)^T, & x_2 = 1 \& \mathbf{x} \in \Gamma, \\ (0,0)^T, & x_2 \neq 1 \& \mathbf{x} \in \Gamma, \end{cases} \quad t > 0.$$
(44)

Let

$$Err(mh) = \max_{i,j} \left(\left(u_{i,j}^{m+1} - u_{i,j}^{m} \right)^2 + \left(v_{i,j}^{m+1} - v_{i,j}^{m} \right)^2 \right)^{1/2}$$

be the maximum root square error between two consequent times t_m and t_{m+1} . When $Err(mh) < 10^{-7}$, we regard the cavity problem as a steady state. In order to see the streamline contours of the solution, we simulate the problem with spatial grids of fixed size $N_1 = N_2 = 256$ and fixed time step h = 0.01 for different Reynolds numbers Re = 1000, 3200, 5000, 7500. The numerical results are exhibited in Fig. 1. One can see the typical separations and secondary vortices at the bottom corners of the cavity, as well as the top left. These numerical results are in very good agreement with the benchmark results of Ghia et al. [15] and other established numerical results [5,7,17,31,33]. It confirms that the proposed method gives quantitatively accurate solutions.

Also, we calculate the horizontal velocities on the vertical centerline of the cavity and compare the numerical results with those obtained from Ghia [15], the second explicit method ERK2 and the implicit second method Fixed2. For all computations, the same fine grid mesh spacing $\frac{1}{256}$ and the time step size 0.01 are used. In particular, when Re = 7500, two numerical simulations are generated. We use the time step size h = 0.001 only for ERK2 because ERK2 does not reach the steady state with h = 0.01, while we use the time step sizes h = 0.01 and h = 0.001 for the other three methods. The comparisons are exhibited in Fig. 2. As shown in Fig. 2, the two implicit type methods, the proposed scheme and Fixed2 have similar performance and results are close to that of Ghia [15]. In contrast, the explicit method, ERK2, has a different performance compared to the other methods.

In Table 10, in order to examine effectiveness of ECM2, we list CPU time for IVPs, CPU time for the rest of the time step, and overall CPU time for the time step obtained by three methods (ECM2, ERK2 and Fixed2) with spatial step size $\frac{1}{256}$ and time step size 0.01 by varying the Reynolds number *Re*. Here, CPU time for IVPs means an average per time step. From Table 10, ECM2 needs similar overall CPU time to Fixed2, while ECM2 requires less CPU time than Fixed2 for solving IVPs, which is one of main focuses in this paper to develop an iteration free method. From the results discussed in Fig. 2 and Table 10, we can conclude that the proposed scheme is superior to the other methods.

Example 3. As our last example, we consider the doubly periodic shear layer flow problem [6,11,12] defined on the unit square $\Omega = [0, 1]^2$. The initial conditions are taken as

$$u(0, \mathbf{x}) = \mathbb{1}_{x_2 \le \frac{1}{2}} \tanh\left(\rho\left(x_2 - \frac{1}{4}\right)\right) + \mathbb{1}_{x_2 > \frac{1}{2}} \tanh\left(\rho\left(\frac{3}{4} - x_2\right)\right), \qquad v(0, \mathbf{x}) = \frac{1}{20}\sin(2\pi x_1), \tag{45}$$

which correspond to a layer of thickness $\mathcal{O}(\frac{1}{\rho})$. Here, $\mathbb{1}_A$ denotes the characteristic function.

We first calculate the vorticities of the solution of the shear-layer rollup problem with $\rho = 30$, $Re = 10^5$, the number of spatial grids $N_1 = N_2 = 128$ and the time step size h = 0.01 at different times t = 0.8, 1.0, 1.2 and 1.5. The numerical results are displayed in Fig. 3.

To show the effectiveness of the proposed method, we next estimate the conservations of the energy and the enstrophy defined by



Fig. 1. The steady-state stream contours for different Reynolds numbers with $N_1 = N_2 = 256$ and h = 0.01.

$$Energy(t) = \int_{\Omega} \left(u^{2}(t, \mathbf{x}) + v^{2}(t, \mathbf{x}) \right) d\mathbf{x} \approx \Delta x_{1} \Delta x_{2} \sum_{i=0}^{N_{1}} \sum_{j=0}^{N_{2}} \left(u^{2}(t, \mathbf{x}_{i,j}) + v^{2}(t, \mathbf{x}_{i,j}) \right) d\mathbf{x}$$

and

Enstrophy(t) =
$$\int_{\Omega} w^2(t, \mathbf{x}) d\mathbf{x} \approx \Delta x_1 \Delta x_2 \sum_{i=0}^{N_1} \sum_{j=0}^{N_2} w^2(t, \mathbf{x}_{i,j}),$$

where $w(t, \mathbf{x})$ is the vorticity defined by

$$w(t, \mathbf{x}) = \nabla \times \mathbf{u}(t, \mathbf{x}) = \frac{\partial}{\partial x_1} v(t, \mathbf{x}) - \frac{\partial}{\partial x_2} u(t, \mathbf{x}).$$

In Fig. 4, we compare the numerical results obtained by the three methods, ECM2, ERK2 and Fixed2. One can confirm that the present scheme is superior to the other methods in the sense of both conservations. Note that all schemes satisfy that the total energy decreases with time.



Fig. 2. Profiles of *u*-velocity along vertical line through geometric center in cavity obtained from different schemes by varying Reynolds numbers with $N_1 = N_2 = 256$ and h = 0.01 (especially for the case Re = 7500, h = 0.001 in the above figure only for ERK2 and h = 0.001 in the below figure for three methods).

Table 10Comparison of ECM2, ERK2 and Fixed2 with $N_1 = N_2 = 256$ and h = 0.01.

Re	ECM2	ECM2			ERK2			Fixed2		
	IVPs	Rest	Total	IVPs	Rest	Total	IVPs	Rest	Total	
1000	0.0803	0.0849	0.1652	0.0711	0.0878	0.1589	0.0901	0.0781	0.1682	
3200	0.0778	0.0859	0.1637	0.0704	0.0869	0.1574	0.0887	0.0770	0.1657	
5000	0.0790	0.0842	0.1632	0.0693	0.0884	0.1577	0.0868	0.0771	0.1638	
7500	0.0779	0.0855	0.1634	-	-	-	0.0884	0.0781	0.1665	



Fig. 3. Vorticities of the solution for the shear-layer rollup problem at time t = 0.8, 1.0, 1.2 and 1.5 with $\rho = 30, Re = 10^5, N_1 = N_2 = 128$ and h = 0.01.

Also, we examine the effect of the time step size and the viscosity number for energy conservation by varying the time step sizes h = 0.03, 0.02, 0.01, 0.005 and the viscosities $v = 10^{-k}$, k = 5, 6, 7, 8. The results are shown in Fig. 5. One can see that the smaller the time step size and the viscosity number are, the better the conservation is.

6. Conclusion and further discussion

A new iteration free backward semi-Lagrangian scheme for solving an incompressible Navier–Stokes equation is developed with ECM framework for finding the departure points of the characteristic curves and a projection method to split the steady state Stokes equation into the velocity and the pressure. Unlike the traditional method of calculating departure points



Fig. 4. Comparisons of the energy and enstrophy conservation properties among several methods for the shear-layer rollup problem with $\rho = 30$, $Re = 10^5$, when $N_1 = N_2 = 128$ and h = 0.01.



Fig. 5. Energy conservation properties of the proposed scheme with fixed $\nu = 1.0 \times 10^{-5}$ varying the time step sizes (on the left) and with fixed time step size h = 0.005 different $\nu = 1.0 \times 10^{-5}$, 1.0×10^{-6} , 1.0×10^{-7} , 1.0×10^{-8} (on the right), when $N_1 = N_2 = 128$ and $\rho = 30$.

in an existing implicit scheme or explicit scheme, we suggest a new method that is an iteration free explicit scheme but has all the good properties of the conventional second-order implicit method. With several numerical results, it is shown that the proposed method obtains outstanding numerical results compared to existing methods. Also, it is shown that the proposed scheme has a good conservation property of the total energy for small time steps and small viscosity numbers.

In order to fully explore the efficiency of ECM, several extended issues are currently being pursued. One is to develop a higher order time integration scheme which is not possible in implicit scheme due to the self-consistency constraint of the backward semi-Lagrangian method. Notice that all schemes to calculate the velocity, pressure and the characteristic curves are completely iteration free. Hence, another issue is to develop a parallelization algorithm for the proposed method. The proposed method is developed only for two-dimensional problems. Thus, the other challenge is to extend the idea of the proposed method into three dimensional problems. Additionally, the generalization of the proposed idea will be applied to many physical problems containing the convection and diffusion terms. Results along these directions will be reported in the future.

References

- [1] A. Allievi, R. Bermejo, Finite element modified method of characteristics for the Navier–Stokes equations, Int. J. Numer. Methods Fluids 32 (2000) 439–463.
- [2] A. Bermúdez, M.R. Nogueiras, C. Vázquez, Numerical analysis of convection-diffusion-reaction problems with higher order characteristics/finite elements. Part I: time discretization, SIAM J. Numer, Anal. 44 (2006) 1829–1853.
- [3] A. Bermúdez, M.R. Nogueiras, C. Vázquez, Numerical analysis of convection-diffusion-reaction problems with higher order characteristics/finite elements. Part II: fully discretized scheme and quadrature formulas, SIAM J. Numer. Anal. 44 (2006) 1854–1876.
- [4] K. Boukir, Y. Maday, B. Métivet, A high order characteristic method for the incompressible Navier–Stokes equations, Comput. Methods Appl. Mech. Eng. 116 (1994) 211–218.

- [5] O. Botella, On the solution of the Navier–Stokes equations using Chebyshev projection schemes with third-order accuracy in time, Comput. Fluids 26 (1997) 107–116.
- [6] D.L. Brown, M.L. Minion, Performance of under-resolved two-dimensional incompressible flow simulations, J. Comput. Phys. 122 (1995) 165–183.
- [7] C. Bruneau, C. Jouron, An efficient scheme for solving steady incompressible Navier–Stokes equations, J. Comput. Phys. 89 (1990) 389–413.
- [8] A.J. Chorin, Numerical solution of the Navier-Stokes equations, Math. Comput. 22 (1968) 745-762.
- [9] R.E. Ewing, T.F. Russell, Multistep Galerkin methods along characteristics for convection-diffusion problems, in: R. Vichnevetsky, R.S. Stepleman (Eds.), Advances in Computer Methods for Partial Differential Equations IV, IMACS Rutgers University, New Brunswick, NJ, 1981, pp. 28–36.
- [10] R.E. Ewing, H. Wang, A summary of numerical methods for time-dependent advection-dominated partial differential equations, J. Comput. Appl. Math. 128 (2001) 423-445.
- [11] P.F. Fischer, J. Mullen, Filter-based stabilization of spectral element methods, C. R. Acad. Sci. Paris Sér. I Math. 332 (2001) 265–270.
- [12] P.F. Fischer, G.W. Kruse, F. Loth, Spectral element method for transitional flows in complex geometries, J. Sci. Comput. 17 (2002) 81–98.
- [13] B. Fornberg, A Practical Guide to Pseudospectral Methods, Cambridge University Press, Cambridge, UK, 1996.
- [14] P. Galan del Sastre, R. Bermejo, A comparison of semi-Lagrangian and Lagrange–Galerkin hp-FEM methods in convection–diffusion problems, Commun. Comput. Phys. 9 (4) (2011) 1020–1039.
- [15] U. Ghia, K. Ghia, C. Shin, High-Re solutions for incompressible flow using the Navier-Stokes equations and a multigrid method, J. Comput. Phys. 48 (1982) 387-411.
- [16] J.L. Guermond, J. Shen, Velocity-correction projection methods for incompressible flows, SIAM J. Numer. Anal. 41 (1) (2003) 112–134.
- [17] S. Hou, Q. Zou, S. Chen, G. Doolen, A.C. Cogley, Simulation of cavity flow by the lattice Boltzmann method, J. Comput. Phys. 118 (1995) 329-347.
- [18] H. Johnston, J.G. Liu, Accurate, stable and efficient Navier–Stokes solvers based on explicit treatment of the pressure term, J. Comput. Phys. 199 (2004) 221–259.
- [19] D. Kim, O. Song, H. Ko, A semi-Lagrangian CIP fluid solver without dimensional splitting, Comput. Graph. Forum 27 (2008) 467-475.
- [20] P. Kim, X. Piao, S.D. Kim, An error corrected Euler method for solving stiff problems based on Chebyshev collocation, SIAM J. Numer. Anal. 49 (2011) 2211–2230.
- [21] S.D. Kim, X. Piao, D. Kim, P. Kim, Convergence on error correction methods for solving initial value problems, J. Comput. Appl. Math. 236 (2012) 4448-4461.
- [22] P. Kim, S.D. Kim, E. Lee, Simple ECEM algorithms using function values only, Kyungpook Math. J. 53 (2013) 573–591.
- [23] J.L. McGregor, Economical determination of departure points for semi-Lagrangian models, Mon. Weather Rev. 121 (1993) 221-230.
- [24] C. Min, F. Gibou, A second order accurate projection method for the incompressible Navier–Stokes equations on non-graded adaptive grids, J. Comput. Phys. 219 (2006) 912–929.
- [25] H. Notsu, M. Tabata, A single-step characteristic-curve finite element scheme of second order in time for the incompressible Navier–Stokes equations, J. Sci. Comput. 38 (2009) 1–14.
- [26] R. Peyret, Spectral Methods for Incompressible Viscous Flow, Springer, New York, 2002.
- [27] X. Piao, J. Kwon, D. Yi, S.D. Kim, P. Kim, An iteration free backward semi-Lagrangian scheme for guiding center problems, SIAM J. Numer. Anal., to appear.
- [28] A. Robert, A stable numerical integration scheme for the primitive meteorological equations, Atmos.-Ocean 19 (1981) 35-46.
- [29] P.K. Smolarkiewicz, J.A. Pudykiewicz, A class of semi-Lagrangian approximations for fluids, J. Atmos. Sci. 49 (22) (1992) 2082-2096.
- [30] R. Shukla, M. Tatineni, X. Zhong, Very high-order compact finite difference schemes on non-uniform grids for incompressible Navier–Stokes equations, J. Comput. Phys. 224 (2007) 1064–1094.
- [31] A. Stephane, V. Stephane, A 2D compact fourth-order projection decomposition method, J. Comput. Phys. 206 (2005) 252–276.
- [32] C. Temperton, A. Staniforth, An efficient two-time-level semi-Lagrangian semi-implicit integration scheme, Q. J. R. Meteorol. Soc. 113 (1987) 1025–1039.
 [33] Z. Tian, X. Liang, P. Yu, A higher order compact finite difference algorithm for solving the incompressible Navier–Stokes equations, Int. J. Numer. Methods Eng. 88 (2011) 511–532.
- [34] D. Xiu, G. Karniadakis, A semi-Lagrangian high-order method for Navier-Stokes equations, J. Comput. Phys. 172 (2001) 658-684.
- [35] D. Xiu, S.J. Sherwin, S. Dong, G.E. Karniadakis, Strong and auxiliary forms of the semi Lagrangian method for incompressible flows, J. Sci. Comput. 25 (2005) 323–346.