J Sci Comput
DOI 10.1007/s10915-017-0536-2

A Projection Method for the Conservative Discretizations of Parabolic Partial Differential Equations

Darae Jeong[1] · **Junseok Kim**[1]

Received: 1 September 2016 / Revised: 3 June 2017 / Accepted: 7 August 2017
© Springer Science+Business Media, LLC 2017

Abstract We present a projection method for the conservative discretizations of parabolic partial differential equations. When we solve a system of discrete equations arising from the finite difference discretization of the PDE, we can use iterative algorithms such as conjugate gradient, generalized minimum residual, and multigrid methods. An iterative method is a numerical approach that generates a sequence of improved approximate solutions for a system of equations. We repeat the iterative algorithm until a numerical solution is within a specified tolerance. Therefore, even though the discretization is conservative, the actual numerical solution obtained from an iterative method is not conservative. We propose a simple projection method which projects the non-conservative numerical solution into a conservative one by using the original scheme. Numerical experiments demonstrate the proposed scheme does not degrade the accuracy of the original numerical scheme and it preserves the conservative quantity within rounding errors.

Keywords Projection method · Conservative discretization · Iterative methods

1 Introduction

There are many conservative parabolic partial differential equations such as diffusion equation and the Cahn–Hilliard equation [1–3]. In those equations, conservation is important, i.e., the conservation of total mass. It is generally known that the conservative schemes are better than the nonconservative ones [4]. For some cases, the nonconservative schemes can generate blow-up of the numerical solution. Authors in [5–19] used conservative finite difference schemes for

✉ Junseok Kim
cfdkim@korea.ac.kr

Darae Jeong
tinayoyo@korea.ac.kr

[1] Department of Mathematics, Korea University, Seoul 02841, Republic of Korea

Published online: 17 August 2017

- Nonlinear Schrödinger equation [5–9]: $iU_t + U_{xx} + a|U|^2U = 0$,
- Regularized long wave equation [10]: $U_t + U_x + \beta UU_x - \gamma^2 U_{xxt} = 0$,
- Sine–Gordon equation [11]: $U_{tt} - U_{xx} + \sin U = 0$,
- Klein–Gordon equation [12]: $U_{tt} - U_{xx} + f(U) = 0$,
- Zakharov equation [13,14]: $iU_t - U_{xx} - NU = 0,\ N_{tt} - (N + |U|^2)_{xx} = 0$,
- Cahn–Hilliard equation [15–18]: $c_t = \Delta\mu,\ \mu = c^3 - \frac{3}{2}c^2 + \frac{1}{2}c - \epsilon^2\Delta c$,
- Convective Cahn–Hilliard equation [19]: $c_t + \nabla \cdot (c\mathbf{u}) = \frac{1}{Pe}\nabla \cdot (M(c)\nabla\mu)$, $\mu = c^3 - \frac{3}{2}c^2 + \frac{1}{2}c - \epsilon^2\Delta c$.

To numerically approximate the solution of the equations, we normally use iterative methods such as Jacobi [20], Gauss–Seidel [20], successive over relaxation [20], multigrid [21–24], conjugate gradient [20], GMRES (Generalized Minimal Residual) [25] methods. In the iterative methods, we iterate an algorithm until the norm of the residual reaches within a given tolerance and then we define the final solution as the next time step solution. We also use the norm of the difference of the consecutive solutions instead of the residual. Typically, the given tolerance is much larger than the rounding errors. Therefore, the numerical approximation does not satisfy the discrete system of equations within rounding errors. In that sense, most conservative schemes with iterative methods are in fact not conservative. The main purpose of this paper is to propose a conservative projection scheme for existing conservative schemes. The proposed scheme conserves conservative quantities within rounding errors.

This paper is organized as follows. In Sect. 2, we proposed a projection method for conservative numerical schemes. In Sect. 3, we present several numerical results showing the performance of the proposed scheme with the heat equation and the Cahn–Hilliard equation. Then, we summarize our results in Sect. 4.

2 Conservative Projection Scheme

For simplicity of exposition, we consider a one-dimensional diffusion equation on a unit domain.

$$u_t(x,t) = u_{xx}(x,t), \quad 0 < x < 1,\ t > 0, \tag{1}$$

with the homogeneous Neumann boundary condition

$$u_x(0,t) = u_x(1,t) = 0, \quad t > 0 \tag{2}$$

and initial conditions $u(x,0),\ 0 \le x \le 1$. For the solution u of the diffusion equation (1) with the boundary condition (2), the following conservation property holds

$$\frac{d}{dt}\int_0^1 u(x,t)dx = \int_0^1 u_t(x,t)dx = \int_0^1 u_{xx}(x,t)dx = u_x(1,t) - u_x(0,t) = 0.$$

We want to point out that our results can be extended to more general parabolic partial differential equations. Let us first discretize the given computational domain $\Omega = (0,1)$ with a uniform space step $h = 1/N_x$ and a time step $\Delta t = T/N_t$. Here, N_x and N_t are the number of grid points in the x- and t-direction, respectively. Furthermore, let us denote the numerical approximation of the solution by $u_i^n \equiv u(x_i,t_n) = u\left((i-0.5)h, n\Delta t\right)$, where $i = 1,\ldots,N_x$ and $n = 0,\ldots,N_t$. The backward-time central-space difference scheme for the diffusion equation (1) is

$$\frac{u_i^{n+1}-u_i^n}{\Delta t}=\Delta_d u_i^{n+1}=\frac{u_{i-1}^{n+1}-2u_i^{n+1}+u_{i+1}^{n+1}}{h^2},\quad 1\le i\le N_x. \tag{3}$$

The homogenous Neumann boundary condition (2) is discretized as

$$u_0^{n+1}=u_1^{n+1},\ u_{N_x+1}^{n+1}=u_{N_x}^{n+1}. \tag{4}$$

This scheme is second-order accurate in space and first-order in time. We rewrite Eq. (3) as

$$-\alpha u_{i-1}^{n+1}+(1+2\alpha)\,u_i^{n+1}-\alpha u_{i+1}^{n+1}=u_i^n,\quad 1\le i\le N_x, \tag{5}$$

where$\alpha=\Delta t/h^2$. Then, we can represent the system of discrete equation (5) in matrix form as

$$A\mathbf{u}^{n+1}=\mathbf{u}^n, \tag{6}$$

where $\mathbf{u}=(u_1,\ldots,u_{N_x})^T$ and A is a tridiagonal matrix constructed from Eq. (5) with the homogeneous Neumann boundary condition, i.e.,

$$A=\begin{pmatrix}1+\alpha & -\alpha & & & 0\\ -\alpha & 1+2\alpha & -\alpha & & \\ & \ddots & \ddots & \ddots & \\ & & -\alpha & 1+2\alpha & -\alpha\\ 0 & & & -\alpha & 1+\alpha\end{pmatrix}.$$

We can directly solve Eq. (6) by using the Thomas algorithm [20] and have a solution exactly up to a machine precision or rounding errors. While the discrete system of Eq. (5) can be solved by a direct method such as the Thomas algorithm, we consider an iterative solver to illustrate our projection method. Although many iterative methods are available, among them we choose a Gauss–Seidel (GS) iteration since it is one of simplest iterative methods. Let $\mathbf{u}^{n+1,0}=\mathbf{u}^n$ be an initial guess. In the GS iteration, for $m=1,\ldots,$ we iterate the following algorithm until the updated solution converges under a specific condition:

$$\begin{aligned}
u_1^{n+1,m}&=\left(u_1^n+\alpha u_2^{n+1,m-1}\right)/(1+\alpha),\\
u_i^{n+1,m}&=\left[u_i^n+\alpha\left(u_{i-1}^{n+1,m}+u_{i+1}^{n+1,m-1}\right)\right]/(1+2\alpha),\ 2\le i\le N_x-1,\\
u_{N_x}^{n+1,m}&=\left(u_{N_x}^n+\alpha u_{N_x-1}^{n+1,m}\right)/(1+\alpha).
\end{aligned} \tag{7}$$

If it converges, then we define $\mathbf{u}^{n+1}=\mathbf{u}^{n+1,m}$.

Let $\mathbf{r}=(r_1,\ldots,r_{N_x})^T$ be the residual vector for $\mathbf{u}^{n+1}$ with respect to Eq. (3), i.e,

$$r_i=\frac{u_i^{n+1}-u_i^n}{\Delta t}-\frac{u_{i-1}^{n+1}-2u_i^{n+1}+u_{i+1}^{n+1}}{h^2},\ 1\le i\le N_x, \tag{8}$$

where r_i is a residual error, that is the magnitude by which the approximation $\mathbf{u}^{n+1}$ fails to satisfy the original problem (3). We define the discrete l_2-norm and the maximum norm as

$$\|\mathbf{r}\|_2=\sqrt{\sum_{i=1}^{N_x}r_i^2/N_x},\quad \|\mathbf{r}\|_\infty=\max_{1\le i\le N_x}|r_i|. \tag{9}$$

However, the numerical approximation from the iterative method has non-zero residual vector $\mathbf{r}$ despite use of conservative scheme.

Therefore, in this paper, we propose the following projection method for the conservative schemes: Let $\mathbf{u}^*$ be a numerical solution from an iterative method for the discrete equation (3), then it satisfies the following equation: $(u_i^* - u_i^n)/\Delta t \approx \Delta_d u_i^*,\ 1 \le i \le N_x$. Typically, we do not keep the iteration until the residual norm reaches rounding errors. Instead, we iterate it until the error is within a tolerance. Then, we define the conservative numerical solution $\mathbf{u}^{n+1}$ as

$$u_i^{n+1} = u_i^n + \Delta t \Delta_d u_i^*,\ 1 \le i \le N_x, \tag{10}$$

where we applied the homogeneous Neumann boundary condition, i.e., $u_0^* = u_1^*$, $u_{N_x+1}^* = u_{N_x}^*$. Summing Eq. (10) from 1 to N_x, we have

$$\sum_{i=1}^{N_x} u_i^{n+1} = \sum_{i=1}^{N_x} (u_i^n + \Delta t \Delta_d u_i^*) = -\frac{u_1^* - u_0^*}{h^2} + \frac{u_{N_x+1}^* - u_{N_x}^*}{h^2} + \sum_{i=1}^{N_x} u_i^n = \sum_{i=1}^{N_x} u_i^n,$$

where we have used the discrete homogeneous Neumann boundary conditions (4). Therefore, the proposed conservative projection method is indeed numerically conservative. For the conservative numerical schemes of the general parabolic equations, we can straightforwardly define the conservative projection method.

3 Numerical Experiments

In this section, we report numerical experiments demonstrating the good conservation property of our proposed conservative projection method. Unless otherwise specified, we use a unit domain $\Omega = (0, 1)$.

3.1 Convergence Test

As the first numerical experiment, we perform convergence tests. The numerical scheme for the diffusion equation (3) is second-order accurate in space and first-order accurate in time. To obtain an estimate of the rate of convergence, we perform a number of simulations for a sample initial problem on a set of increasingly finer grids. First, we perform the spatial convergence test. We compute the numerical solutions on uniform grids, $h = 0.1/2^n$ for $n = 0, 1, \ldots, 7$. For each case, we run the calculation up to time $T = 500\Delta t$ with a fixed time step, $\Delta t = 10^{-9}$ and a tolerance, $tol = 1.0\text{e}{-11}$. The initial condition is $u(x, 0) = \cos(2\pi x)$. Then, the analytic solution of Eq. (1) is $\tilde{u}(x, t) = \cos(2\pi x)e^{-4\pi^2 t}$. Let $\mathbf{e}$ be the error vector with components $e_i = \tilde{u}(x_i, T) - u_i^{N_t}$. Tables 1 and 2 demonstrate the l_2-norm and maximum norm errors and rates of convergence of Thomas algorithm, GS, and GS with projection, respectively. The results show that the numerical schemes are second-order accurate in space.

Next, we perform the temporal convergence test. We fix the spatial step size as $h = 0.0025$ and we run the simulation up to time $T = 0.1$ with $\Delta t = 0.01/2^n$ for $n = 0, 1, \ldots, 7$. The other parameters are the same to the space convergence test. The l_2- and maximum norm errors and rates of convergence are given in Tables 3 and 4, respectively. The results show that the schemes are first-order accurate in time. From these spatial and temporal convergence tests, we can confirm that the proposed conservative projection method does not degrade the accuracy of the original numerical scheme.

We also consider the effect of the tolerance, tol, on the numerical soluts ions by the GS and GS with projection scheme. Table 5 shows l_2-norm error $\|\mathbf{e}\|_2$, maximum norm error

Table 1 l_2-norm error and convergence rates for numerical solution at $T = 500\Delta t$ with respect to h

	Thomas		GS		GS with projection	
h	$\|\mathbf{e}\|_2$	Rate	$\|\mathbf{e}\|_2$	Rate	$\|\mathbf{e}\|_2$	Rate
0.10000000	4.5318e–07		4.5318e–07		4.5318e–07	
0.05000000	1.1442e–07	1.9858	1.1442e–07	1.9858	1.1442e–07	1.9858
0.02500000	2.8675e–08	1.9964	2.8676e–08	1.9964	2.8676e–08	1.9964
0.01250000	7.1735e–09	1.9991	7.1735e–09	1.9991	7.1735e–09	1.9991
0.00625000	1.7939e–09	1.9996	1.7939e–09	1.9996	1.7939e–09	1.9996
0.00312500	4.4873e–10	1.9992	4.4887e–10	1.9987	4.4869e–10	1.9993
0.00156250	1.1235e–10	1.9979	1.1237e–10	1.9980	1.1238e–10	1.9973
0.00078125	2.8239e–11	1.9922	2.8335e–11	1.9877	2.8302e–11	1.9894

Here, we use $\Delta t = 10^{-9}$

Table 2 Maximum norm error and convergence rates for numerical solution at $T = 500\Delta t$ with respect to h

	Thomas		GS		GS with projection	
h	$\|\mathbf{e}\|_\infty$	Rate	$\|\mathbf{e}\|_\infty$	Rate	$\|\mathbf{e}\|_\infty$	Rate
0.10000000	6.0953e–07		6.0953e–07		6.0953e–07	
0.05000000	1.5982e–07	1.9312	1.5982e–07	1.9312	1.5982e–07	1.9312
0.02500000	4.0428e–08	1.9830	4.0428e–08	1.9830	4.0428e–08	1.9830
0.01250000	1.0137e–08	1.9957	1.0137e–08	1.9957	1.0137e–08	1.9957
0.00625000	2.5365e–09	1.9988	2.5365e–09	1.9987	2.5364e–09	1.9988
0.00312500	6.3458e–10	1.9989	6.3518e–10	1.9976	6.3451e–10	1.9991
0.00156250	1.5892e–10	1.9975	1.5892e–10	1.9988	1.5893e–10	1.9973
0.00078125	3.9992e–11	1.9905	4.0259e–11	1.9809	4.0025e–11	1.9894

Here, we use $\Delta t = 10^{-9}$

Table 3 l_2-norm error and convergence rates for numerical solution with respect to Δt

	Thomas		GS		GS with projection	
Δt	$\|\mathbf{e}\|_2$	Rate	$\|\mathbf{e}\|_2$	Rate	$\|\mathbf{e}\|_2$	Rate
0.010000000	1.1732e–2		1.1732e–2		1.1732e–2	
0.005000000	5.6212e–3	1.0615	5.6213e–3	1.0615	5.6212e–3	1.0615
0.002500000	2.7393e–3	1.0371	2.7393e–3	1.0371	2.7393e–3	1.0371
0.001250000	1.3507e–3	1.0200	1.3508e–3	1.0200	1.3507e–3	1.0200
0.000625000	6.7087e–4	1.0096	6.7089e–4	1.0096	6.7087e–4	1.0096
0.000312500	3.3470e–4	1.0032	3.3472e–4	1.0031	3.3470e–4	1.0032
0.000156250	1.6758e–4	0.9980	1.6759e–4	0.9980	1.6758e–4	0.9980
0.000078125	8.4262e–5	0.9919	8.4273e–5	0.9918	8.4262e–5	0.9919

Here, we use $h = 0.0025$ and $T = 0.1$

$\|\mathbf{e}\|_\infty$, and total mass change $M = \left|\sum_{i=1}^{N_x} u_i^0 - \sum_{i=1}^{N_x} u_i^{N_t}\right|$ for numerical solutions at time T with respect to tol. In this test, we use $h = 5.0\text{e–}3$, $\Delta t = 1.0\text{e–}4$, and $T = 0.1$. Both the methods generate more accurate results as tol decreases. However, Gauss–Seidel method

Table 4 Maximum norm error and convergence rates for numerical solution with respect to Δt

	Thomas		GS		GS with projection	
Δt	$\|\mathbf{e}\|_\infty$	Rate	$\|\mathbf{e}\|_\infty$	Rate	$\|\mathbf{e}\|_\infty$	Rate
0.010000000	1.6592e–2		1.6592e–2		1.6592e–2	
0.005000000	7.9494e–3	1.0615	7.9495e–3	1.0615	7.9494e–3	1.0615
0.002500000	3.8738e–3	1.0371	3.8738e–3	1.0371	3.8738e–3	1.0371
0.001250000	1.9102e–3	1.0200	1.9102e–3	1.0200	1.9102e–3	1.0200
0.000625000	9.4872e–4	1.0096	9.4877e–4	1.0096	9.4872e–4	1.0096
0.000312500	4.7332e–4	1.0032	4.7336e–4	1.0031	4.7332e–4	1.0032
0.000156250	2.3699e–4	0.9980	2.3701e–4	0.9980	2.3699e–4	0.9980
0.000078125	1.1916e–4	0.9919	1.1918e–4	0.9918	1.1916e–4	0.9919

Here, we use $h = 0.0025$ and $T = 0.1$

Table 5 l_2-norm error, maximum norm error, and total mass change for numerical solution at T with respect to tol

	GS			GS with projection		
tol	$\|\mathbf{e}\|_2$	$\|\mathbf{e}\|_\infty$	Mass loss	$\|\mathbf{e}\|_2$	$\|\mathbf{e}\|_\infty$	Mass loss
1.0e–3	3.0958e–1	4.4281e–1	2.1547e–3	5.8452e–4	2.4260e–3	1.0492e–16
1.0e–4	6.0997e–2	8.9848e–2	1.3709e–3	4.1592e–5	1.8559e–4	1.0922e–16
1.0e–5	4.6259e–3	7.2884e–3	3.0182e–4	9.0322e–5	1.2939e–4	1.4447e–16
1.0e–6	4.4400e–4	7.2683e–4	4.1756e–5	1.0961e–4	1.5511e–4	1.3864e–16
1.0e–7	1.3752e–4	2.0686e–4	4.9663e–6	1.1081e–4	1.5670e–4	1.0936e–16
1.0e–8	1.1315e–4	1.6144e–4	5.5875e–7	1.1090e–4	1.5682e–4	1.4759e–16
1.0e–9	1.1111e–4	1.5727e–4	6.0961e–8	1.1091e–4	1.5683e–4	1.3316e–16

Here, we use $h = 5.0\text{e–}3$, $\Delta t = 1.0\text{e–}4$, and $T = 0.1$

results in change of the total mass and the proposed projection method conserves total mass within rounding errors regardless of tol.

3.2 Stability Test

The proposed projection method consists of two steps. First step is to solve the governing parabolic PDE by using an iterative algorithm. After then, we project the non-conservative numerical solution into a conservative one by using the governing equation. The second step is similar to the explicit Euler scheme. Now, we investigate a stability of numerical solution by the proposed projection method. For simplicity of exposition, let us consider the one-dimensional heat equation $u_t = \Delta u$. That is, the second step in the projection method is given as

$$u^{n+1} = u^n + \Delta t \Delta_d u^*.$$

Here, u^* denotes the numerical approximation of an implicit scheme, which is obtained by the first step, not u^n. Since u^* is evaluated by the iterative algorithm within a specified tolerance, the value is nearly close to u^{n+1}. Therefore, it is different from the numerical solution from the explicit scheme. The correction step does not have the same restriction of temporal

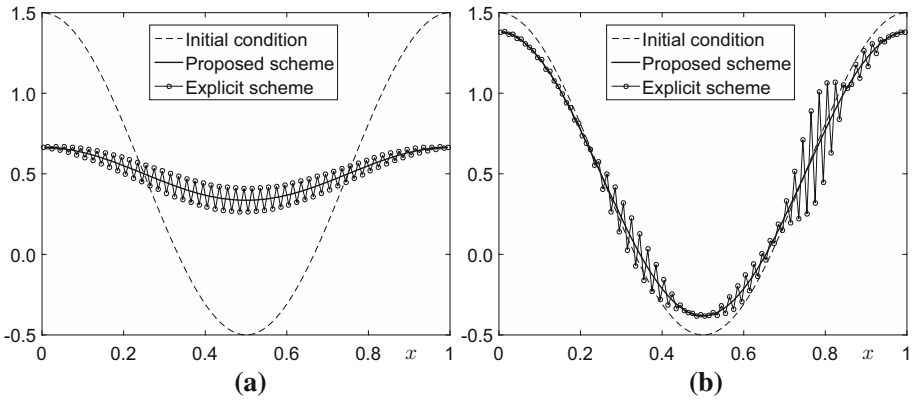

Fig. 1 Temporal evolution of numerical solution by the proposed and explicit schemes with the stability condition $\Delta t/h^2$. Here, we fix $h = 0.01$ and $tol = 1.0\text{e}{-5}$. Numerical results in **a** and **b** are obtained at $T = 900\Delta t$ and $T = 33\Delta t$, respectively. **a** $\Delta t/h^2 = 0.51$. **b** $\Delta t/h^2 = 1$

and spatial step sizes, unlike the explicit scheme. It is well known that the ratio $\Delta t/h^2$ should be smaller than 0.5 for stability of the numerical solution of the one-dimensional heat equation $u_t = \Delta u$ by the explicit scheme. Figure 1a, b show the temporal evolutions of the numerical solutions by the proposed and explicit schemes with $\Delta t/h^2 = 0.51$ and $\Delta t/h^2 = 1$, respectively. Here, we use $h = 0.01$ and $tol = 1.0\text{e}{-5}$. The initial condition is set to $u(x, 0) = 0.5 + \cos(2\pi x)$. The results numerically demonstrate the proposed scheme has better stability than the explicit scheme.

3.3 Comparison of Residual and Consecutive Errors

In this section, we perform comparison study of the residual and consecutive errors of the Gauss–Seidel method. From the test results, we want to emphasize that unless we have approximate solutions up to rounding errors of the residual norm, we lose the conservation property even though we use conservative discretizations. With $\mathbf{u}^0 = \cos(2\pi\mathbf{x})$, $h = 0.01$, $\Delta t = 0.01$, and $tol = 1.0\text{e} - 9$, we calculate two l_2-norm errors : $\|\mathbf{r}^m\|_2$ for residual error and $\|\mathbf{u}^{1,m}\text{-}\mathbf{u}^{1,m-1}\|_2$ for consecutive error with respect to the iteration number m during one time step. Here, $\mathbf{r}^m = (\mathbf{u}^{1,m} - \mathbf{u}^0)/\Delta t - \Delta_d \mathbf{u}^{1,m}$. The numerical results are shown in Fig. 2.

Note that we plot l_2-norm of residual error and consecutive error versus m with y-axis labeled on the right and the left, respectively. The values next to the circle and star symbols are l_2-norms of the residual and the consecutive errors with $m = 10$, 100, and 1000, respectively. As shown in Fig. 2, the convergence of the consecutive error is faster than that of the residual error.

Table 6 lists the iteration numbers required by the residual and consecutive l_2-norm errors to be smaller than the prescribed tolerance with various time step sizes $\Delta t = 0.1,\ 0.01$, and 0.001. Here, we fix $h = 0.001$, $T = \Delta t$, and $\mathbf{u}^0 = \cos(2\pi\mathbf{x})$. We can also check that the convergence speed of the consecutive error is faster than that of the residual error. With the larger time step size and tolerance, the fast convergence of the consecutive error is more evident.

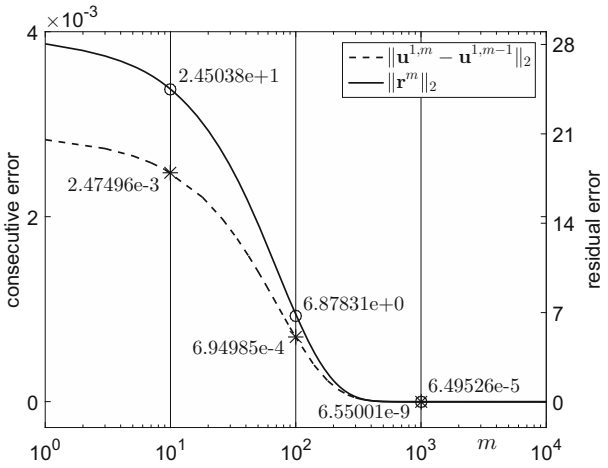

Fig. 2 Discrete l_2-norms of the residual and consecutive errors with respect to the iteration number m during one time step. Here, we use $h = 0.01$, $\Delta t = 0.01$, and $tol = 1.0\text{e}{-9}$

Table 6 Iteration numbers required by the residual and consecutive l_2-norm errors to be smaller than the prescribed tolerance with various time step sizes $\Delta t = 0.1,\ 0.01$, and 0.001

	$\Delta t = 0.1$		$\Delta t = 0.01$		$\Delta t = 0.001$	
tol	$\|\mathbf{r}^m\|_2$	$\|\mathbf{u}^m - \mathbf{u}^{m-1}\|_2$	$\|\mathbf{r}^m\|_2$	$\|\mathbf{u}^m - \mathbf{u}^{m-1}\|_2$	$\|\mathbf{r}^m\|_2$	$\|\mathbf{u}^m - \mathbf{u}^{m-1}\|_2$
1.0e-6	919958	67194	124026	23838	16471	3201
1.0e-7	1149987	113680	142757	40316	18684	5412
1.0e-8	1380015	161144	163391	56800	20896	7624
1.0e-9	1610092	244662	185388	73304	23116	9836

Here, we fix $h = 0.001$ and $T = \Delta t$

3.4 Conservative Property

We now consider the main test problem of the conservation property. The numerical initial condition is $u_i^0 = 0.5 + 10\cos(2\pi x_i) + \alpha$ for $i = 1, \ldots, N_x$, where α is a discrete mass correction parameter so that $\sum_{i=1}^{N_x} u_i^0 h = 0.5$ is satisfied. Here, we use $N_x = 1000$ and $T = 0.3$.

Figures 3a, b show the temporal behavior of discrete total mass $\sum_{i=1}^{N_x} u_i^n h$ of the numerical solution by GS method and GS method with projection with different time step size Δt and tolerance tol, respectively. Here, we use the following parameters : $tol = 1.0\text{e}{-6}$ for Fig. 3a and $\Delta t = 0.01$ for Fig. 3b. The results from GS with projection are consistent to initial discrete total mass as shown in Fig. 3. However, the results from GS have the deviation from the initial discrete total mass. These deviations are getting larger as time step size or tolerance is getting larger.

3.5 Conservative Property on the Complex Domain

In this section, we study how the proposed method behaves when the complex boundary is described by a level-set function. As test problems, we consider the two-dimensional heat

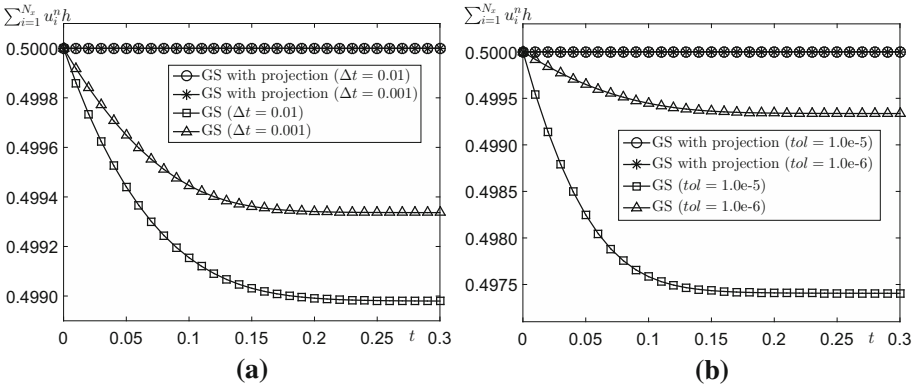

Fig. 3 Temporal behavior of discrete total mass of the numerical solution by GS and GS with projection with different **a** time step size Δt and **b** tolerance, tol

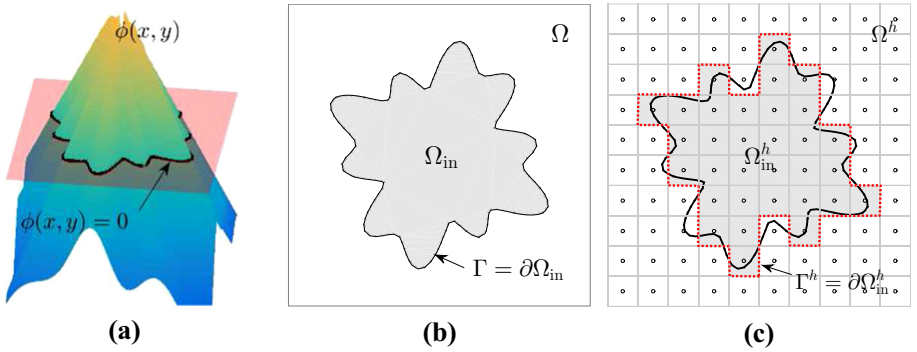

Fig. 4 **a** Illustration of surface plot of level-set function $\phi(x, y)$ with the zero contour $\phi(x, y) = 0$. **b** Complex domain Ω_{in} and its boundary $\Gamma = \partial\Omega_{\text{in}}$ embedded in Ω. **c** Discrete complex domain Ω^h_{in} and its boundary Γ^h

equation on a complex domain with homogeneous Neumann boundary conditions. Figure 4b shows the complex domain Ω_{in} and its boundary $\Gamma = \partial\Omega_{\text{in}}$ that is embedded in the whole domain Ω. Here, the boundary curve Γ is determined by the level-set function $\phi(x, y)$ (see Fig. 4a) such that ϕ is negative inside and positive outside the curve $\Gamma = \{(x, y)|\phi(x, y) = 0\}$.

To solve the heat equation in the complex domain Ω^h_{in} (see Fig. 4c), we propose the following numerical scheme [38]:

$$\frac{u_{ij}^{n+1} - u_{ij}^n}{\Delta t} = \nabla_d \cdot \left(G_{ij} M_{ij} \nabla_d u_{ij}^{n+1} \right), \tag{11}$$

for $i = 1, \ldots, N_x$ and $j = 1, \ldots, N_y$. Here, M_{ij} denotes the diffusion coefficient at (x_i, y_j) and the boundary control value G_{ij} is defined in Ω^h as

$$G_{ij} = G(x_i, y_j) = \begin{cases} 1, & \text{if } (x_i, y_j) \in \Omega^h_{\text{in}}, \\ 0, & \text{if } (x_i, y_j) \in \Omega^h_{\text{out}} = \Omega \backslash \Omega^h_{\text{in}}, \end{cases} \tag{12}$$

where Ω_{in}^h is an open region that consists of interior points of the union of the closed cells whose centers are inside the interface Γ. By the central-space difference scheme, Eq. (11) is

$$\begin{aligned}\frac{u_{ij}^{n+1}-u_{ij}^n}{\Delta t} = \frac{1}{h^2}\Big[&G_{i+\frac{1}{2},j}M_{i+\frac{1}{2},j}\left(u_{i+1,j}^{n+1}-u_{ij}^{n+1}\right)\\ &-G_{i-\frac{1}{2},j}M_{i-\frac{1}{2},j}\left(u_{ij}^{n+1}-u_{i-1,j}^{n+1}\right)+G_{ij+\frac{1}{2}}M_{ij+\frac{1}{2}}\left(u_{ij+1}^{n+1}-u_{ij}^{n+1}\right)\\ &-G_{ij-\frac{1}{2}}M_{ij-\frac{1}{2}}\left(u_{ij}^{n+1}-u_{ij-1}^{n+1}\right)\Big]. \end{aligned} \tag{13}$$

Note that $G_{i+\frac{1}{2},j}=(G_{i+1,j}+G_{ij})/2$, $M_{i+\frac{1}{2},j}=(M_{i+1,j}+M_{ij})/2$, and the other terms are similarly defined. Rewriting Eq. (13), we get

$$\begin{aligned}&\left(\frac{1}{\Delta t}+\frac{G_{i+\frac{1}{2},j}M_{i+\frac{1}{2},j}+G_{i-\frac{1}{2},j}M_{i-\frac{1}{2},j}+G_{ij+\frac{1}{2}}M_{ij+\frac{1}{2}}+G_{ij-\frac{1}{2}}M_{ij-\frac{1}{2}}}{h^2}\right)u_{ij}^{n+1}\\ &\quad=\frac{u_{ij}^n}{\Delta t}+\frac{G_{i+\frac{1}{2},j}M_{i+\frac{1}{2},j}u_{i+1,j}^{n+1}+G_{i-\frac{1}{2},j}M_{i-\frac{1}{2},j}u_{i-1,j}^{n+1}}{h^2}\\ &\qquad+\frac{G_{ij+\frac{1}{2}}M_{ij+\frac{1}{2}}u_{ij+1}^{n+1}+G_{ij-\frac{1}{2}}M_{ij-\frac{1}{2}}u_{ij-1}^{n+1}}{h^2}.\end{aligned}$$

By the GS iterative method, we iterate the following algorithm until the updated solution converges under a specific tolerance as

$$\begin{aligned}u_{ij}^{n+1,m}=&\left[\frac{u_{ij}^n}{\Delta t}+\frac{G_{i+\frac{1}{2},j}M_{i+\frac{1}{2},j}u_{i+1,j}^{n+1,m-1}+G_{i-\frac{1}{2},j}M_{i-\frac{1}{2},j}u_{i-1,j}^{n+1,m}}{h^2}\right.\\ &\left.+\frac{G_{ij+\frac{1}{2}}M_{ij+\frac{1}{2}}u_{ij+1}^{n+1,m-1}+G_{ij-\frac{1}{2}}M_{ij-\frac{1}{2}}u_{ij-1}^{n+1,m}}{h^2}\right]\\ &\Big/\left(\frac{1}{\Delta t}+\frac{G_{i+\frac{1}{2},j}M_{i+\frac{1}{2},j}+G_{i-\frac{1}{2},j}M_{i-\frac{1}{2},j}}{h^2}\right.\\ &\left.+\frac{G_{ij+\frac{1}{2}}M_{ij+\frac{1}{2}}+G_{ij-\frac{1}{2}}M_{ij-\frac{1}{2}}}{h^2}\right)\end{aligned} \tag{14}$$

for $m=1,\ldots$. Here, m denotes the iteration index of the GS method during one time step. Then, for the mass conservation, we apply the following projection method as

$$u_{ij}^{n+1}=u_{ij}^n+\Delta t\ \nabla_d\cdot(G_{ij}M_{ij}\nabla_d u_{ij}^*)\ \text{ for all }\ i,\ j,$$

where u_{ij}^* is the numerical solution obtained from Eq. (11).

Now, we implement two tests about heat equation with constant and piecewise constant diffusion coefficient functions. As the first test, we consider the heat equation with the constant diffusion function $M(x,y)=1$ for all x and y. In the other test, we investigate dynamics of the heat equation with a piecewise constant diffusion function as

$$M(x,y)=\begin{cases}0.1 & \text{if } 0\le x\le 0.5,\\ 1 & \text{otherwise.}\end{cases}$$

For numerical tests, we use $h=0.0025$, $\Delta t=0.001$, $T=0.02$, and the initial condition $u(x,y,0)=\sin(\pi x)\sin(\pi y)$ on the complex domain $\Omega_{\text{in}}\subset\Omega=(0,1)\times(0,1)$. Here, the

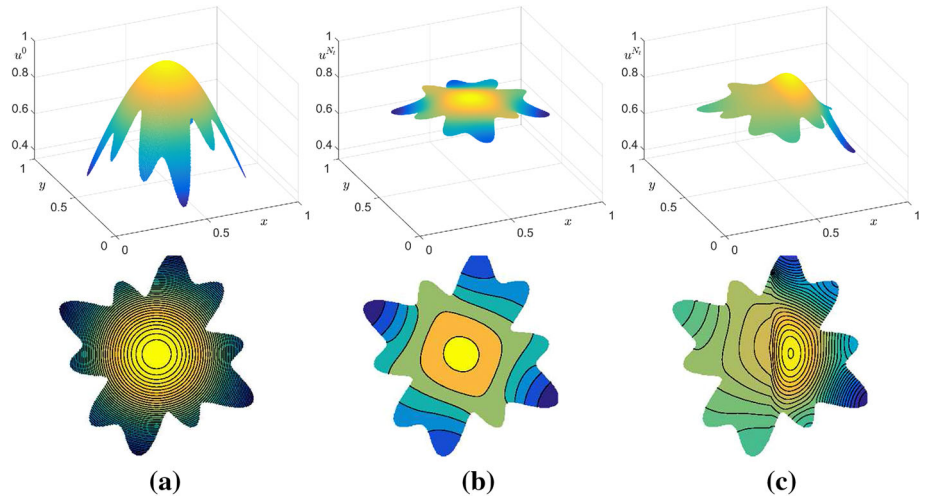

Fig. 5 **a** Initial condition. Numerical solutions at $T = 0.2$ by the proposed method with **b** constant and **c** piecewise constant diffusion coefficients. *Bottom row* represents the contour plot of numerical solutions

Table 7 Total mass change for numerical solution with respect to tol

	Constant diffusion		Piecewise constant diffusion	
tol	GS	GS with projection	GS	GS with projection
1.00e–06	7.28987e–05	3.88578e–16	1.88394e–04	8.32667e–17
1.00e–07	1.49888e–05	1.94289e–16	3.28959e–05	5.55112e–17
1.00e–08	2.46454e–06	1.00000e–17	4.60941e–06	8.32667e–17
1.00e–09	3.65534e–07	1.38778e–16	6.07127e–07	1.11022e–16
1.00e–10	5.11809e–08	5.55112e–17	7.72303e–08	8.32667e–17

Here, we use $h = 0.0025$, $\Delta t = 0.001$, and $T = 0.2$

complex domain is defined by zero contour $\phi(x, y) = 0$ of a level set function as

$$\phi(x, y) = \sqrt{(x - 0.5)^2 + (y - 0.5)^2} - 0.3 - 0.1\cos(7\theta)\sin(3\theta),$$

where $\theta = \tan^{-1}\left(\frac{y-0.5}{x-0.5}\right)$.

Figure 5 shows the initial condition and numerical solutions at $T = 0.2$ by the proposed method when the diffusion coefficient is constant or piecewise constant. Table 7 represents the results about change of total mass with respect to tolerance, tol. Similar to the previous test, we calculate the total mass change $M = \left|\sum_i \sum_j u_{ij}^0 - \sum_i \sum_j u_{ij}^{N_t}\right|$. As shown in Table 7, the proposed projection method preserves the total mass under the rounding errors with any tolerance, tol. However, the deviations of total mass in GS scheme are getting larger as tol is getting larger.

3.6 The Cahn–Hilliard Equation

To verify the proposed conservative scheme, we consider the Cahn–Hilliard (CH) equation. The equation was originally proposed by Cahn and Hilliard [1,2] to model binary alloys

and has subsequently been adopted to model many other physical situations such as phase transitions and interface dynamics in multiphase fluid [19]. There has been many numerical studies of the CH equation using iterative methods [19,26–28]. In this section, we take a two-dimensional CH equation as follows:

$$c_t(x, y, t) = \Delta\mu(x, y, t), \tag{15}$$

$$\mu(x, y, t) = c^3(x, y, t) - \frac{3}{2}c^2(x, y, t) + \frac{1}{2}c(x, y, t) - \epsilon^2\Delta c(x, y, t), \tag{16}$$

for $(x, y) \in \Omega = (0, 1) \times (0, 1)$ and $t > 0$. The boundary conditions for the CH equation are $\partial c/\partial \mathbf{n} = \partial\mu/\partial\mathbf{n} = 0$ on $\partial\Omega$, where $\mathbf{n}$ is the normal unit vector pointing out of Ω. The solution $c(x, y, t)$ of the CH Eqs. (15) and (16) has the property that the total mass $\int_\Omega c \, d\mathbf{x}$ is conserved, that is,

$$\frac{d}{dt}\int_\Omega c \, d\mathbf{x} = \int_\Omega \frac{\partial c}{\partial t} \, d\mathbf{x} = \int_\Omega \Delta\mu \, d\mathbf{x} = \int_{\partial\Omega} \frac{\partial\mu}{\partial\mathbf{n}} \, ds = 0.$$

3.6.1 Numerical Solution with the Projection Method

We discretiize the given domain $\Omega = (0, 1) \times (0, 1)$ with a uniform space step $h = 1/N_x = 1/N_y$ and a time step $\Delta t = T/N_t$. Here, N_x, N_y, and N_t are the number of grid points in the x-, y-, and t-direction, respectively. Let c_{ij}^n be an approximation of $c(x_i, y_j, n\Delta t)$. By applying the convex splitting scheme which is originally developed by Eyre [29], the discretization of the CH Eqs. (15) and(16) is given by

$$\frac{c_{ij}^{n+1} - c_{ij}^n}{\Delta t} = \Delta_d\mu_{ij}^{n+1}, \tag{17}$$

$$\mu_{ij}^{n+1} = f(c_{ij}^{n+1}) - \frac{1}{4}c_{ij}^n - \epsilon^2\Delta_d c_{ij}^{n+1}. \tag{18}$$

Here, the discrete Laplacian operator is defined by $\Delta_d c_{ij} = (c_{i+1,j} + c_{i-1,j} + c_{i,j+1} + c_{i,j-1} - 4c_{ij})/h^2$ and $f(c_{ij}^{n+1}) = (c_{ij}^{n+1})^3 - \frac{3}{2}(c_{ij}^{n+1})^2 + \frac{3}{4}c_{ij}^{n+1}$. For all n, the discrete boundary condition is defined as

$$c_{0j}^n = c_{1j}^n, \; c_{N_x+1,j}^n = c_{N_x,j}^n, \; c_{i0}^n = c_{i1}^n, \; c_{i,N_y+1}^n = c_{i,N_y}^n,$$
$$\mu_{0j}^n = \mu_{1j}^n, \; \mu_{N_x+1,j}^n = \mu_{N_x,j}^n, \; \mu_{i0}^n = \mu_{i1}^n, \; \mu_{i,N_y+1}^n = \mu_{i,N_y}^n,$$

where $i = 1, 2, \ldots, N_x$ and $j = 1, 2, \ldots, N_y$. The resulting non-linear system of the CH Eqs. (17) and (18) can be solved using several iterative methods. It is well known that classical iterative methods, such as Gauss–Seidel, Jacobi, SOR, or CG, converge very slowly for solving large linear systems [30]. On the other hand, a multigrid method damps strongly the oscillatory error components [31] and its solution is obtained in $O(N)$ time, where N is the total number of grid points [32]. The multigrid method is widely used for solving the CH equation in the two-dimensional [19,33], three-dimensional [34], and axisymmetric [35] domains. In this test, we use a multigrid method as the numerical solver. See [36,37] for more details about the multigrid method.

Given c^n and μ^n, we calculate c^{n+1} and μ^{n+1} using the multigrid method. Let $c^{n+1,m}$ and $c^{n+1,m+1}$ be the approximations of c^{n+1} before and after a V-cycle of the multigrid method. Starting from initial values $c^{n+1,0} = c^n$ and $\mu^{n+1,0} = \mu^n$, we iterate the multigrid cycle until the discrete l_2-norm of the two consecutive approximations is less than a given tolerance, i.e.,

$\|c^{n+1,m+1} - c^{n+1,m}\|_2 < tol$. By the Gauss–Seidel iteration in a V-cycle of the multigrid method, for $m = 1, 2, \cdots$, we iterate the following algorithm

$$\begin{aligned}
&\frac{1}{\Delta t} c_{ij}^{n+1,m,s+1} + \frac{4}{h^2}\mu_{ij}^{n+1,m,s+1} \\
&\quad = \frac{c_{ij}^{n}}{\Delta t} + \frac{\mu_{i+1,j}^{n+1,m,s} + \mu_{i-1,j}^{n+1,m,s+1} + \mu_{ij+1}^{n+1,m,s} + \mu_{ij-1}^{n+1,m,s+1}}{h^2}, \\
&\quad - \left(\frac{\partial f}{\partial c}(c_{ij}^{n+1,m,s}) + \frac{4\epsilon^2}{h^2}\right) c_{ij}^{n+1,m,s+1} + \mu_{ij}^{n+1,m,s+1} = -\frac{c_{ij}^{n}}{4} + f(c_{ij}^{n+1,m,s}) \\
&\quad - \frac{\partial f}{\partial c}(c_{ij}^{n+1,m,s}) c_{ij}^{n+1,m,s} - \frac{\epsilon^2}{h^2}\left(c_{i+1,j}^{n+1,m,s} + c_{i-1,j}^{n+1,m,s+1} + c_{ij+1}^{n+1,m,s} + c_{ij-1}^{n+1,m,s+1}\right).
\end{aligned}$$

Here, the indices s and $s+1$ denote the current and the new approximations during a Gauss–Seidel iteration. Since $f(c^{n+1,m,s+1})$ is nonlinear with respect to $c^{n+1,m,s+1}$, we linearize $f(c^{n+1,m,s+1})$ at $c^{n+1,m,s}$, i.e.,

$$\left(c^{n+1,m,s+1}\right)^3 \approx \left(c^{n+1,m,s}\right)^3 + 3\left(c^{n+1,m,s}\right)^2\left(c^{n+1,m,s+1} - c^{n+1,m,s}\right).$$

If the solution $c^{n+1,m}$ converges under the given tolerance, then we obtain new solution c^{n+1} which is defined by $c^{n+1} = c^{n+1,m}$. Now, we denote the residual error for c_{ij}^{n+1} with respect to Eq. (17), i.e,

$$r_{ij} = \frac{c_{ij}^{n+1} - c_{ij}^{n}}{\Delta t} - \frac{\mu_{i+1,j}^{n+1} + \mu_{i-1,j}^{n+1} + \mu_{ij+1}^{n+1} + \mu_{ij-1}^{n+1} - 4\mu_{ij}^{n+1}}{h^2} \tag{19}$$

for $i = 1, 2, \cdots, N_x$ and $j = 1, 2, \cdots, N_y$. We also define the discrete l_2-norm and the maximum norm as

$$\mathbf{e}_2 = \sqrt{\frac{1}{N_x N_y}\sum_{i=1}^{N_x}\sum_{j=1}^{N_y} r_{ij}^2} \quad \text{and} \quad \mathbf{e}_\infty = \max_{\substack{1\le i\le N_x \\ 1\le j\le N_y}} |r_{ij}|.$$

With the numerical solution c^* obtained from an iterative method, we get the conservative numerical solution c^{n+1} by the projection method as follows:

$$c^{n+1} = c^n + \Delta t \Delta_d \mu^*,$$

where we use the homogeneous Neumann boundary conditions: $\mu_{0j}^* = \mu_{1j}^*$, $\mu_{N_x+1,j}^* = \mu_{N_x,j}^*$, $\mu_{i0}^* = \mu_{i1}^*$, and $\mu_{i,N_y+1}^* = \mu_{i,N_y}^*$.

3.6.2 Convergence Test

We have the convergence tests with the l_2-norm and maximum norm of the errors to show accuracy of the proposed scheme. However, it is well known that the CH equation does not have non-trivial analytic solutions. Therefore, in this paper, we use a reference solution on behalf of the analytic solution. The initial condition is $c(x, y, 0) = 0.25(1 - \cos(2\pi x))(1 - \cos(2\pi y))$ in a domain $\Omega = (0, 1) \times (0, 1)$. To compute the spatial accuracy of the proposed projection method, we run simulations on increasingly finer grids, i.e., $h = 1/2^{n+1}$ for

Table 8 Maximum and l_2 norm of the errors and its convergence rates for numerical solution with respect to h

Here, we use $\Delta t = 0.0001$ and $T = 0.01$

h	$\|\mathbf{e}\|_\infty$	Rate	$\|\mathbf{e}\|_2$	Rate
1/32	9.60903e-4		3.38298e-4	
1/64	2.49461e-4	1.94557	8.57879e-5	1.97945
1/128	6.52057e-5	1.93575	2.20716e-5	1.95858
1/256	1.87569e-5	1.79758	6.26686e-6	1.81638

Table 9 Maximum and l_2 norm of the errors and its convergence rates for numerical solution with respect to Δt

Here, we use $h = 1/2048$ and $T = 0.1$

h	$\|\mathbf{e}\|_\infty$	Rate	$\|\mathbf{e}\|_2$	Rate
0.010000	1.6592e-2		1.1732e-2	
0.005000	7.9494e-3	1.0615	5.6213e-3	1.0615
0.002500	3.8738e-3	1.0371	2.7393e-3	1.0371
0.001250	1.9102e-3	1.0200	1.3508e-3	1.0200
0.000625	9.4872e-4	1.0096	6.7089e-4	1.0096

$n = 5, 6, \ldots, 10$. The reference solution is obtained using $h = 1/2048$. For each case, the calculation is run up to time $T = 0.01$ with the time step $\Delta t = 0.0001$ and a fixed $\epsilon = 0.00375$.

Table 8 shows the errors and rates of spatial convergence. The results suggest that the proposed scheme is second-order accurate in space as we expect from the discretization.

Next, we perform the temporal convergence test with $h = 1/256$. For each case, the calculation is run up to time $T = 0.01$ with the time step $\Delta t = 0.001/2^n$ for $n = 5, 6, \ldots, 10$. Also, we use the reference solution with $\Delta t = 0.0001$ and $h = 1/2048$. The results are shown in Table 9. As we expected, the proposed scheme is first-order accurate in time.

3.6.3 Conservation of Total Mass

In this section, we test the conservative property of numerical solution of the CH equation. We use the gradient energy coefficient $\epsilon = 0.00375$, spatial step size $h = 1/256$, temporal step size $\Delta t = 0.1$, the computational domain $\Omega = (0, 1) \times (0, 1)$, and the total time $T = 100$. The numerical initial condition is $c(x, y, 0) = 0.25(1 - \cos(2\pi x))(1 - \cos(2\pi y))$.

Figure 6 represents the total mass with respect to time. To show the conservative property of the proposed projection scheme, we also add the total mass (solid line) of the numerical solution by the projection scheme with $tol = 1.0\text{e}{-3}$. As tolerance tol decreases, we can see that the total mass of the numerical solution is conserved. However, the numerical solution by the projection scheme is always conserved as 0.25 even if we use $tol = 1.0\text{e}{-3}$.

Next, Fig. 7 shows the CPU time with respect to tolerance in both cases; without correction and with correction. As shown in Fig. 7, the additional computational cost with correction is negligible. This result implies the efficiency of the proposed projection method for conservative problems. That is, we can achieve the conservation of the total mass with the larger tolerance levels than the required one.

In addition, our proposed numerical scheme has good performance with low cost. To show this, we represent the difference of reference solution $\bar{c}$ and numerical solution c with

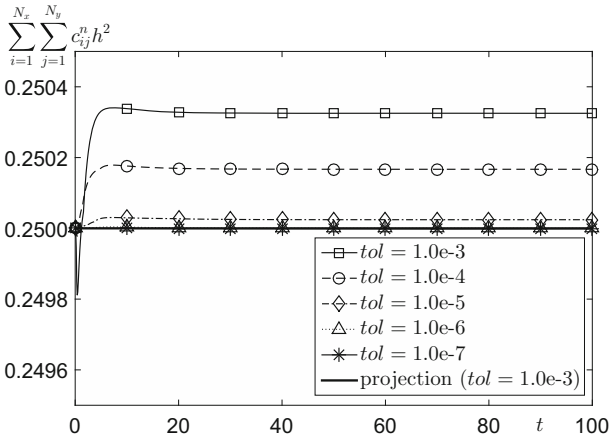

Fig. 6 Total mass with respect to the tolerance tol. Here, we use $h = 1/128$, $\Delta t = 0.1$, and $T = 100$

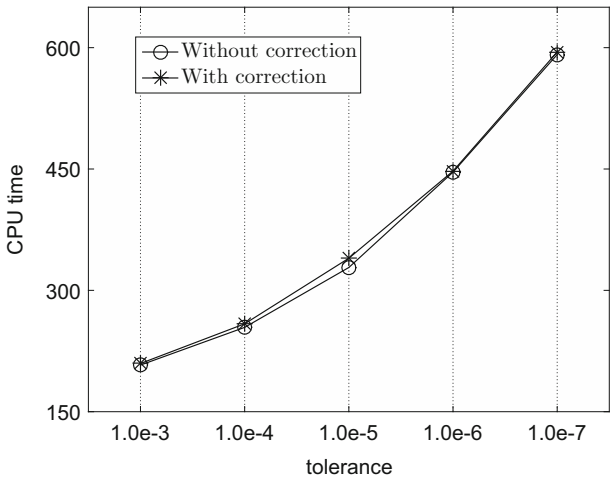

Fig. 7 Comparison of CPU time without and with correction procedure

$tol = 1.0\text{e}{-3}$ in Fig. 8. Here, the reference solution $\bar{c}$ is defined as the numerical solution without correction scheme by using $tol = 1.0\text{e}{-7}$ and two numerical solutions are obtained with and without correction scheme.

Figure 9a, b represent the relative error between the reference solution $\bar{c}$ and numerical solution c without and with correction schemes, respectively. As shown in Fig. 8, we obtain similar behaviors except for the amplitude of the value.

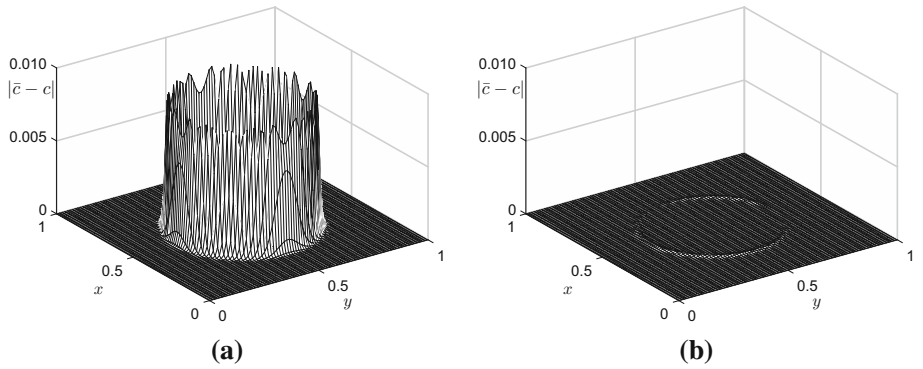

Fig. 8 Absolute difference between reference solution $\bar{c}$ and numerical solution c **a** without and **b** with correction scheme. Here, we use the reference value (without correction and $tol = 1.0\text{e}{-7}$) and two numerical values ($tol = 1.0\text{e}{-3}$) at time $T = 100$

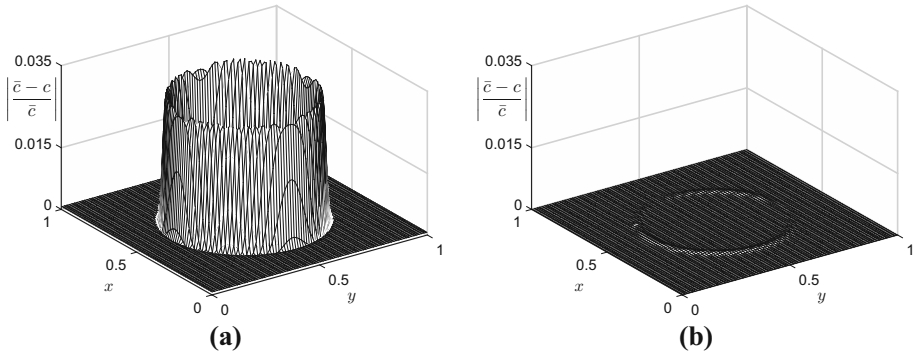

Fig. 9 Relative error between reference solution $\bar{c}$ and numerical solution c **a** without and **b** with correction scheme. Here, we use the reference value (without correction and $tol = 1.0\text{e}{-7}$) and two numerical values ($tol = 1.0\text{e}{-3}$) at time $T = 100$

4 Conclusion

In this paper, we proposed a conservative projection method for the conservative discretizations of parabolic PDEs. The proposed scheme conserves the conservative quantity within rounding errors. The new algorithm can be used with pre-existing iterative algorithms because we just add one explicit update. To verify the our proposed scheme, we test two problems: the heat equation and the Cahn–Hilliard equation. Some directions for future research include applying the proposed projection scheme to various conservative parabolic equations.

Acknowledgements The first author (D. Jeong) was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (2014R1A6A3A01009812). The corresponding author (J.S. Kim) was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIP) (NRF-2014R1A2A2A01003683). The authors greatly appreciate the reviewers for their constructive comments and suggestions, which have improved the quality of this paper.

References

1. Cahn, J.W.: On spinodal decomposition. Acta Metall. **9**, 795–801 (1961)
2. Cahn, J.W., Hilliard, J.E.: Free energy of a nonuniform system. I. Interfacial free energy. J. Chem. Phys. **28**(2), 258–267 (1958)
3. Lee, D., Huh, J.Y., Jeong, D., Shin, J., Yun, A., Kim, J.: Physical, mathematical, and numerical derivations of the Cahn–Hilliard equation. Comp. Mater. Sci. **81**, 216–225 (2014)
4. Li, X., Zhang, L., Wang, S.: A compact finite difference scheme for the nonlinear Schrödinger equation with wave operator. Appl. Math. Comput. **219**, 3187–3197 (2012)
5. Fei, Z., Perez-Garcia, V.M., Vazquez, L.: Numerical simulation of nonlinear Schrödinger systems: a new conservative scheme. Appl. Math. Comput. **71**, 165–177 (1995)
6. Chang, Q., Xu, L.: A numerical method for a system of generalized nonlinear Schrödinger equations. J. Comput. Math. **4**, 191–199 (1986)
7. Chang, Q., Wang, G.: Multigrid and adaptive algorithm for solving the nonlinear Schrödinger equation. J. Comput. Phys. **88**, 362–380 (1990)
8. Chang, Q., Jia, E., Sun, W.: Difference schemes for solving the generalized nonlinear Schrödinger equation. J. Comput. Phys. **148**, 397–415 (1999)
9. Matsuo, T., Furihata, D.: Dissipative or conservative finite-difference schemes for complex-valued nonlinear partial differential equations. J. Comput. Phys. **171**(2), 425–447 (2001)
10. Chang, Q., Wang, G., Guo, B.: Conservative scheme for a model of nonlinear dispersive waves and its solitary waves induced by boundary notion. J. Comput. Phys. **93**, 360–375 (1991)
11. Zhang, F., Vazquez, L.: Two energy conserving numerical schemes for the Sine–Gordon equation. Appl. Math. Comput. **45**, 17–30 (1991)
12. Wong, Y.S., Chang, Q., Gong, L.: An initial-boundary value problem of a nonlinear Klein–Gordon equation. Appl. Math. Comput. **84**, 77–93 (1997)
13. Chang, Q., Jiang, H.: A conservative scheme for the Zakharov equation. J. Comput. Phys. **113**, 309–319 (1994)
14. Chang, Q., Guo, B., Jiang, H.: Finite difference method for generalized Zakharov equation. J. Comput. Phys. **113**, 309–319 (1994)
15. Furihata, D.: A stable and conservative finite difference scheme for the Cahn–Hilliard equation. Numer. Math. **87**, 675–699 (2001)
16. Choo, S.M., Chung, S.K.: Conservative nonlinear difference scheme for the Cahn–Hilliard equation. Comput. Math. Appl. **36**, 31–39 (1998)
17. Furihata, D., Matsuo, T.: A stable, convergent, conservative and linear finite difference scheme for the Cahn–Hilliard equation. Jpn. J. Ind. Appl. Math. **20**, 65–85 (2003)
18. De Mello, E.V.L., da Silveira Filho, O.T.: Numerical study of the Cahn–Hilliard equation in one, two and three dimensions. Phys. A. **347**, 429–443 (2005)
19. Kim, J., Kang, K., Lowengrub, J.: Conservative multigrid methods for Cahn–Hilliard fluids. J. Comput. Phys. **193**, 511–543 (2004)
20. Burden, R.L., Faires, J.D.: Numerical Analysis, 9th edn. Brooks Cole, Boston (2011)
21. Hackbusch, W.: Multi-grid Methods and Applications. Springer, Berlin (1985)
22. Briggs, W.L., McCormick, S.F.: A Multigrid Tutorial. SIAM, Philadelphia (2000)
23. Trottenberg, U., Oosterlee, C.W., Schuller, A.: Multigrid. Academic press, London (2000)
24. Wesseling, P.: Introduction to Multigrid Methods. Wiley, Chichester (1992)
25. Saad, Y., Schultz, M.H.: GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems. SIAM J. Sci. Statist. Comput. **7**, 856–869 (1986)
26. Elliott, C.M.: The Cahn–Hilliard Model for the Kinetics of Phase Separation Number 88 in International Series of Numerical Mathematics. Birkhauser, Basel (1989)
27. Cahn, J.W., Chow, S.N., van Vleck, E.S.: Spatially iscrete nonlinear diffusion equations. Rocky Mt. J. Math. **25**(1), 87–118 (1995)
28. Maier, R.S., Rath, W., Petzold, L.R.: Parallel solution of large-scale differential-algebraic systems. Concurr. Comput.-Pract. E. **7**(8), 795–822 (1995)
29. Eyre, D.J.: Unconditionally gradient stable time marching the Cahn–Hilliard equation. In: MRS Proceedings, **529** pp. 39. Cambridge University Press (1998)
30. Gupta, M.M., Zhang, J.: High accuracy multigrid solution of the 3D convection-diffusion equation. Appl. Math. Comput. **113**(2), 249–274 (2000)
31. Guillet, T., Teyssier, R.: A simple multigrid scheme for solving the Poisson equation with arbitrary domain boundaries. J. Comput. Phys. **230**(12), 4756–4771 (2011)
32. Fulton, S.R., Ciesielski, P.E., Schubert, W.H.: Multigrid methods for elliptic problems: a review. Mon. Weather Rev. **114**(5), 943–959 (1986)

33. Lee, C., Jeong, D., Shin, J., Li, Y., Kim, J.: A fourth-order spatial accurate and practically stable compact scheme for the Cahn–Hilliard equation. Phys. A. **409**(1), 17–28 (2014)
34. Kim, J.: Three-dimensional numerical simulations of a phase-field model for anisotropic interfacial energy. Commun. Korean. Math. Soc. **22**(3), 453–464 (2007)
35. Kim, J.: A diffuse-interface model for axisymmetric immiscible two-phase flow. Appl. Math. Comput. **160**(2), 589–606 (2005)
36. Briggs, W.: A Multigrid Tutorial. SIAM, Philadelphia (1977)
37. Trottenberg, U., Oosterlee, C.W., Schüller, A.: Multigrid. Academic press, London (2000)
38. Shin, J., Jeong, D., Kim, J.: A conservative numerical method for the Cahn–Hilliard equation in complex domains. J. Comput. Phys. **230**(19), 7441–7455 (2011)