

RESEARCH

Open Access



Numerical solution of advection–diffusion type equation by modified error correction scheme

Soyoon Bak¹, Philsu Kim¹, Xiangfan Piao¹ and Sunyoung Bu^{2*}

*Correspondence:

syboo@hongik.ac.kr

²Department of Liberal Arts, Hongik University, Sejong, Republic of Korea
Full list of author information is available at the end of the article

Abstract

In this paper, we consider a numerical solution for nonlinear advection–diffusion equation by a backward semi-Lagrangian method. The numerical method is based on the second-order backward differentiation formula for the material derivative and the fourth-order finite difference formula for the diffusion term along the characteristic curve. A modified error correction scheme is newly introduced to efficiently find the departure point of the characteristic curve. Through several numerical simulations, we demonstrate that the proposed method has second and third convergence orders in time and space, respectively, and is efficient and accurate compared to existing techniques. In addition, it is numerically shown that the proposed method has good properties in terms of energy and mass conservation.

Keywords: Semi-Lagrangian method; Advection–diffusion equations; Burgers equations

1 Introduction

The nonlinear advection–diffusion type equation is one of the popular and important models describing many phenomena derived from various areas of mathematical physics and engineering fields such as gas dynamics, hydrodynamics, shock waves, heat conduction and so on. Also, the type of equation represents the Burgers equation, the heat conduction equation, the nonlinear Schrödinger equation, the Navier–Stokes equation, etc. Hence, the development of efficient and accurate algorithms for solving the equations is of great importance in the computational fluid dynamics community and has been widely studied by many researchers. To solve these equations, there has become a great quantity of research available [1–14] in recent decades. In particular, the error correction method (ECM) [15, 16] for solving the characteristic curve in the backward semi-Lagrangian method (BSLM) [17, 18] was recently developed. This method solves a problem implicitly along the characteristic curves of fluid particles in the opposite direction with large time steps, which is the main advantage of the BSLM [17, 19]. Also, the method does not only have second and third convergence orders in time and space, respectively, but also does not have any iterative processes required to solve a nonlinear initial value problem (IVP) for departure points. To discretize the material derivative and the diffusion term along the characteristic curve in the BSLM framework, the backward difference formula

(BDF2) and the fourth-order finite difference method (FDM) [20] are applied, respectively. The departure point of the characteristic curve is found by the ECM.

The aim of this article is to introduce newly a modified ECM instead of simply using the conventional ECM, motivated by the success of the ECM combined with the BSLM. To do this, we firstly present a modified Euler polygon, for the consideration of the physical domains in which the particles can move (see Sect. 3 for more information). In addition, boundary values are calculated by the same fourth-order finite difference scheme, unlike previous work that used a lower order scheme for boundary conditions. Moreover, to reduce the computational cost, the proposed scheme approximates the Jacobian value as a fixed one while maintaining the scale of the error, whereas the conventional ECM once again performs the interpolation with the derivatives of the original interpolation function. The interpolation for Jacobian in the conventional ECM required considerable computational cost since it occurs in every spatial variables at each time step. As a simple model to show this technique effectively, we apply the proposed method to the one-dimensional and the two-dimensional Burgers equations. Through several numerical simulations, it is shown that the proposed method has second temporal and third spatial convergence orders. Further, we discuss the energy and the mass conservation properties and numerically show a good performance on these properties.

The remainder of this paper is organized as follows. In Sect. 2, the BSLM based on the FDM and the BDF2 is reviewed for the Burgers equations. In Sect. 3, we introduce a modified error correction technique to solve the highly nonlinear IVP. Three test problems are numerically solved in Sect. 4 in order to demonstrate the accuracy and the efficiency of the proposed method. Finally, conclusions are given in Sect. 5.

2 Backward semi-Lagrangian method

This section aims to give brief descriptions for the BSLM based on both the FDM and the BDF2 for the one-dimensional case and the system of Burgers equations.

2.1 One-dimensional Burgers equation

As a model problem, we consider the one-dimensional Burgers equation described by

$$u_t + uu_x = \nu u_{xx}, \quad x_L < x < x_R, 0 < t \leq T, \quad (1)$$

with the boundary and the initial conditions given by

$$\begin{cases} u(t, x_L) = g_1(t), & u(t, x_R) = g_2(t), & t > 0, \\ u(0, x) = u^0(x), & x \in [x_L, x_R], \end{cases}$$

where $\nu > 0$ is the coefficient of kinematic viscosity. Here, $u^0(x)$ and $g_k(t)$ ($k = 1, 2$) are assumed to be sufficiently smooth functions for the existence and the uniqueness of the solution [21, 22]. The solution u may represent a temperature for heat transfer or a species concentration for mass transfer at position x and time t with the advection velocity u .

From the Lagrangian view, consider the characteristic curve $\pi(x_i, t_{n+1}; t)$ satisfying the following IVP with an initial value $\pi(x, s; s) = x$:

$$\frac{d\pi(x, s; t)}{dt} = u(t, \pi(x, s; t)), \quad t \leq s, \quad (2)$$

where $u(t, \cdot)$ is the solution of (1). Then the material derivative $\frac{D}{Dt}u(t, \pi(x, s; t))$ satisfies

$$\frac{D}{Dt}u(t, \pi(x, s; t)) = \nu u_{xx}(t, \pi(x, s; t)), \tag{3}$$

which is valid along the characteristic curve $\pi(x, s; t)$. Solving these two equations (2) and (3) simultaneously, the solution u of (1) can be obtained.

For further discussion, we introduce uniform discretizations of the temporal and the spatial domains as follows:

$$t_n = nh \quad (0 \leq n \leq N), \quad x_i = x_L + i\Delta x \quad (0 \leq i \leq M_x), \tag{4}$$

where $h := T/N$ and $\Delta x := (x_R - x_L)/M_x$ are the temporal step and the spatial grid sizes, respectively. Additionally, to approximate the first and the second-order partial derivatives in the considered backward semi-Lagrangian algorithm, we introduce the fourth-order finite difference weight matrices of size $\widehat{M}_x := M_x + 1$ (see [20]) as follows:

$$\mathcal{W}_x^1 = \frac{1}{12\Delta x} \begin{bmatrix} -25 & 48 & -36 & 16 & -3 & \cdots & 0 \\ -3 & -10 & 18 & -6 & 1 & & \\ 1 & -8 & 0 & 8 & -1 & & \\ & \ddots & \ddots & \ddots & \ddots & & \\ & & 1 & -8 & 0 & 8 & -1 \\ & & -1 & 6 & -18 & 10 & 3 \\ 0 & \cdots & 3 & -16 & 36 & -48 & 25 \end{bmatrix},$$

$$\mathcal{W}_x^2 = \frac{1}{12\Delta x^2} \begin{bmatrix} 45 & -154 & 214 & -156 & 61 & -10 & \cdots & 0 \\ 10 & -15 & -4 & 14 & -6 & 1 & & \\ -1 & 16 & -30 & 16 & -1 & & & \\ & -1 & 16 & -30 & 16 & -1 & & \\ & & \ddots & \ddots & \ddots & \ddots & \ddots & \\ & & & -1 & 16 & -30 & 16 & -1 \\ & & & 1 & -6 & 14 & -4 & -15 & 10 \\ 0 & \cdots & -10 & 61 & -156 & 214 & -154 & 45 \end{bmatrix}.$$

These sparse matrices \mathcal{W}_x^1 and \mathcal{W}_x^2 can be obtained by using a five point finite difference scheme [20] at the interior points and one-side method for points near the boundary. Using these matrices, the partial derivatives of a smooth function f can be expressed in matrix form as follows:

$$\frac{\partial^m}{\partial x^m}f(x_i) = (\mathcal{W}_x^m \mathbf{f})_i + \mathcal{O}(\Delta x^4), \quad m = 1, 2, \tag{5}$$

where $\mathbf{f} := (f_i)_{\widehat{M}_x \times 1}$ for $f_i := f(x_i)$ and $(\mathbf{a})_i$ denotes the i th component of a vector \mathbf{a} . In the proposed scheme, \mathcal{W}_x^1 and \mathcal{W}_x^2 are used for the interpolation scheme and the diffusion term, respectively.

The BSLM we are concerned with is combined with both the BDF2 and the FDM to find an approximation of the solution u in each grid point x_i at $t = t_{n+1}$ as follows. Firstly,

we denote $u^n(x_i) := u(t_n, x_i)$ and its approximation by U_i^n for each x_i at time $t = t_n$. In addition, we introduce the notations $\mathbf{U}^n := [U_0^n, U_1^n, \dots, U_{\overline{M}_x}^n]^T$ and $\tilde{\mathbf{U}}^n := [U_1^n, U_2^n, \dots, U_{\overline{M}_x}^n]^T$ for $\overline{M}_x := M_x - 1$ and the superscript \mathcal{T} denotes the transpose. Here, we assume that the approximations U_i^k ($k \leq n$) have been previously calculated at all grid points. After evaluating (3) at point $(t, x) = (t_{n+1}, x_i)$ and applying the BDF2 for the material derivative in the left hand side of (3), we get the following formula:

$$u^{n+1}(x_i) - \tilde{\mu} u_{xx}^{n+1}(x_i) = \frac{4}{3} u^n(\pi(x_i, t_{n+1}; t_n)) - \frac{1}{3} u^{n-1}(\pi(x_i, t_{n+1}; t_{n-1})) + \mathcal{O}(h^3), \tag{6}$$

for each i ($i = 1, \dots, \overline{M}_x$) with $\tilde{\mu} = \frac{2hv}{3}$.

By applying the finite difference formula (5) with $m = 2$ to the diffusion term of (6), we get the semi-discretization system in matrix form as follows:

$$\begin{aligned} \mathcal{A} \tilde{\mathbf{U}}^{n+1} &= \mathbf{d}^{n+1} + \mu \mathbf{b}^{n+1}, \\ \mathcal{A} &:= (\mathcal{I} - \tilde{\mu} \tilde{\mathcal{W}}_x^2), \quad \mathbf{d}^{n+1} := [d_1^{n+1}, d_2^{n+1}, \dots, d_{\overline{M}_x}^{n+1}]^T, \\ \mathbf{b}^{n+1} &:= [10g_1(t_{n+1}), -g_1(t_{n+1}), 0, \dots, 0, -g_2(t_{n+1}), 10g_2(t_{n+1})]^T, \\ d_i^{n+1} &= \frac{4}{3} u^n(\pi(x_i, t_{n+1}; t_n)) - \frac{1}{3} u^{n-1}(\pi(x_i, t_{n+1}; t_{n-1})) + \mathcal{O}(h^3 + h\Delta x^4), \end{aligned} \tag{7}$$

where \mathcal{I} denotes an identity matrix of size \overline{M}_x and $\tilde{\mathcal{W}}_x^2$ is the matrix constructed from \mathcal{W}_x^2 by taking the interior elements $(i, j) = (1 : \overline{M}_x, 1 : \overline{M}_x)$.

For the full-discretization of (7), let $\pi_i^{n+1,k}$ be an approximation of the departure position $\pi(x_i, t_{n+1}; t_k)$ of grid point (t_{n+1}, x_i) at each time step t_k , which are discussed in detail in Sect. 3. Because these departure points generally do not coincide with any grid points and we only know the approximate values on the grid points at $t \leq t_n$, a proper interpolation method is needed. In the proposed scheme, the Hermite cubic interpolation I_H [23] is adopted among various interpolation schemes.

Now, after the use of the interpolation scheme, taking Taylor expansions of $u^n(\cdot)$ and $u^{n-1}(\cdot)$ about $\pi_i^{n+1,k}$ ($k = n, n - 1$) and dropping truncation errors, we can obtain a full-discretization system for the semi-discretization system (7) by the following approximation of d_i^{n+1} :

$$d_i^{n+1} \approx \frac{4}{3} I_H \mathbf{U}^n(\pi_i^{n+1,n}) - \frac{1}{3} I_H \mathbf{U}^{n-1}(\pi_i^{n+1,n-1}),$$

where $I_H \mathbf{U}^k$ denotes the Hermite cubic interpolation with the approximate vector \mathbf{U}^k . Notice that, since $\tilde{\mu} > 0$ and \mathcal{A} is non-singular, the solution of the full-discretization system can easily be obtained by using the LU factorization.

2.2 System of two-dimensional Burgers equations

In this subsection, we extend the BSLM to solve a system of two-dimensional Burgers equations given by

$$\begin{cases} u_t + uu_x + vu_y = v(u_{xx} + u_{yy}), \\ v_t + uv_x + vv_y = v(v_{xx} + v_{yy}), \end{cases} \tag{8}$$

with the initial and the boundary conditions

$$\begin{cases} u(0, x, y) = u^0(x, y), & v(0, x, y) = v^0(x, y), & (x, y) \in \mathcal{D} = [x_L, x_R] \times [y_L, y_R], \\ u(t, x, y) = g_1(t, x, y), & v(t, x, y) = g_2(t, x, y), & t \in [0, T], (x, y) \in \partial\mathcal{D}, \end{cases}$$

where ν is a positive constant and u^0, v^0 and g_i are given smooth functions in the physical domain \mathcal{D} . For further discussion, together with (4) we additionally introduce the discretization of the spatial domain in the y -direction as follows:

$$y_j = y_L + j\Delta y, \quad 0 \leq j \leq M_y, \tag{9}$$

where $\Delta y := (y_R - y_L)/M_y$ is the spatial grid size in the y -direction. Also, let $u^k(\mathbf{x}_{i,j}) := u(t_k, \mathbf{x}_{i,j})$ with $\mathbf{x}_{i,j} = [x_i, y_j]^T$ and $U_{i,j}^k$ be its approximate value. In addition, we introduce some notations $\mathbf{U}^k := [U_{0,0}^k, U_{1,0}^k, \dots, U_{M_x,0}^k, U_{0,1}^k, U_{1,1}^k, \dots, U_{M_x,M_y}^k]^T$ and $\tilde{\mathbf{U}}^k := [U_{1,1}^k, U_{2,1}^k, \dots, U_{\bar{M}_x,1}^k, U_{1,2}^k, U_{2,2}^k, \dots, U_{\bar{M}_x,\bar{M}_y}^k]^T$ for $\bar{M}_x := M_x - 1$ and $\bar{M}_y := M_y - 1$, and the superscript \mathcal{T} denotes the transpose.

From the Lagrangian view, problem (8) is equivalent as

$$\frac{D}{Dt} \mathbf{u}(t, \boldsymbol{\pi}(\mathbf{x}, t_{n+1}; t)) = \nu \Delta \mathbf{u}(t, \boldsymbol{\pi}(\mathbf{x}, t_{n+1}; t)), \quad t \in [0, T], \tag{10}$$

along the characteristic curve $\boldsymbol{\pi}(\mathbf{x}, t_{n+1}; t) := [\pi_1(x, t_{n+1}; t), \pi_2(y, t_{n+1}; t)]^T$ satisfying the following IVP:

$$\begin{cases} \frac{d\boldsymbol{\pi}(\mathbf{x}, t_{n+1}; t)}{dt} = \mathbf{u}(t, \boldsymbol{\pi}(\mathbf{x}, t_{n+1}; t)), & t \leq t_{n+1}, \\ \boldsymbol{\pi}(\mathbf{x}, t_{n+1}; t_{n+1}) = \mathbf{x}, \end{cases} \tag{11}$$

where $\mathbf{u}(t, \boldsymbol{\pi}(\mathbf{x}, t_{n+1}; t)) = [u(t, \boldsymbol{\pi}(\mathbf{x}, t_{n+1}; t)), v(t, \boldsymbol{\pi}(\mathbf{x}, t_{n+1}; t))]^T$.

As described in Sect. 2.1, after evaluating (10) at time $t = t_{n+1}$ and grid point $\mathbf{x} = \mathbf{x}_{i,j}$, the BDF2 and the finite difference formula (5) are applied to approximate the material derivative and diffusion term, respectively. Then, by a similar process to the derivation of equations (6) and (7), we obtain the full-discretization systems for (10) given by

$$\begin{aligned} \mathcal{A}\tilde{\mathbf{U}}^{n+1} &= \mathbf{d}^{n+1} + \mu_1 \mathbf{b}^{n+1}, & \mathcal{A}\tilde{\mathbf{V}}^{n+1} &= \hat{\mathbf{d}}^{n+1} + \mu_2 \hat{\mathbf{b}}^{n+1}, \\ \mathcal{A} &= \mathcal{I}_y \otimes \mathcal{I}_x - \tilde{\mu}(\mathcal{I}_y \otimes \tilde{\mathcal{W}}_x^2) - \tilde{\mu}(\tilde{\mathcal{W}}_y^2 \otimes \mathcal{I}_x), \end{aligned} \tag{12}$$

where $\mathbf{d}^{n+1} := [d_{1,1}^{n+1}, d_{2,1}^{n+1}, \dots, d_{\bar{M}_x,1}^{n+1}, d_{1,2}^{n+1}, d_{2,2}^{n+1}, \dots, d_{\bar{M}_x-1,\bar{M}_y}^{n+1}, d_{\bar{M}_x,\bar{M}_y}^{n+1}]^T$ and $\hat{\mathbf{d}}^{n+1} := [\hat{d}_{1,1}^{n+1}, \hat{d}_{2,1}^{n+1}, \dots, \hat{d}_{\bar{M}_x,1}^{n+1}, \hat{d}_{1,2}^{n+1}, \hat{d}_{2,2}^{n+1}, \dots, \hat{d}_{\bar{M}_x-1,\bar{M}_y}^{n+1}, \hat{d}_{\bar{M}_x,\bar{M}_y}^{n+1}]^T$ are $\bar{M}_x \bar{M}_y \times 1$ vectors whose entries are given by

$$\begin{aligned} d_{i,j}^{n+1} &\approx \frac{4}{3} I_H \mathbf{U}^n(\boldsymbol{\pi}_{i,j}^{n+1,n}) - \frac{1}{3} I_H \mathbf{U}^{n-1}(\boldsymbol{\pi}_{i,j}^{n+1,n-1}), \\ \hat{d}_{i,j}^{n+1} &\approx \frac{4}{3} I_H \mathbf{V}^n(\boldsymbol{\pi}_{i,j}^{n+1,n}) - \frac{1}{3} I_H \mathbf{V}^{n-1}(\boldsymbol{\pi}_{i,j}^{n+1,n-1}). \end{aligned}$$

Here, $\tilde{\mu} := \frac{2\nu h}{3}$, and \mathcal{I}_x and \mathcal{I}_y are identity matrices of sizes \bar{M}_x and \bar{M}_y , respectively. Additionally, the matrix $\tilde{\mathcal{W}}_y^2$ can be defined similarly from the definition of $\tilde{\mathcal{W}}_x^2$. The notation

\otimes is for the tensor product, and \mathbf{b}^{n+1} and $\hat{\mathbf{b}}^{n+1}$ are vectors induced from the boundary conditions, which are obtained by a similar process to that described in Sect. 2.1.

To solve the system (12) effectively, we employ the following eigenvalue decomposition:

$$\tilde{\mathcal{W}}_k^2 = Q_k \Lambda_k Q_k^{-1} \quad (k = x, y), \tag{13}$$

where Q_k and Λ_k are the matrices of eigenpair for the matrix $\tilde{\mathcal{W}}_k^2$. Then, using the property of the tensor product and (13), the coefficient matrix \mathcal{A} defined by (12) can be decomposed as

$$\mathcal{A} = (Q_y \otimes Q_x) \Sigma (Q_y \otimes Q_x)^{-1},$$

where $\Sigma := (\mathcal{I}_y \otimes \mathcal{I}_x) - \tilde{\mu}(\mathcal{I}_y \otimes \Lambda_x) - \tilde{\mu}(\Lambda_y \otimes \mathcal{I}_x)$. Hence, the system (12) can be solved effectively by three linear systems for u as follows:

$$\check{\mathbf{d}}^{n+1} = (Q_y \otimes Q_x)^{-1}(\mathbf{d}^{n+1} + \mu_1 \mathbf{b}^{n+1}), \quad \hat{\mathbf{U}}^{n+1} = \Sigma^{-1} \check{\mathbf{d}}^{n+1}, \quad \tilde{\mathbf{U}}^{n+1} = (Q_y \otimes Q_x) \hat{\mathbf{U}}^{n+1}.$$

In a similar way, three linear systems can be obtained for v .

3 Modified error correction method

The goal of this section is to develop a modified ECM for the approximate values $\pi_i^{n+1,k}$ ($k = n, n - 1$) of the departure points, which maintains the advantage of the original ECM introduced in [17, 18] with less computational costs. For the two-dimensional case, $\pi_{ij}^{n+1,k}$ ($k = n, n - 1$) can be obtained by extension of the following process. Let $\pi_i(t) := \pi(x_i, t_{n+1}; t)$ be the solution of the nonlinear IVP with an initial value $\pi_i(t_{n+1}) = x_i$ given by

$$\frac{d\pi_i(t)}{dt} = u(t, \pi_i(t)), \quad t \in [t_{n-1}, t_{n+1}], \tag{14}$$

where u is the solution of the Burgers equation (1) for an arbitrary grid point x_i .

We begin by introducing modified Euler polygons:

$$y_i(t) := \min\{\max\{\hat{y}_i(t), x_L\}, x_R\} \quad (\text{Dirichlet boundary case}),$$

$$y_i(t) := \begin{cases} \hat{y}_i(t) & \hat{y}_i(t) \in [x_L, x_R], \\ x_R - (x_L - \hat{y}_i(t)) & \hat{y}_i(t) < x_L, \\ x_L + (\hat{y}_i(t) - x_R) & \hat{y}_i(t) > x_R, \end{cases} \quad (\text{periodic boundary case}), \tag{15}$$

where $\hat{y}_i(t) := x_i + (t - t_{n+1})u(t_n, x_i)$. Note that the above modified Euler polygons are constructed so that the position of the particle remains in the physical domain. Hereafter, we regard $y_k(t)$ as $\hat{y}_k(t)$ for simplicity of the discussion.

Let $\pi_i(t)$ be the solution to the nonlinear IVP (14) at the fixed grid point x_i , and assume that it is sufficiently smooth with respect to both t and x_i . Then the Taylor expansions of $\pi_i(t)$ at t_{n+1} and $u(t_{n+1}, x_i)$ at t_n give

$$\begin{aligned} \pi_i(t) &= \pi_i(t_{n+1}) + (t - t_{n+1})u(t_{n+1}, \pi(t_{n+1})) + \mathcal{O}(h^2) \\ &= x_i + (t - t_{n+1})u(t_n, x_i) + \mathcal{O}(h^2). \end{aligned} \tag{16}$$

Moreover, an error between of the $\pi_i(t)$ and $y_i(t)$ is defined

$$\psi_i(t) := \pi_i(t) - y_i(t), \quad i = 1, \dots, \overline{M}_x \tag{17}$$

with asymptotic behavior of $\psi_i(t) = \mathcal{O}(h^2)$, which is obtained by combining (16) and (17). After differentiating (17), combining the result with (14) and the Euler polygons leads to the following formula:

$$\psi'_i(t) = \pi'_i(t) - y'_i(t) = u(t, \pi_i(t)) - u(t_n, x_i) = u(t, y_i(t) + \psi_i(t)) - u(t_n, x_i).$$

By taking a Taylor expansion $\pi_i(t)$ about $y_i(t)$, we can see the asymptotic linear IVP for $\psi_i(t)$ given by

$$\psi'_i(t) = u(t, y_i(t)) + u_x(t, y_i(t))\psi_i(t) - u(t_n, x_i) + \mathcal{O}(\psi(t)^2), \quad t \leq t_{n+1} \quad , \tag{18}$$

with the initial value $\psi_i(t_{n+1}) = 0$. Then, instead of solving the nonlinear problem (14), the proposed method solves the linear IVP (18).

To reduce the computational cost, the proposed method approximates the Jacobian value $u_x(t, y_i(t))$ as a fixed one $u_x(t_n, x_i)$. By Taylor's expansion the Jacobian $u_x(t, y_i(t))$ about $t = t_n$, (18) can be changed with the asymptotic behavior of $\psi_i(t)$ as follows:

$$\psi'_i(t) = u_x(t_n, x_i)\psi_i(t) + u(t, y_i(t)) - u(t_n, x_i) + \mathcal{O}(h^3), \quad t \leq t_{n+1}. \tag{19}$$

Note that the fixed value does not affect the scale of the second-order asymptotic error. On the other hand, the conventional ECM once again performs the interpolation with the derivatives of the original interpolation function for every spatial grid in each time integration.

To solve the asymptotic IVP (19), after integrating both sides of (19) over $[t_{n-1}, t_{n+1}]$, we use the midpoint integration rule and $\psi_i(t_n) = \frac{1}{2}(\psi_i(t_{n+1}) + \psi_i(t_{n-1})) + \mathcal{O}(h^2)$. Then we obtain

$$(1 + hu_x(t_n, x_i))\psi_i(t_{n-1}) = 2h(u(t_n, y_i(t_n)) - u(t_n, x_i)) + \mathcal{O}(h^3). \tag{20}$$

Recall that we only know the approximations U_i^k ($k \leq n$) at the grid points. Thus, to evaluate the solution at the non-grid point $y_i(t_n)$, we apply the Hermite cubic interpolation. Then, instead of solving (20), we solve the following equation:

$$\psi_i^{n-1} := 2h \frac{U_i^n - I_H \mathbf{U}^n(y_i(t_n))}{1 + h\mathcal{J}_i}, \quad 0 \leq i \leq \overline{M}_x,$$

to obtain an approximation $\pi_i^{n+1, n-1}$ defined by

$$\pi_i^{n+1, n-1} := y_i(t_{n-1}) + \psi_i^{n-1}, \tag{21}$$

where $\mathcal{J}_i := (\tilde{\mathcal{N}}_x^1 \tilde{\mathbf{U}}^n)_i$ for $\tilde{\mathcal{W}}_x^1$ constructed by interior elements of \mathcal{W}_x^1 .

To find an approximate value $\pi_i^{n+1,n}$, after using the Taylor expansion of $\pi_i(t)$ about t_{n-1} together with the initial condition $\pi_i(t_{n+1}) = x_i$, evaluating at time t_n leads to

$$\pi_i(t_n) = \frac{1}{4}(x_i + 3\pi_i(t_{n-1}) + 2hu(t_{n-1}, \pi_i(t_{n-1}))) + \mathcal{O}(h^3). \tag{22}$$

By using $\pi_i^{n+1,n-1}$, $I_H \mathbf{U}^{n-1}(\pi_i^{n+1,n-1})$ and (22), we can define the approximation of $\pi_i(t_n)$ as

$$\pi_i^{n+1,n} := \frac{1}{4}(x_i + 3\pi_i^{n+1,n-1} + 2hI_H \mathbf{U}^{n-1}(\pi_i^{n+1,n-1})).$$

Remark The method discussed above for finding $\pi_i^{n+1,k}$ ($k = n, n - 1$) can be naturally extended to the case of system for the characteristic curve.

4 Numerical experiments

In this section, we carry out numerical simulations to illustrate the accuracy and the efficiency of the proposed method. To measure the computational error of the proposed scheme, we use the maximum norm error $err_\infty(t)$ and the relative L_2 norm error $err_{R2}(t)$, which are, respectively, defined by

$$err_\infty(t_k) = \max_i |u^k(x_i) - U_i^k|,$$

$$err_{R2}(t_k) = \left(\sum_i |u^k(x_i) - U_i^k|^2 \right)^{1/2} / \left(\sum_i |u^k(x_i)|^2 \right)^{1/2},$$

where $u^k(x_i)$ is the exact solution and U_i^k is its approximation at time $t = t_k$ and each grid point x_i . For the two-dimensional case, we similarly define both the maximum norm error and the relative L_2 norm error. All numerical simulations are executed with MATLAB 2013a (8.1.0.604) using Windows 10 OS.

Example 1 Consider the one-dimensional Burgers equation (1) on $[0, 1]$ with the shock initial condition [24] as follows:

$$u(0, x) = \begin{cases} 9x, & 0 \leq x \leq \frac{1}{10}, \\ 1 - x, & \frac{1}{10} \leq x \leq 1, \end{cases}$$

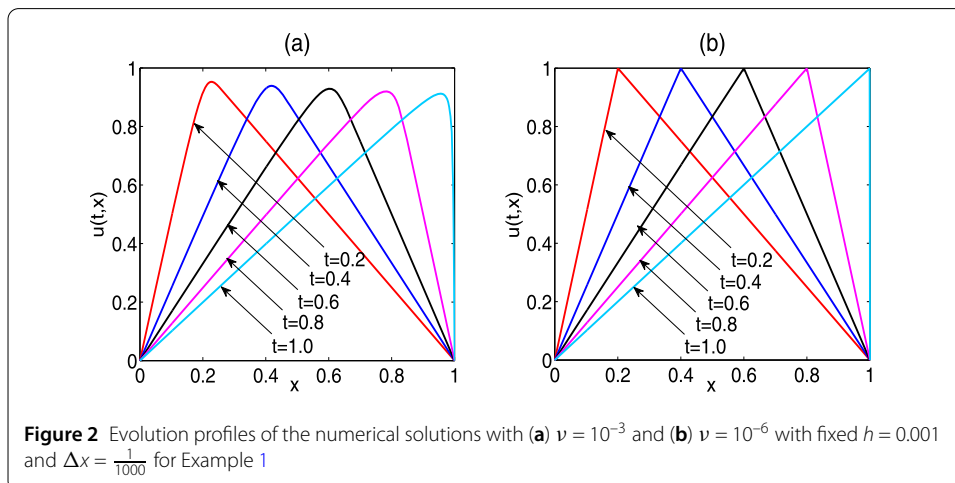
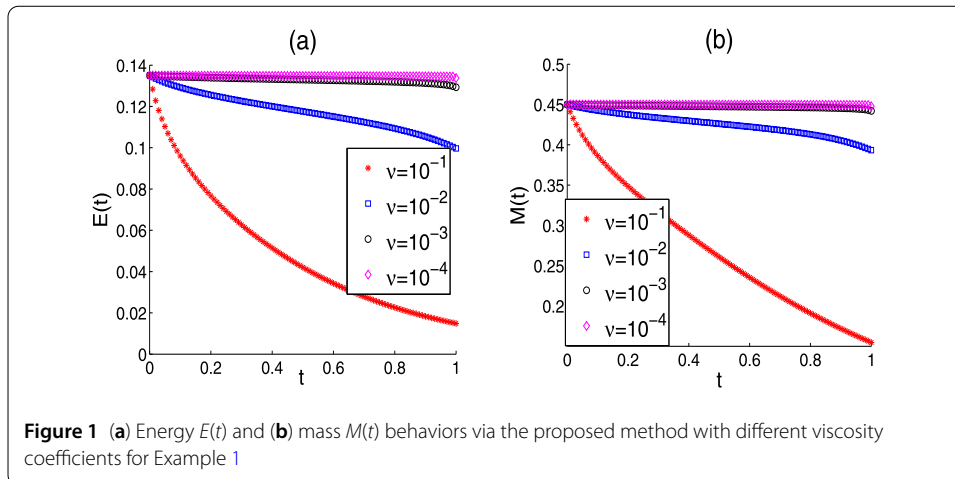
and the homogeneous boundary condition or the periodic boundary condition. For the discussion of the energy and the mass conservation properties, we measure the energy and the mass by the energy function $E(t)$ and the mass function $M(t)$ defined as

$$\frac{d}{dt}E(t) = -v \int_0^1 u_x^2(t, x) dx, \quad \frac{d}{dt}M(t) = v(u_x(t, 1) - u_x(t, 0)),$$

$$E(t) = \frac{1}{2} \int_0^1 u^2(t, x) dx, \quad M(t) = \int_0^1 u(t, x) dx. \tag{23}$$

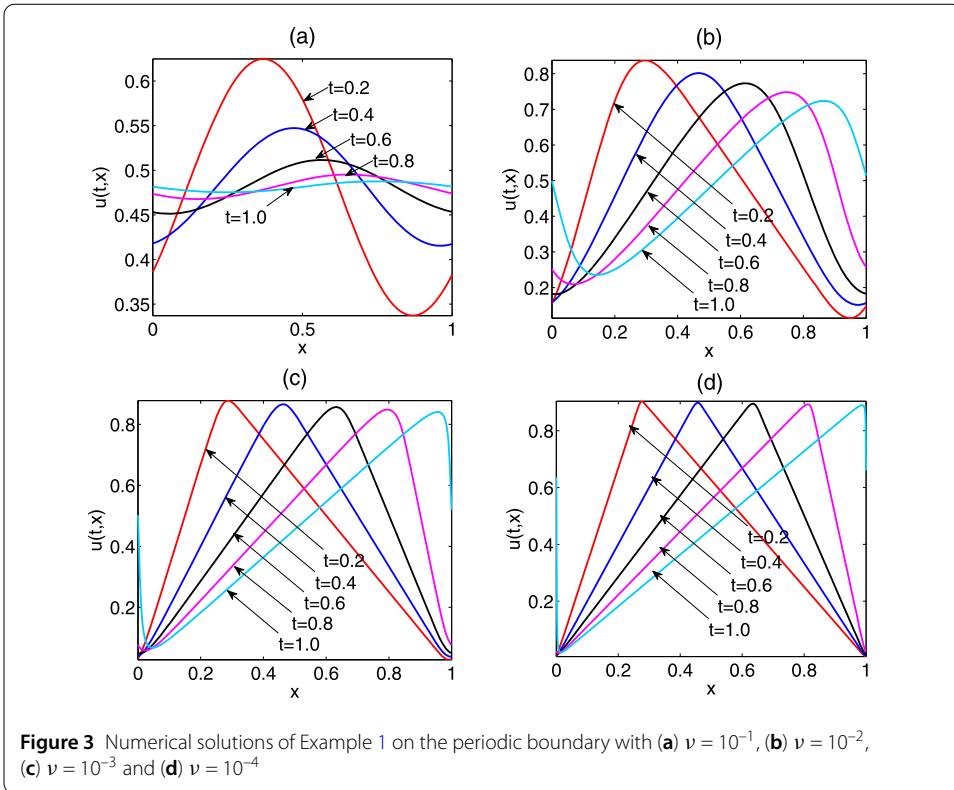
Both $E(t)$ and $M(t)$ are approximated with the composed trapezoidal rule defined by

$$E(t_k) \approx \frac{\Delta x}{2} \sum_{i=1}^{\overline{M_x}} (U_i^k)^2, \quad M(t_k) \approx \Delta x \sum_{i=1}^{\overline{M_x}} (U_i^k).$$



We solve the problem using fixed time and grid sizes of $h = 0.01$ and $\Delta x = \frac{1}{400}$, respectively, by varying of the viscosity coefficients $\nu = 10^{-k}$ ($k = 1, 2, 3, 4$) with homogeneous boundary condition. Using the obtained numerical solutions, we calculate the mentioned energy $E(t)$ and mass $M(t)$ at discrete time levels and display the results in Fig. 1. From the formula of (23), $E(t)$ and $M(t)$ must tend to constants when the viscosity coefficient ν theoretically goes to zero. As expected, the numerical results clearly support that the proposed method has a good performance for both the energy and the mass conservation properties when the viscosity coefficient goes to zero. We additionally observe the behaviors of the numerical solutions over time for relatively small viscosity coefficients $\nu = 10^{-k}$ ($k = 3, 6$) with homogeneous boundary condition and the results are plotted in Fig. 2. In Fig. 2, the parameters used in (a)–(b) are $h = 0.01$, $\Delta x = \frac{1}{200}$ and (c)–(d) are $h = 0.002$, $\Delta x = \frac{1}{1000}$. It can be seen that the proposed method shows good performance even for the advection dominated case $\nu = 10^{-6}$ without unnecessary oscillation.

Finally, we also observed the behavior of the numerical solutions for various viscosity coefficients $\nu = 10^{-k}$ ($k = 0, 1, 2, 3$) on the periodic boundary condition to observe how they behave with mild to smaller viscosity coefficients. The wave propagation outline of $u(t, x)$ is illustrated in Fig. 3. It is clear from Fig. 3(a) and (b) that there is no sharp front in the solutions for $\nu = 10^{-1}$ and $\nu = 10^{-2}$, which becomes sharper for $\nu = 10^{-3}$ and $\nu =$



10^{-4} . In other words, as ν becomes smaller, the curves steepen and develop a shock-like discontinuity. As seen in Fig. 3(c) and (d), the proposed method is able to capture the sharp front very well under the periodic boundary conditions. Note that this test is valuable, since the Burgers equation has been studied in a limited work over the periodic boundary conditions compared to non-periodic boundary conditions.

Example 2 We consider the two-dimensional unsteady Burgers equation,

$$u_t + uu_x + uu_y = \nu(u_{xx} + u_{yy})$$

on $(x, y) \in [0, 1]^2$ with the Dirichlet boundary condition whose analytic solution [4] is given by

$$u(t, x, y) = \frac{1}{1 + \exp\left(\frac{x+y-t}{2\nu}\right)}.$$

We first examine the temporal convergence rate for the proposed method for the two viscosity coefficients $\nu = 0.1$ and $\nu = 0.01$. Results are measured by both the maximum norm and the relative L_2 norm errors with a fixed spatial grid size $\Delta x = \Delta y = 1/2000$ and varying time step size h at two different times $t = 0.1$ and $t = 1.0$. The numerical results are listed in Table 1, and they show that the proposed method has second-order temporal convergence. Also, to assess the spatial convergence rate for the proposed method, err_∞ and err_{R2} are calculated with a sufficiently small temporal step size $h = 0.00002$ by the variation of spatial resolutions $M_x = M_y$ from 20 to 640 at two different time points $t = 0.1$ and $t = 1.0$.

Table 1 Temporal convergence rate for Example 2

h	$t = 0.1$				$t = 1.0$			
	$err_{\infty}(t)$	Rates	$err_{R2}(t)$	Rates	$err_{\infty}(t)$	Rates	$err_{R2}(t)$	Rates
$\nu = 0.1, \Delta x = \Delta y = \frac{1}{2000}$								
$\frac{1}{50}$	3.23×10^{-4}	–	2.37×10^{-4}	–	5.56×10^{-4}	–	5.01×10^{-4}	–
$\frac{1}{100}$	9.00×10^{-5}	1.84	6.61×10^{-5}	1.84	1.44×10^{-4}	1.94	1.32×10^{-4}	1.92
$\frac{1}{200}$	2.38×10^{-5}	1.92	1.74×10^{-5}	1.92	3.68×10^{-5}	1.97	3.39×10^{-5}	1.96
$\frac{1}{400}$	6.28×10^{-6}	1.92	4.39×10^{-6}	1.99	9.29×10^{-6}	1.99	8.57×10^{-6}	1.98
$\frac{1}{800}$	1.12×10^{-6}	2.49	1.28×10^{-6}	1.78	2.21×10^{-6}	2.07	2.01×10^{-6}	2.09
$\nu = 0.01, \Delta x = \Delta y = \frac{1}{2000}$								
$\frac{1}{50}$	5.91×10^{-2}	–	3.29×10^{-2}	–	3.37×10^{-1}	–	7.08×10^{-2}	–
$\frac{1}{100}$	1.17×10^{-2}	2.34	8.81×10^{-2}	1.90	2.12×10^{-2}	3.99	5.18×10^{-3}	3.77
$\frac{1}{200}$	3.16×10^{-3}	1.89	2.68×10^{-3}	1.72	3.81×10^{-3}	2.48	1.08×10^{-3}	2.26
$\frac{1}{400}$	8.54×10^{-4}	1.89	7.59×10^{-4}	1.82	9.46×10^{-4}	2.01	2.84×10^{-4}	1.93
$\frac{1}{800}$	2.12×10^{-4}	2.01	1.90×10^{-4}	2.00	2.34×10^{-5}	2.02	7.02×10^{-5}	2.01

Table 2 Spatial convergence rate for Example 2 with fixed $h = 0.00002$

$M_x = M_y$	$t = 0.1$				$t = 1.0$			
	$err_{\infty}(t)$	Rates	$err_{R2}(t)$	Rates	$err_{\infty}(t)$	Rates	$err_{R2}(t)$	Rates
$\nu = 0.1$								
20	7.45×10^{-6}	–	7.81×10^{-6}	–	2.11×10^{-5}	–	1.30×10^{-5}	–
40	1.36×10^{-7}	5.78	3.28×10^{-7}	4.58	1.31×10^{-6}	4.02	8.67×10^{-7}	3.90
80	1.33×10^{-8}	3.36	2.47×10^{-8}	3.73	8.16×10^{-8}	4.00	5.51×10^{-8}	3.98
160	9.59×10^{-10}	3.79	1.64×10^{-9}	3.91	5.24×10^{-9}	3.96	3.58×10^{-9}	3.94
320	9.00×10^{-11}	3.41	1.67×10^{-10}	3.30	5.05×10^{-10}	3.38	3.60×10^{-10}	3.31
$\nu = 0.01$								
40	4.04×10^{-2}	–	2.16×10^{-2}	–	2.63×10^{-2}	–	3.13×10^{-3}	–
80	5.90×10^{-4}	6.10	3.84×10^{-4}	5.81	8.02×10^{-4}	5.03	2.07×10^{-4}	3.92
160	5.06×10^{-5}	3.54	3.00×10^{-5}	3.68	5.51×10^{-5}	3.86	1.35×10^{-5}	3.94
320	3.18×10^{-6}	3.99	2.08×10^{-6}	3.85	3.39×10^{-6}	4.02	8.58×10^{-7}	3.98
640	1.98×10^{-7}	4.01	1.33×10^{-7}	3.96	2.28×10^{-7}	3.90	5.99×10^{-8}	3.84

Table 3 Comparison of numerical errors obtained from the proposed method and the CSCM for Example 2

ν	CSCM				Proposed			
	h	$M = N$	$err_{R2}(0.05)$	CPU	h	$M_x = M_y$	$err_{R2}(0.05)$	CPU
1.0	0.005	6	3.79×10^{-9}	0.0361	0.005	10	1.18×10^{-9}	0.0061
	0.0005	11	3.02×10^{-10}	0.0370	0.0005	20	8.48×10^{-11}	0.0189
0.1	0.005	6	2.15×10^{-4}	0.0317	0.005	20	6.20×10^{-6}	0.0074
	0.001	11	1.08×10^{-6}	0.0343	0.001	40	1.94×10^{-7}	0.0320

The results are displayed in Table 2. It is seen that the rate of spatial convergence is greater than or equal to 3.

To explore the efficiency of the proposed method, the method is compared with the Chebyshev spectral collocation method (CSCM) combined with the fourth-order Runge–Kutta time integration scheme developed by [4]. For the CSCM, we use the numbers of Chebyshev–Gauss–Lobatto points N and M shown in Table 3. We calculate err_{R2} at time $t = 0.05$ and estimate the computational time cost (CPU) for each method. The numerical results are displayed in Table 3. It can be seen that the proposed method is superior to the CSCM in terms of CPU for the two cases $\nu = 1.0$ and $\nu = 0.1$. In particular, for $\nu = 0.1$, the

result shows that the proposed method is superior to the CSCM in terms of both CPU and accuracy.

Example 3 Consider the system of the two-dimensional Burgers equations (8) on $(x, y) \in [0, 1]^2$ with the Dirichlet boundary condition whose analytic solution [4] is given by

$$u(t, x, y) = v(t, x, y) = \frac{3}{4} - \frac{1}{4(1 + \exp(\frac{4y-4x-t}{32\nu}))}.$$

We compute the relative L_2 norm error, $err_{R2}(t)$, at two times $t = 0.5$ and $t = 2.0$ with the viscosity coefficient $\nu = 0.01$ and fixed spatial grid size $\Delta x = \Delta y = \frac{1}{160}$. Additionally, we investigate the temporal convergence rate by the variation of time step sizes from $\frac{1}{80}$ to $\frac{1}{2560}$. The numerical results are listed in Table 4, and the second-order convergence is numerically obtained for the two velocities u and v . To explore the efficiency of the present method, we compare the proposed method with the CSCM. The relative L_2 norm error for u is calculated at time $t = 0.01$ with various viscosity coefficients $\nu = 1.0, 0.1, 0.01$, and 0.005 . The numerical results are displayed in Table 5. One can see that our method has a good performance with a significantly reduced computational cost when compared to that of the CSCM. Overall, it could be noted that, with an increased size of the system, there is an increased efficiency in the proposed method when compared to that of the CSCM. It must be noted that the result for v has a similar aspect to that of u .

To compare the proposed method and the original ECM in terms of computational costs, we measured the relative L_2 norm error and CPU with fixed time and grid sizes $h = 0.005$ and $\Delta x = 1/500$ at the different time levels $t = 1.0, 2.0$ and 3.0 , and the results are listed in Table 6. Table 6 shows that the proposed method requires less CPU without any loss in accuracy when compared with the original ECM. Because similar results are obtained for u from the symmetric property of u and v , the results of u are omitted.

Table 4 Temporal convergence rates for Example 3, with $\Delta x = \Delta y = \frac{1}{160}$ and $\nu = 0.01$

h	$t = 0.5$				$t = 2.0$			
	u		v		u		v	
	$err_{R2}(t)$	Rates	$err_{R2}(t)$	Rates	$err_{R2}(t)$	Rates	$err_{R2}(t)$	Rates
$\frac{1}{80}$	2.50×10^{-4}	–	1.67×10^{-4}	–	4.28×10^{-4}	–	2.38×10^{-4}	–
$\frac{1}{160}$	1.75×10^{-4}	0.51	1.17×10^{-4}	0.51	2.38×10^{-4}	0.85	1.32×10^{-4}	0.85
$\frac{1}{320}$	9.89×10^{-7}	7.47	6.61×10^{-7}	7.47	8.14×10^{-7}	8.19	4.53×10^{-7}	8.19
$\frac{1}{640}$	4.28×10^{-7}	1.21	2.86×10^{-7}	1.21	3.52×10^{-7}	1.21	1.96×10^{-7}	1.21
$\frac{1}{1280}$	1.22×10^{-7}	1.81	8.14×10^{-8}	1.81	9.94×10^{-8}	1.82	5.53×10^{-8}	1.82
$\frac{1}{2560}$	2.77×10^{-8}	2.14	1.85×10^{-8}	2.14	2.22×10^{-8}	2.16	1.24×10^{-8}	2.16

Table 5 Comparison of numerical results for u obtained from the proposed method and the CSCM for Example 3

ν	CSCM				Proposed			
	h	$M = N$	$err_{R2}(0.01)$	CPU	h	$M_x = M_y$	$err_{R2}(0.01)$	CPU
1.0	0.0050	11	2.24×10^{-6}	0.0956	0.0050	20	3.13×10^{-15}	0.0063
0.1	0.0025	11	1.87×10^{-7}	0.1036	0.0025	20	1.29×10^{-10}	0.0072
0.01	0.0020	21	7.20×10^{-7}	0.1026	0.0020	60	7.31×10^{-8}	0.0205
0.005	0.0010	31	4.73×10^{-6}	0.1106	0.0010	120	1.14×10^{-7}	0.0924

Table 6 Comparison of errors and CPUs of v via the proposed method and the original ECM for Example 3

t	ν	Original ECM		Proposed	
		$err_{R2}(t)$	CPU	$err_{R2}(t)$	CPU
1.0	10^{-1}	2.21×10^{-6}	37.0937	2.21×10^{-6}	33.9425
	10^{-3}	4.99×10^{-4}	37.1391	4.70×10^{-4}	34.2735
2.0	10^{-1}	2.03×10^{-6}	74.9226	2.03×10^{-6}	66.7614
	10^{-3}	4.52×10^{-4}	74.1522	4.30×10^{-4}	66.5531
3.0	10^{-1}	1.76×10^{-6}	111.3824	1.76×10^{-6}	100.0451
	10^{-3}	3.08×10^{-4}	110.8592	3.00×10^{-4}	99.9561

5 Conclusions

A modified error correction scheme has been developed for efficiently finding numerical solutions in the BSLM framework. Instead of using the traditional way to find the departure points of the particles, we suggest a new technique by constructing new Euler polygon in the error correction strategy, depending on the given boundary conditions. To reduce the computational cost, the proposed method approximates the Jacobian value by a fixed value while maintaining the scale of error, whereas the conventional ECM performs the interpolation with derivatives newly updated in each time integration step. Through several numerical results, the proposed method has a convergence rate of 2 in time. Also, it is shown that the proposed method obtains outstanding numerical results compared with the existing methods, and it well preserves the energy and mass.

Funding

The first author, Bak, was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (grant number NRF-2017R1A6A3A01002341). The second author, Kim, the third author, Piao, and the corresponding author, Bu, were supported by the basic science research program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (NRF-2016R1A2B2011326, NRF-2017R1C1B1002370 and NRF-2016R1D1A1B03930734), respectively.

Competing interests

The authors declare that they have no competing interests.

Authors' contributions

All authors jointly worked on the results and they read and approved the final manuscript.

Author details

¹Department of Mathematics, Kyungpook National University, Daegu, Republic of Korea. ²Department of Liberal Arts, Hongik University, Sejong, Republic of Korea.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Received: 30 November 2017 Accepted: 19 November 2018 Published online: 23 November 2018

References

- Xiu, D., Karniadakis, G.E.: A semi-Lagrangian high-order method for Navier–Stokes equations. *J. Comput. Phys.* **172**, 658–684 (2001)
- Hassanien, I.A., Salama, A.A., Hosham, H.A.: Fourth-order finite difference method for solving Burgers' equation. *Appl. Math. Comput.* **170**, 781–800 (2005)
- Dehghan, M., Shokri, A.: A numerical method for two-dimensional Schrödinger equation using collocation and radial basis functions. *Comput. Math. Appl.* **54**, 136–146 (2007)
- Khater, A.H., Tamsah, R.S., Hassan, M.M.: A Chebyshev spectral collocation method for solving Burgers'-types equations. *J. Comput. Appl. Math.* **222**, 333–350 (2008)
- Liao, W.: An implicit fourth-order compact finite difference scheme for one-dimensional Burgers' equation. *Appl. Math. Comput.* **206**, 755–764 (2008)
- Asaithambi, A.: Numerical solution of the Burgers equation by automatic differentiation. *Appl. Math. Comput.* **216**, 2700–2708 (2010)

7. Wang, J., Layton, A.: New numerical methods for Burgers' equation based on semi-Lagrangian and modified equation approaches. *Appl. Numer. Math.* **60**, 645–657 (2010)
8. Zhang, L., Ouyang, J., Wang, X., Zhang, X.: Variational multiscale element-free Galerkin method for 2D Burgers' equation. *J. Comput. Phys.* **229**, 7147–7161 (2010)
9. Mittal, R.C., Jain, R.K.: Numerical solutions of nonlinear Burgers' equation with modified cubic B-splines collocation method. *Appl. Math. Comput.* **218**, 7839–7855 (2012)
10. Arora, G., Singh, B.K.: Numerical solution of Burgers' equation with modified cubic B-spline differential quadrature method. *Appl. Math. Comput.* **224**, 166–177 (2013)
11. Jiwari, R., Mittal, R.C., Sharma, K.K.: A numerical scheme based on weighted average differential quadrature method for the numerical solution of Burgers' equation. *Appl. Math. Comput.* **219**, 6680–6691 (2013)
12. Zhang, X., Tian, H., Chen, W.: Local method of approximate particular solutions for two-dimensional unsteady Burgers' equations. *Comput. Math. Appl.* **66**, 2425–2432 (2014)
13. Shi, Y., Xu, B., Guo, Y.: Numerical solution of Korteweg–de Vries–Burgers equation by the compact-type CIP method. *Adv. Differ. Equ.* **2015**, 353 (2015)
14. Bak, S., Kim, P., Kim, D.: A semi-Lagrangian approach for numerical simulation of coupled Burgers' equations. *Commun. Nonlinear Sci. Numer. Simul.* **69**, 31–44 (2019)
15. Kim, P., Piao, X., Kim, S.D.: An error corrected Euler method for solving stiff problems based on Chebyshev collocation. *SIAM J. Numer. Anal.* **49**, 2211–2230 (2011)
16. Kim, S.D., Piao, X., Kim, P.: Convergence on error correction methods for solving initial value problem. *J. Comput. Appl. Math.* **236**, 4448–4461 (2012)
17. Piao, X., Bu, S., Bak, S., Kim, P.: An iteration free backward semi-Lagrangian scheme for solving incompressible Navier–Stokes equations. *J. Comput. Phys.* **283**, 189–204 (2015)
18. Piao, X., Kim, S., Kim, P., Kwon, J., Yi, D.: An interaction free backward semi-Lagrangian scheme for guiding center problem. *SIAM J. Numer. Anal.* **53**, 619–643 (2015)
19. Ewing, R.E., Wang, H.: A summary of numerical methods for time-dependent advection-dominated partial differential equations. *J. Comput. Appl. Math.* **128**, 423–445 (2001)
20. Fornberg, B.: Calculation of weights in finite difference formulas. *SIAM Rev.* **40**, 685–691 (1998)
21. Evans, C.L.: *Partial Differential Equations*. Am. Math. Soc., Providence (1998)
22. Ladyzenskaya, O.A., Solonnikov, V.A., Uralceva, N.N.: *Linear and Quasilinear Equations of Parabolic Type*. Am. Math. Soc., Providence (1968)
23. Kim, D., Song, O., Ko, H.-S.: A semi-Lagrangian CIP fluid solver without dimensional splitting. *Comput. Graph. Forum* **27**, 467–475 (2008)
24. Anguelov, R., Djoko, J.K., Lubuma, J.M.-S.: Energy properties preserving schemes for Burgers' equation. *Numer. Methods Partial Differ. Equ.* **24**, 41–59 (2008)

Submit your manuscript to a SpringerOpen[®] journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

Submit your next manuscript at ► [springeropen.com](https://www.springeropen.com)
