Applied Mathematics and Computation 484 (2025) 128993

Contents lists available at ScienceDirect

Applied Mathematics and Computation

journal homepage: www.elsevier.com/locate/amc

Full Length Article

Solving Allen-Cahn equations with periodic and nonperiodic boundary conditions using mimetic finite-difference operators

Saulo Orizaga [a], Gilberto González-Parra [a,b,*], Logan Forman [a], Jesus Villegas-Villanueva [a]

[a] Department of Mathematics, New Mexico Tech, Socorro, NM, USA
[b] Instituto de Matemática Multidisciplinar, Universitat Politècnica de València, Valencia, Spain

ARTICLE INFO

Keywords:
Allen-Cahn
Heat equation
Periodic boundary conditions
Non-periodic boundary conditions
Mimetic operator
Finite difference

ABSTRACT

In this paper, we investigate and implement a numerical method that is based on the mimetic finite difference operator in order to solve the nonlinear Allen–Cahn equation with periodic and non-periodic boundary conditions. In addition, we also analyze the performance of this mimetic-based method by using the classical heat equation with a variety of boundary conditions. We assess the performance of the mimetic-based numerical method by comparing the errors of its solutions with those obtained by a classical finite difference method and the pdepde built-in Matlab function. We compute the errors by using the exact solutions when they are available or with reference solutions. We adapt and implement the mimetic-based numerical method by using the MOLE (Mimetic Operators Library Enhanced) library that includes some built-in functions that return representations of the curl, divergence and gradient operators, in order to deal with the Allen-Cahn and heat equations. We present several results with regard to errors and numerical convergence tests in order to provide insight into the accuracy of the mimetic-based numerical method. The results show that the numerical method based on the mimetic difference operator is a reliable method for solving the Allen–Cahn and heat equations with periodic and non-periodic boundary conditions. The numerical solutions generated by the mimetic-based method are relatively accurate. We also proposed a new method based on the mimetic finite difference operator and the convexity splitting approach to solve Allen-Cahn equation in 2D. We found that, for small time step sizes the solutions generated by the mimetic-based method are more accurate than the ones generated by the pdepe Matlab function and similar to the solutions given by a finite difference method.

1. Introduction

Allen and Cahn originally proposed the Allen–Cahn (AC) equation to explain the generation and migration of antiphase grain boundaries in crystalline materials [1]. The Allen–Cahn equation is an important model in the materials science literature [2,3]. It is a non-linear parabolic partial differential equation (PDE) that reads

* Corresponding author at: Department of Mathematics, New Mexico Tech, Socorro, NM, USA.
E-mail addresses: saulo.orizaga@nmt.edu (S. Orizaga), Gilberto.Gonzalezparra@nmt.edu (G. González-Parra), logan.forman@student.nmt.edu (L. Forman), jesus.villegas-villanueva@student.nmt.edu (J. Villegas-Villanueva).

https://doi.org/10.1016/j.amc.2024.128993
Received 28 November 2023; Received in revised form 17 July 2024; Accepted 30 July 2024

Available online 9 August 2024

$$u_t = \Delta u - \frac{1}{\epsilon^2} f(u). \tag{1}$$

The main goal of this paper is to solve the Allen–Cahn equation using mimetic finite difference approach since to the best of our knowledge this has not been done and it can provide new insights about numerically solving the Allen–Cahn equation. One inhomogeneous two-component material's concentration is used to represent the Allen–Cahn system using the order parameter u, which is a scalar function for one phase. Values $u \pm 1$ are corresponding to the two distinct phases. Within the interfacial area, the order parameter u exhibits a constant yet sharp variation from one phase to the next across the phase interface. The thickness of the interfacial region is directly related to a tiny, positive constant called ϵ.

Although it originated in material science, the Allen–Cahn equation is now widely utilized to model non-conservative events in many different domains. A Lagrange multiplier is added to the Allen–Cahn equation in order to conserve the total quantity of the closed system. Numerous fields have made use of this modified Allen–Cahn equation [4,5]. This phase-field equation has frequently been examined in fluids, biological processes, and in other topics. Furthermore, there are numerous works pertaining to the Allen–Cahn type equations [6–8]. The Allen–Cahn equation has been used to model the inter-facial dynamics of two-phases fluids [9].

Multiphase flows are relevant in a variety of natural and industrial applications [5]. The basic dynamics of multiphase flows can be studied by laboratory experiments that provide little understanding of the controlling fluid mechanics. Moreover, when the number of phases increases (the number of interfaces increases) it is difficult to observe the flow experimentally. Thus, a variety of Allen–Cahn type models are necessary [10,11]. For instance, in [10] a conservative Allen–Cahn model was used instead of the Cahn–Hilliard model, in order to reduce the computations and increase the applicability of the model. Therefore, numerical schemes to numerically solve Allen–Cahn mode are necessary.

The use of numerical methods to explore the dynamics of real world processes has increased over the last decades. The numerical simulations enable further insight into many aspects that cannot be done by physical investigations due to complexity, in-feasibility and costs. Numerical solutions oftentimes are quicker to obtain and less expensive than physical experiments. Multiphase flows appear in many real world processes and many works have been devoted to study them. In particular, Allen-Cahn equation and its variants have been studied and numerically solved in order to better understand the processes that this equation represents. For instance, in [12] the Allen–Cahn equation with non-periodic boundary conditions was numerical solved by using Chebyshev spectral method for the space discretization and using the exponential time differencing 4th-order Runge-Kutta for the time space. In [13], a local discontinuous Galerkin method was developed to solve the Allen-Cahn equation with periodic boundary conditions. A first order semi-implicit numerical scheme based on the convex splitting principle of the discrete Allen-Cahn energy was constructed. The local discontinuous Galerkin method has also been applied to periodic boundary conditions for the Kortheweg-de Vries [14]. Oftentimes PDEs problems with non-periodic boundary conditions are more difficult to solve and Fourier based methods require modifications to handle this type of problems [15].

One of the main aims of this work is to investigate and implement a numerical method that is based on the mimetic finite-difference operator in order to solve the nonlinear Allen–Cahn equation with periodic and non-periodic boundary conditions. The mimetic finite-difference method has been used more often during the last decade [16,17]. The early works in mimetic methods can be traced back to the last century [18,19]. The fully mimetic methods construct a discrete calculus that mimics the properties of the continuum one. The high-order mimetic finite difference idea is related in some way to the original work about the summation by parts method [16,20]. Although the main motivation of the paper is to explore and analyze the suitability of the mimetic method to solve the Allen-Cahn equation, we also assessed the accuracy of the pdepe built-in Matlab function for solving the Allen-Cahn equation. To the best of our knowledge the mimetic finite-difference operator method has not been applied to phase field models and in particular to the Allen-Cahn equation. We adapt and implement the proposed mimetic-based numerical method by using the MOLE (Mimetic Operators Library Enhanced) that includes some built-in functions in order to deal with the Allen-Cahn and heat equations [17,16,21]. This MOLE library includes some examples for some particular PDEs. However, for some of the PDEs problems that we present in this study we need to make modifications on the discretization in order to be able to adapt to the specific boundary and initial conditions of the PDEs problems that we solve in this study. These adaptations need careful attention in order to keep the consistency of the mimetic-based numerical method. Moreover, there are a variety of boundary and initial conditions that can be taken, but each one requires a different discretization adaptation in order to be able to implement the MOLE library [21]. We test this mimetic-based method by solving the Allen–Cahn equation for different time step sizes in order to observe its order of convergence. In addition, we also analyze the performance of this method by using the classical heat equation with a variety of boundary conditions. We assess the performance of the mimetic-based numerical method by comparing the errors of its solutions with those obtained by a classical finite difference method and the pdepe built-in Matlab function [22,23]. The pdepe built-in Matlab function has been used to solve mathematical models of brown stock washing problems where a system of governing partial differential equations arise [23]. In [24] the authors compared the reliability of the pdepe built-in Matlab function by solving one dimensional steady-state advection dispersion reaction equations.

In our study, we compute the errors of the numerical solutions by using the exact solutions when they are available or with reference solutions obtained with a very refined mesh. We present several results with regard to errors and numerical convergence in order to provide insight into the effectiveness and accuracy of the mimetic-based numerical method. This method has been used previously to numerically solve the wave equation, convection-diffusion equation and recently the Navier-Stokes equation [17,25,26].

It has been mentioned that the mimetic finite difference operator method discretizes the continuum theory of the problem. Oftentimes, this discretization involves a physical conservation or constitutive law. The advantage of this is that it allows to resemble approximately the behavior of the real word phenomenon that is associated with the continuum problem. Thus, the mimetic discretization enables tackling many challenging problems that are based on PDEs [27]. The order of the mimetic-based numerical method can

be increased and it has been done in a previous work [16]. In [28] a mimetic finite difference scheme for solving the acoustic wave equation is presented. The authors combined a leapfrog approximation in time and second order tensor mimetic discretizations in space to create an explicit scheme. The authors concluded that the proposed scheme performs better than standard finite difference schemes that use ghost points. One of the motivations of this work is to address one open question that was presented in [28]. The authors were uncertain if the improvement of the stability condition achieved by the mimetic method could be also obtained in other time dependent PDEs. Finally, we would like to point out that methods based on the general mimetic idea have been called with different names [20,18,19]. One important aspect is that the mimetic operators of higher order have been developed recently and this is a main advantage of the MOLE library [17,16,21].

The layout of this paper is as follows. In Section 2 we present the numerical methods that we used in this study to numerically solve the Allen-Cahn and heat equations. Numerical solutions and errors are presented in Section 3. In Section 4 we present numerical solutions to the Allen–Cahn equation using random initial conditions which models the problem of the mixing of two materials. Finally, conclusions are presented in Section 5.

2. Mathematical models and methods

The Allen-Cahn free energy is given in the form [1],

$$\mathcal{E}(u) = \int_{\Omega} \left(\frac{1}{2}|\nabla u|^2 + \frac{1}{\epsilon^2} F(u) \right) dx \tag{2}$$

where $F(u)$ is a symmetric double well potential given by $F(u) = \frac{1}{4}(u^2-1)^2$. Here, $\epsilon > 0$ is a parameter that defines the thickness of the interface that separates the pure faces in a binary mixture. The Allen-Cahn equation can be understood as the gradient flow with respect to the inner product space in L^2 of the Allen-Cahn free energy [29]. The Allen-Cahn equation in 2D reads

$$u_t = \Delta u - \frac{1}{\epsilon^2} f(u) \tag{3}$$

where $f(u) = u^3 - u$. The solution for the Allen-Cahn equation will evolve into configurations that will approach the minima ($u \approx +1, u \approx -1$) and those solution states will be separated by an interface of thickness ϵ. At the continuous level, the Allen-Cahn equation obeys the energy decreasing property $\frac{d\mathcal{E}(u)}{dt} \leq 0$ for all time t [29]. In the design of computational approaches for the Allen-Cahn equation it is of high importance to construct numerical methods that preserve the energy decreasing property, $\mathcal{E}(U^{n+1}) \leq \mathcal{E}(U^n)$.

2.1. Finite difference scheme

In this subsection we present the finite difference scheme that is implemented in this work in order to solve the heat equation, heat equation with lateral heat loss and the Allen-Cahn (AC) equation. We consider the general nonlinear parabolic formulation in 1D which reads

$$u_t = u_{xx} + m(u), \tag{4}$$

where $m(u)$ is a non-linear function and it will be specified in the later sections. We discretize $\Omega = [a, b]$ with equally space points except at the edges giving the following computational grid $x = [x_0, x_1, x_2, ..., x_{N-1}, x_N]$. The spacing of the edge grid points is defined by $dx/2 = x_1 - x_0 = x_N - x_{N-1}$ and interior grid points are spaced with $dx = x_{i+1} - x_i$ for $i = 1, ..., N-2$. We can then apply standard finite difference approximations to the above equation. We consider a semi-implicit formulation that uses a second order central finite difference in space and first order approximation in time leading to

$$\frac{U_i^{n+1} - U_i^n}{h} = \frac{U_{i-1}^{n+1} - 2U_i^{n+1} + U_{i+1}^{n+1}}{dx^2} + m(U_i^n), \tag{5}$$

where $U_i^n \approx u(t_n, x_i)$ and h is the timestep. The scheme given above is applicable in all the interior nodes but due to the $dx/2$ spacing at the edges of the computational domain a slightly different formula needs to be applied near the end points, namely

$$u_{xx}(x_i) \approx \frac{8/3U_{i-1/2}^{n+1} - 4U_i^{n+1} + 4/3U_{i+1}^{n+1}}{dx^2}, \tag{6}$$

$$u_{xx}(x_i) \approx 4\frac{U_{i-1/2}^{n+1} - 2U_i^{n+1} + U_{i+1/2}^{n+1}}{dx^2}, \tag{7}$$

where the above are to be used for the near left and near right end points, respectively. Using equations (2)-(5) gives the finite difference formulation with periodic boundary conditions for the non-linear parabolic problem which can be written as

$$AU^{n+1} = B \tag{8}$$

where A is an $n \times n$ matrix, B is $n \times 1$ vector and U^{n+1} is the $n \times 1$ vector containing the numerical solution $U^{n+1} = [U_1, U_2, ..., U_N]$ at the grid points $x = [x_1, x_2, ..., x_N]$. A formal last step is to pad the numerical solution, since we have periodic boundary conditions, by letting $U_0 = U_N$. The numerical solution is obtained at each time level ($t_n, n = 0, 1, 2, ...$) by solving equation (7).

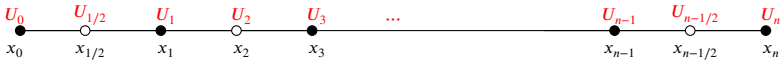

Fig. 1. One dimensional staggered grid.

For the case of no-flux boundary conditions, we start with the same computational grid as in the case of periodic boundary conditions but we add two ghost points at a distance of $dx/2$ from the edges, x_{-1} and x_{N+1} which extends our computational grid to $x = [x_{-1}, x_0, x_1, x_2, ..., x_{N-1}, x_N, x_{N+1}]$ We introduce second order approximations of the first derivative in space at the boundaries in order to match the second order derivative approximations implemented in the interior of the computational domain

$$u_x \approx \frac{U_1^n - U_{-1}^n}{dx}, \tag{9}$$

$$u_x \approx \frac{U_{N+1}^n - U_{N-1}^n}{dx}. \tag{10}$$

Using the fact that $u_x = 0$ at the x_0 and x_N, we can define values for the ghost points $U_{-1}^n = U_1^n$ and $U_{N+1}^n = U_{N-1}^n$ at all time levels $n = 0, 1, ...$ It is then possible to use the scheme given by equation (2) for the case of no-flux boundary conditions which can be written as

$$CU^{n+1} = D \tag{11}$$

where C is an $(n+1)\times(n+1)$ matrix, D is a $(n+1)\times 1$ vector and U^{n+1} is the $(n+1)\times 1$ vector containing the numerical solution $U^{n+1} = [U_0, U_1, U_2, ..., U_N]$. A numerical solution is obtained at each time level $(t_n, n = 0, 1, ...)$ by solving equation (10). We note that the nonlinear parabolic equation reduces to the classic heat equation if $m(u) = 0$. For $m(u) = -\alpha u$, we obtain the heat equation with lateral heat loss and for $m(u) = \frac{1}{\epsilon^2} f(u)$ we obtain the Allen-Cahn equation.

2.2. Mimetic-based numerical method

In this section we present briefly some basic foundations of the mimetic-based numerical method. First we present the mimetic operators and then we introduce some basic aspects of the MOLE library that is used in this work to solve some PDEs problems with different conditions.

One advantage of the discrete mimetic operator is that it resembles its counterpart continuum operator. In addition, it satisfies conservation laws and it is easy to use. Furthermore, high-order approximations of divergence and gradient operators can be obtained [27,16].

2.2.1. Mimetic operators discretizations

In this section we focus on presenting the second order mimetic discretizations for the gradient and divergence operators since these are the ones that are required to solve the PDEs problems presented in this article. In particular, we are interested in the Laplacian or Laplace's operator since this appears in all the PDEs problems presented in this study. The crucial part is that the Laplacian can be written as the mixture of the gradient, ∇, and the divergence, $\nabla\cdot$ [27]. Thus, the general parabolic Eq. (4) can be written as

$$\nabla \cdot_t u = \nabla \cdot \nabla u + m(u), \tag{12}$$

where $\nabla\cdot_t$ represents the gradient operator in time [28]. Now, following the notation of the finite difference method presented in the previous section we can define N cells $[x_i, x_{i+1}]$ for $n = 0, 1, 2, \ldots N$. As above, we can name the endpoints of each of the cells as nodes. Note that we have special nodes x_0 and x_n and these can be referred to as the grid's boundary points or edge grid points [27,28]. Another important aspect of the discretization of the mimetic operators is that this is done over a staggered grid which might be uniform or not [27,28]. The use of staggered grids has been done in many PDEs problems related to fluid dynamics in order to increase the accuracy of the numerical solutions [30,31].

Following the finite difference method from the previous section we will focus on a one dimensional uniform staggered grid since we will use MOLE library with this type of grid for all the numerical tests where the mimetic-based numerical method is implemented. For the sake of easiness of exposition, a one-dimensional uniform staggered grid is depicted in Fig. 1. The white circles represent block's centers and the black circles are the block's edges. The one-dimensional mimetic discretization of the divergence operator is calculated at the block's center by standard central difference [28]. On the other hand, the discretization of the gradient operator is computed at the block's edges by central differences. Thus, the vector quantities are computed at the block's edges and scalar quantities at the block's centers. In other words, the divergence operator at the middle points (white circles) and the gradient operator at the edge points [17].

Let's present the basic ideas of the discretization of the divergence and gradient operators since those are the ones needed for numerically solving the PDEs problems presented in this study. Let's denote the mimetic divergence operator as D, and the mimetic gradient operator as G. Now, D and G are finite-dimensional linear operators, and therefore we can use matrices to write them in an explicit way [27].

Following the exposition presented in [27], we will first start with the discretization of the divergence operator. At the grid points we can define the divergence operator D with matrix dimensions $n\times(n+1)$. Defining the gradient operator G at the middle points

of the cells and at the edge grid points one gets that G can be represented as a matrix with dimension $(n+1)\times(n+2)$ [27]. Thus, we can obtain the Laplacian operator by using the product of the operators D and G, which results in a matrix of dimensions $n\times(n+2)$. This representation of the Laplacian operator maps u values evaluated at the middle points of the cells and at the edge grid points to the values at the middle of the cells. In [27] it was mentioned that the convergence of this particular mimetic gradient G at the edge grid points is not as good as in the interior grid points. For more information about the mimetic operators and some recent applications, we refer interested readers to the following references [17,27,28,32].

There are a variety of ways that can be used to discretize the divergence and gradient operators. It is possible to use discretization to increase the order of convergence and then we will have different stencils [33]. Thus, we can discretize the Laplacian operator using the mimetic based approach using different discretizations that have their own order of convergence. In this study, we deal with the heat equation and the Allen-Cahn equation which can be written in terms of the Laplacian operator and therefore in terms of the product of the operators D and G. Then, we can use different discretizations in order to solve these equations.

We will rely on the MOLE library which has already included these discretizations. However, in order to use the MOLE library to numerically solve the heat equation and Allen-Cahn equation with different boundary conditions, it requires some adaptations to satisfy the boundary conditions. In other words, we modified the MOLE library that has its own space discretization to adapt to the variety boundary conditions that we considered in this study. Next section is devoted to briefly explaining the MOLE library and its main features.

2.2.2. MOLE (mimetic operators library enhanced)

As we have explained in the previous section there are a variety of mimetic discretizations that can be done. In [16], it was presented high-order mimetic finite-difference operators that fulfill the extended Gauss theorem and improved the discretizations presented in [33]. By using these operators it was possible to keep the same order of accuracy in the interior and at the boundary. This was done by utilizing the well-known staggered grids and weighted inner products. The MOLE open-source library for numerically solving PDEs was recently developed using these mimetic finite-difference operators. There are several advantages of having this MOLE library. First, researchers interested in solving some particular PDEs problems can use it. Secondly, since MOLE is an open-source library it is possible that the users can make their own modifications in order to adapt the MOLE library to a variety of PDEs problems.

The MOLE library is available in C++ and MATLAB. This library has discrete versions of the classical operators curl, divergence and gradient. As explained in the previous section the discrete operators are written in terms of matrices and therefore MOLE returns a sparse matrix representation of the each operator. The authors of the MOLE library have implemented all these operators for one, two and three dimensions. In addition, for suitability the library is implemented in uniform and nonuniform staggered grids. The discrete operators that full-fill a variety of laws are based on the work presented in [16]. The mimetic discretizations are designed to numerically solve a variety of PDEs problems. An important aspect to use the MOLE library is to be able to write the PDEs in terms of the continuum differential operators $\nabla\cdot, \nabla, \nabla\times$ and the Laplacian. For instance, a 2nd-order mimetic Laplacian (one dimension) can be obtained using MOLE by:

$$div(k,m,dx) * grad(k,m,dx); \text{ or } lap(k,m,dx).$$

Note that the order of accuracy k, number of cells m and the step size for the space variable can be chosen. For the interested readers we refer them to [21].

2.3. Matlab built-in function: pdepe

The pdepe is a built-in Matlab function that is designed to solve systems of parabolic and elliptic PDEs in one space dimension and time. This built-in Matlab function belongs to the Partial Differential Equation Toolbox from Matlab and it is based on the algorithm developed in [34]. This method uses Galerkin and the Petrov-Galerkin methods depending on the singularity of the PDEs problem. The algorithm discretizes the space and then solves an explicit system of ODE that results from the discretization. The algorithm can use a variety of integrators in time, but the pdepe function uses the ode15s built-in Matlab function. This ode15s function is suitable for numerically solving differential-algebraic equations that emerge from elliptic PDEs [34]. The ode15s function uses a variable-step and it is a variable-order (VSVO) solver based on the numerical differentiation schemes of orders 1 to 5 [35,36]. We note that users can set options for the pdepe solver such as tolerance, the initial size of timestep and maximum step size however there is no restriction for the minimum size of the timestep. This in turn will always produce an optimized result regardless of the timestep used initially. More details on variable time-step will be presented in Section 3. For elliptic equations differential-algebraic equations emerge and some consistency issues related to the initial conditions might arise, but the pdepe function performs some adjustment to deal with this. However, this is not required for elements of the initial conditions vector that correspond to parabolic PDEs. It is important to remark that in all the numerical tests performed in this study we deal with parabolic PDEs as the Allen-Cahn and heat equations.

2.4. Errors

We will be computing the L_1 errors by comparing the solution approximations with the corresponding exact solution at a given t_f by using the formula

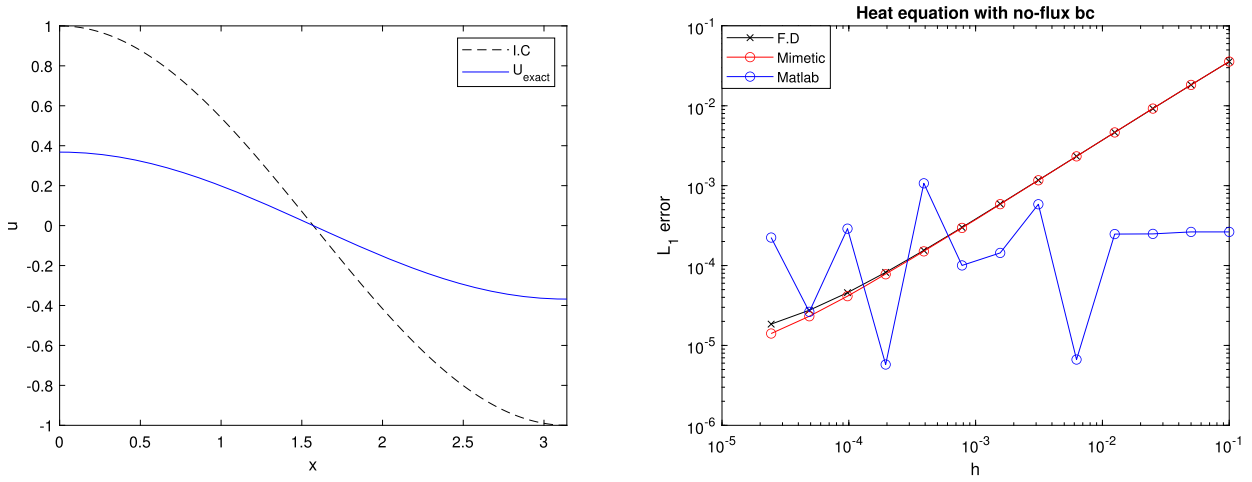

Fig. 2. Heat equation with no-flux boundary conditions: initial condition $\phi(x) = \cos(x)$ and final configuration (left). Right: L_1 errors computed with finite difference, mimetic-based methods and pdepe MATLAB solver. We use $N = 256$ on $\Omega = [0, \pi]$ with $t_f = 1.0$.

$$\text{Error}(h) = \int_{\Omega} |U(x, t_f; h) - U_{\text{exact}}(x, t_f)|\, dx, \tag{13}$$

where Ω defines the computational interval.

3. Numerical experiments

This section is devoted to present the numerical solutions of the Allen–Cahn and heat equations with periodic and non-periodic boundary conditions. We use the mimetic-based numerical method, the finite difference method and the pdepde built-in Matlab function to obtain the numerical solutions. In addition, we compute and present the L_1 errors of each of the numerical solutions in order to compare the accuracy of the solutions. We vary the time step size in order to investigate the order of convergence of each of the aforementioned methods. Since the studied methods are transferable to three-dimensional cases, we only provide numerical results for the two-dimensional Allen–Cahn and heat equations. We utilize the MOLE library with uniform staggered grids which enables us to have a better comparison with the finite difference method and the pdepe built-in Matlab function.

3.1. Heat equation with no flux boundary conditions

Let us first consider the heat equation in order to introduce and compare the mimetic operator method, classical finite difference scheme and the pdepe built-in Matlab function. Consider the heat equation with no-flux boundary conditions on $\Omega = [0, \pi]$,

$$u_t = u_{xx}, \tag{14}$$

$$u(x, 0) = \phi(x). \tag{15}$$

For the case of $\phi(x) = \cos(x)$, we obtain the close form solution given by

$$u(x, t) = e^{-t} \cos(x). \tag{16}$$

We compute the L_1 errors of the solutions given by the mimetic-based numerical method, the finite difference method and the pdepe Matlab function. The solution provided by the finite difference method uses ghost points as previously described. All the errors are computed at $t_f = 1.0$. Fig. 2 shows the L_1 errors for each of the methods and for different time step sizes. Notice that the slopes of the error curves indicate that the convergence rates of the mimetic-based numerical method and the finite difference methods are equal. Computing these slopes one gets approximately a value of 1. However, it can be seen that the solution generated by the pdepe Matlab function doesn't have a good order of convergence and the errors of its solutions don't improve by decreasing the time step size. This might be due to the fact that the pdepe built-in Matlab function solves the PDEs by means of solving a system of ODEs and the underlying Galerkin type method sacrifices accuracy in order to have more robustness [34]. In fact, the pdepe Matlab function relies on the ode15s built-in function that uses variable order. The solutions generated by the mimetic-based numerical method and the finite difference method have nearly the same errors with the mimetic solution being a little bit more accurate. The small difference in their errors is due to the space discretization on the boundary conditions since the particular scheme of the MOLE library is more accurate that the one of the finite difference, but this difference is only noticeable for small time step sizes. The pdepe Matlab function generates more accurate solutions only for large time step sizes which oftentimes are not used when accurate solutions are required

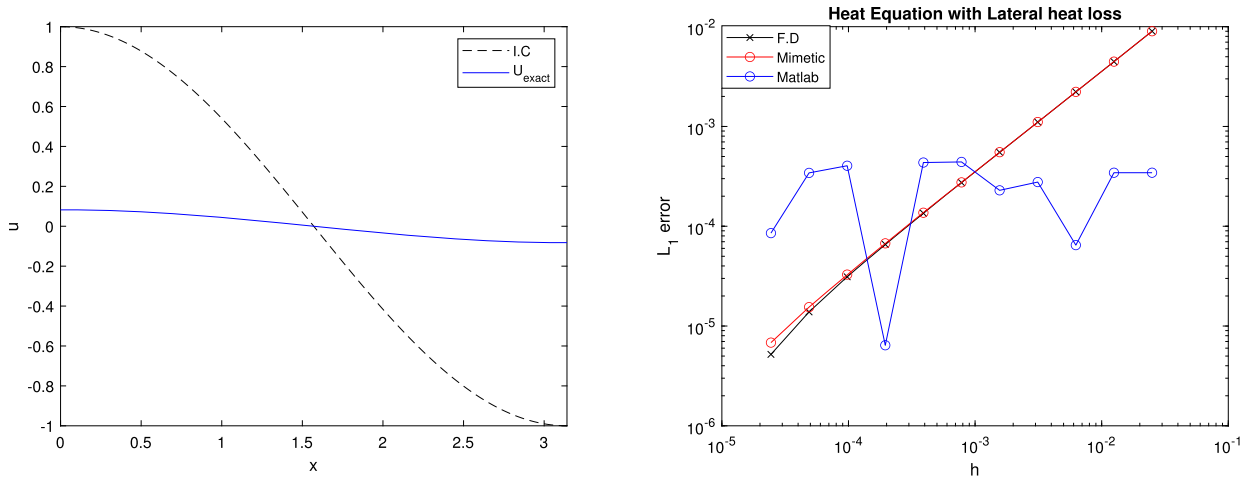

Fig. 3. Heat equation with lateral heat loss: initial condition $\phi(x) = \cos(x)$ and final configuration (left). Right: L_1 errors computed with finite difference, mimetic-based methods and pdepe MATLAB solver. We use $N = 256$ on $\Omega = [0, \pi]$ with $t_f = 1.0$.

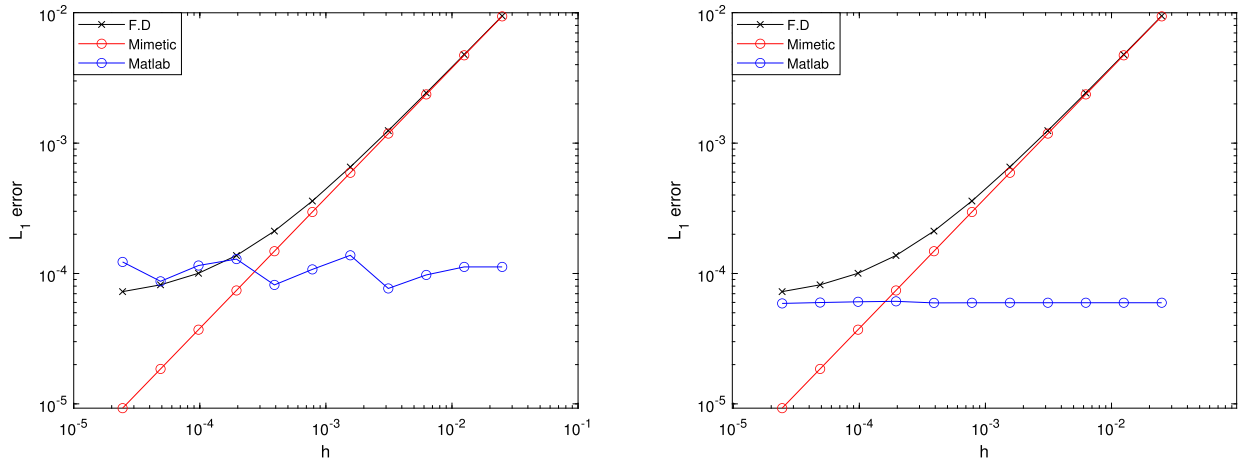

Fig. 4. Continued error plots corresponding to Fig. 3 with mimetic approach of higher order using $N = 100$ and pdepe error tolerance 10^{-4} (left) and 10^{-10} (right).

as in Allen-Cahn applications. Moreover, the time step size used by the pdepe function has a variable step so could be using smaller time step sizes than the finite difference and mimetic methods.

3.2. Heat equation with lateral heat loss

The heat equation with lateral heat loss can be formulated as an extension of the classic heat equation. The main difference relies that a term u is introduced at the right hand side of the classic formulation to account for either heat leaving, $-u$, or heat entering the medium, $+u$, in addition to what may happen at the boundaries. The heat equation with lateral heat loss and subject to no flux boundary conditions is given by

$$u_t = u_{xx} - \alpha u, \tag{17}$$

$$u(x, 0) = \phi(x). \tag{18}$$

Similar to the classic heat equation, it is possible to formulate an exact solution to the heat equation with lateral heat loss. Using $\phi(x) = \cos(x)$, leads to the following exact solution

$$u(x, t) = e^{-t(1+\alpha)} \cos(x). \tag{19}$$

We compute the L_1 errors of the solutions given by the mimetic-based numerical method, the finite difference method and the pdepe Matlab toolbox. All the errors are computed at $t_f = 1.0$. Fig. 3 shows the L_1 errors for each of the methods and for different time step sizes. It can be observed that the convergence rates of the mimetic-based numerical method and the finite difference method

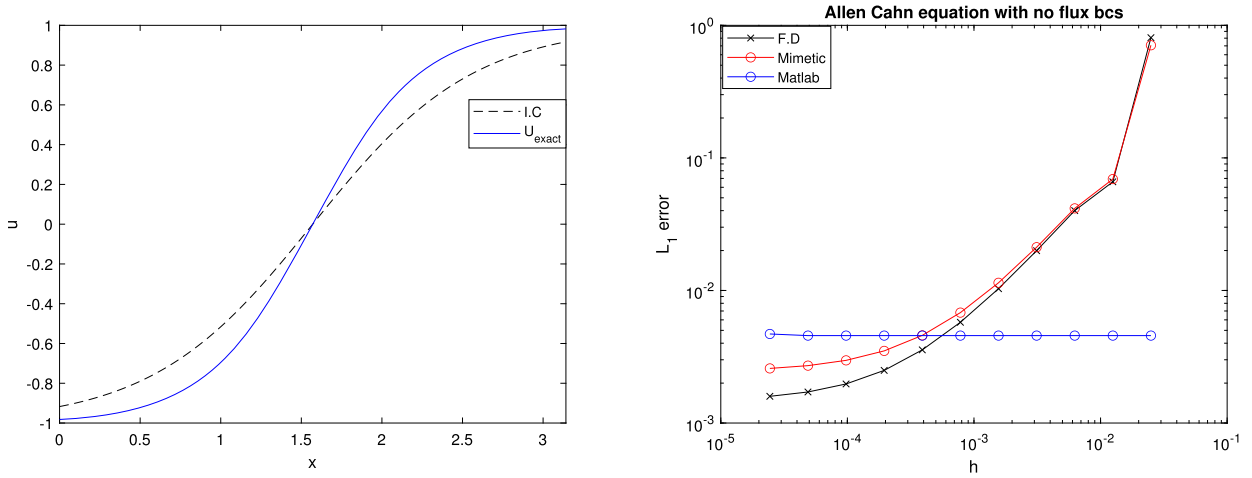

Fig. 5. Allen-Cahn equation with no-flux boundary conditions: initial condition $\phi(x) = \tanh(x - \pi/2)$ and final configuration (left). Right: L_1 errors computed with finite difference, mimetic-based methods and pdepe MATLAB solver. We use $N = 256$ on $\Omega = [0, \pi]$ with $t_f = 0.5$.

are equal. However, it can be seen that the pdepe Matlab toolbox provides a solution that can be considered correct, but its accuracy doesn't improve by decreasing the time step size as one might expect. As we have previously mentioned this could be related to two main factors such as the variable time step size that pdepe function uses and the underlying Galerkin type method. The solutions generated by the mimetic-based numerical method and the finite difference method have nearly the same errors. The small difference in their errors is due to the space discretization on the boundary conditions. Again, the pdepe Matlab function generates more accurate solutions only for large time step sizes and we already explained potential reasons for this. Fig. 4 further examines the behavior of the pdepe solver using $N = 100$ and all other settings the same as in Fig. 3. Using settings for numerical solution tolerance, maximum stepsize and initial stepsize we are able to get a non-fluctuating error plot as the stepsize h decreases. We note that pdepe solver always resources back to a variable timestep regardless of the initial h we start with. For this reason, the error plots from pdepe solver have that particular form which is very different from those obtained from the presented mimetic and finite difference formulations. In addition, Fig. 4 illustrates that the errors from mimetic approach can be superior to those from finite difference formulation when we consider higher order mimetic approximations. Taking advantage of the optimized libraries from MOLE, users can increase the order of approximations in their mimetic approximations easily. Here we choose mimetic approximation of order 4 to illustrate improved errors over the other two approaches.

3.3. Allen-Cahn equation

In this section we present numerical solutions of the Allen-Cahn and heat equations with different boundary conditions. The numerical solutions are computed with the mimetic-based method and its solutions compared with other numerical methods.

3.3.1. Allen-Cahn equation with no-flux boundary conditions

The Allen-Cahn equation can be view as an extended heat equation model that is used in materials science and engineering to describe the phase separation of materials [1]. The Allen-Cahn equation in 1D with no-flux boundary conditions can be written as

$$u_t = u_{xx} - \frac{1}{\epsilon^2} f(u), \tag{20}$$

$$u(x, 0) = \phi(x). \tag{21}$$

Here $\epsilon > 0$ is a parameter that defines thickness of the interface separating values of $u = +1$ and $u = -1$. The function $f(u) = u^3 - u$ is a non-linear term that favors phase separation. There is no closed form solution for the classical Allen-Cahn equation, but we can construct an exact solution by modifying the equation with the use of $u(x, y) = \tanh((t + 1)(x - \pi/2))$ and inserting it into the Allen-Cahn equation. Then, one gets a modified Allen-Cahn equation which is given by

$$u_t = u_{xx} - \frac{1}{\epsilon^2} f(u) + F(x, t), \tag{22}$$

$$u(x, 0) = \phi(x), \tag{23}$$

where $F(x, t) = \text{sech}[(1 + t)(-\pi/2 + x)]^2[-\pi/2 + x + (-42.444 + 2t(2 + t))\tanh((1 + t)(-\pi/2 + x))]$. We note that it has been common to construct exact solutions by modifying the phase field models [37]. We solved the forced Allen-Cahn equation using $\phi(x) = \tanh(x - \pi/2)$ and compare it against the exact solution at time $t_f = 0.5$.

We compute the L_1 errors of the solutions given by the mimetic-based numerical method, the finite difference method and the pdepe built-in Matlab function. The solution provided by the finite difference method uses ghost points as previously described. All

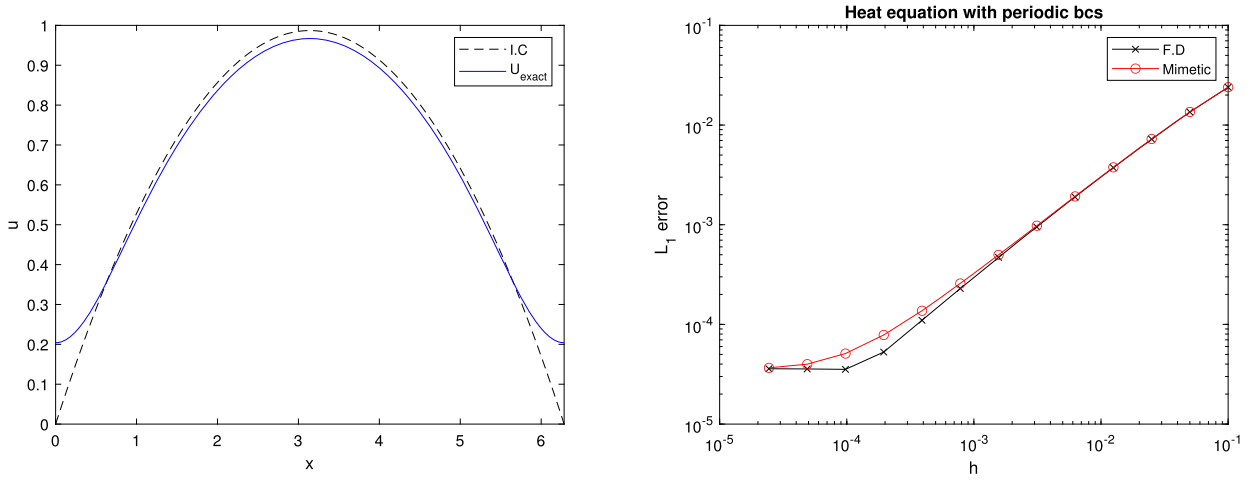

Fig. 6. Heat equation with periodic boundary conditions: initial condition $\phi(x) = -0.1x(x - 2\pi)$ and final configuration (left). Right: L_1 errors computed with finite difference and mimetic-based methods. We use $N = 512$ on $\Omega = [0, 2\pi]$ with $t_f = 0.1$.

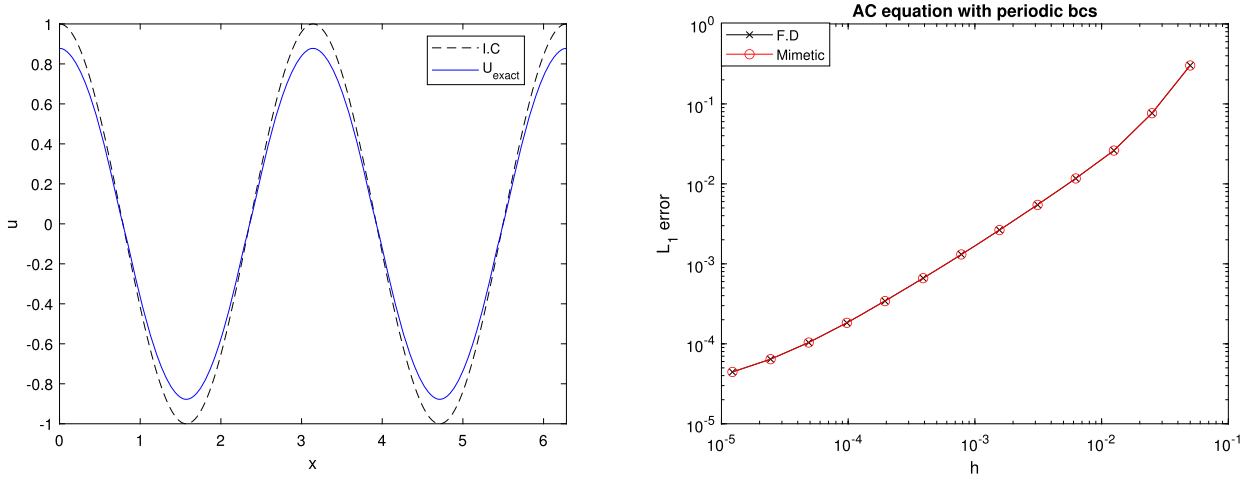

Fig. 7. Allen-Cahn equation with periodic boundary conditions: initial condition $\phi(x) = \cos(2x)$ and final configuration (left). Right: L_1 errors computed with finite difference and mimetic-based methods. We use $N = 512$ on $\Omega = [0, 2\pi]$ with $t_f = 0.1$.

the errors are computed at $t_f = 0.5$. Fig. 5 shows the L_1 errors for each of the methods and for different time step sizes. Notice that the slopes of the error curves indicate that the convergence rates of the mimetic-based numerical method and the finite difference methods are approximately equal. Computing these slopes one gets approximately a value of 1. However, it can be seen that the pdepe Matlab function solution seems to preserve a certain level accuracy for small h values but its associated error doesn't decrease in a consistent way. The solutions generated by the mimetic-based method and the finite difference method have nearly the same errors for larger time step sizes and slightly differ for smaller ones. These differences in the errors are due to the space discretization on the boundary conditions since we adapted the MOLE library in order to be able to introduce the particular boundary conditions used in this Allen-Cahn equation. Nevertheless, by modifying the space discretization on the boundary grid points it could be possible to improve the L_1 errors for smaller time step sizes. This is out of the scope of this study and can be studied in future studies.

3.3.2. Allen-Cahn equation with periodic boundary conditions

In this section, we also consider the heat and Allen-Cahn equations with periodic boundary conditions. Following the same approach as it was done in the previous numerical experiments, for the Allen-Cahn equation we use $\phi(x) = \cos(2x)$ as our initial condition. We compare the numerical solutions against $u(x, t) = \cos(2x)\cos(t)$ at $t_f = 0.1$ on the computational domain $\Omega = [0, 2\pi]$ with $\epsilon = 0.15$. For the heat equation with periodic boundary conditions, we construct a reference numerical solution with a small time step $h = 10^{-6}$ starting from the initial condition $\phi(x) = -0.1x(x - 2\pi)$. We note that similar reference solutions have been constructed for related phase field models [38]. The L_1 errors of the numerical solutions for the heat equation errors are presented in Fig. 6. The L_1 errors for the Allen-Cahn equation using the finite difference method and mimetic approach are presented in Fig. 7. It can be seen

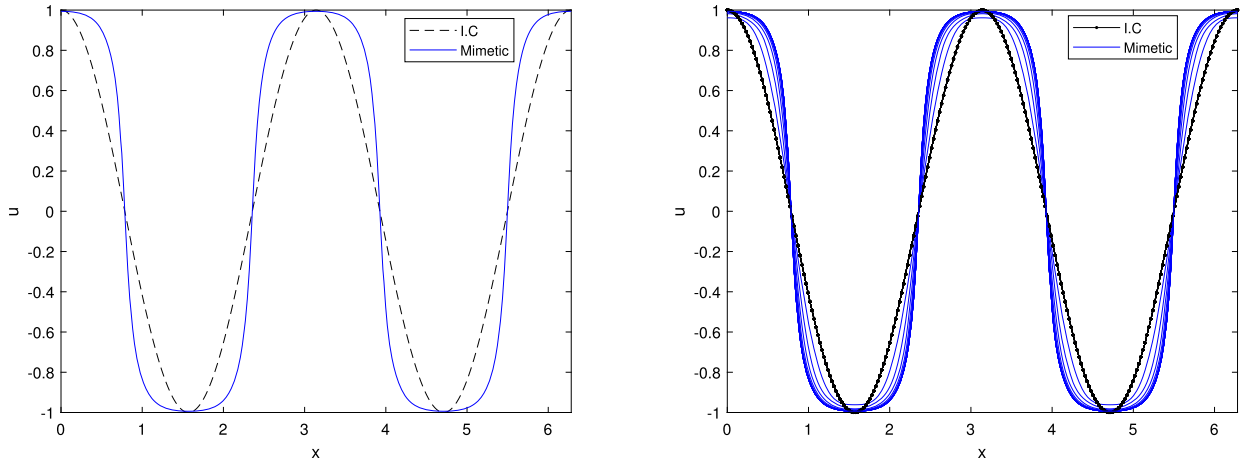

Fig. 8. Allen-Cahn equation: initial condition and final configuration (left). Right: dynamic evolution as t reaches $t_f = 5$. Curves correspond to unit increments of time. We use $N = 256$ on $\Omega = [0, 2\pi]$ with $\epsilon = 0.15$.

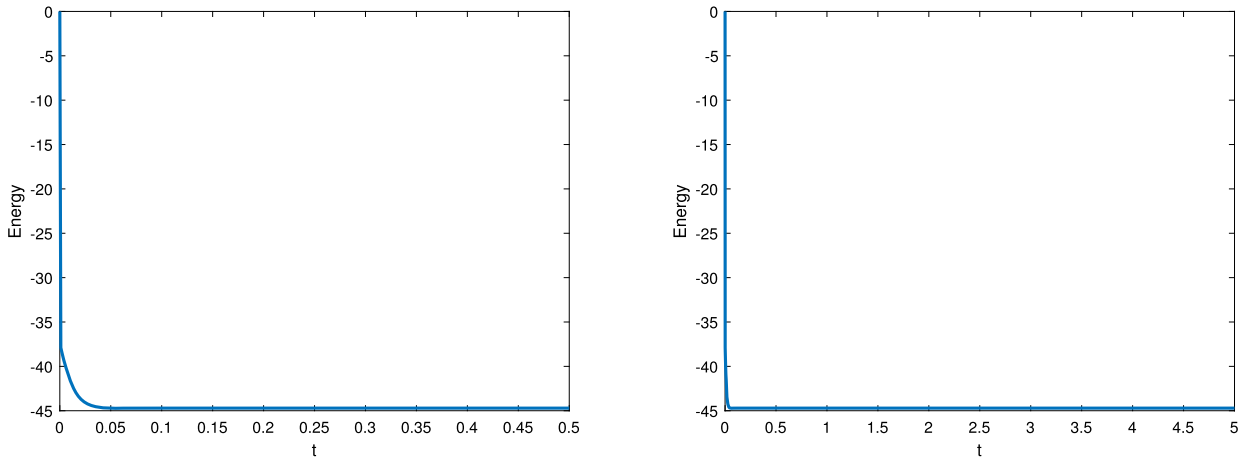

Fig. 9. Energy evolution corresponding to the numerical simulation from Fig. 8. Left figure shows a smaller range of t values to appreciate the energy decreasing property. Right figure contains the range of t values considered in the entire simulation.

that for the case of periodic boundary conditions both methods perform nearly identical. The differences in the L_1 errors are due to the space discretization on the boundary conditions since the MOLE library needs some adaptation.

The order of convergence for both methods is approximately one, this due to the fact that both numerical schemes have order one time discretizations.

4. Computed dynamics

Now that we have tested the mimetic-based numerical approach under a variety of numerical experiments, we now present the numerical solution to the Allen–Cahn equation using random initial conditions which models the problem of the mixing of two materials [37,38]. The motivation for this section is to test the performance of the scheme for longer runs and for initial conditions that are relevant to phase separation process.

4.1. 1D dynamics

We evolve $\phi(x) = \cos(2x)$ using mimetic approach for the Allen-Cahn equation with $N = 256$ using periodic boundary conditions. We let $t_f = 5$ and $\epsilon = 0.15$. Simulations are presented in Fig. 8. It can be seen that the peaks of the initial condition are widen in those regions in which the solution is close to the minima $(+1, -1)$ which is what is expected from the Allen-Cahn equation (without any forcing terms). We also consider an initial condition in terms of random values between $(-1, 1)$. This initial condition is relevant since it models an initial state in which two materials are in the mixed state. Fig. 10 shows the evolution of the random initial condition as it settles into equilibrium reaching to the pure states, $u \approx +1$ and $u \approx -1$. From these two figures, we can also see the slow and fast dynamics that are possible according to the corresponding initial configurations in the problem. It takes only $t_f = 1$ for the case

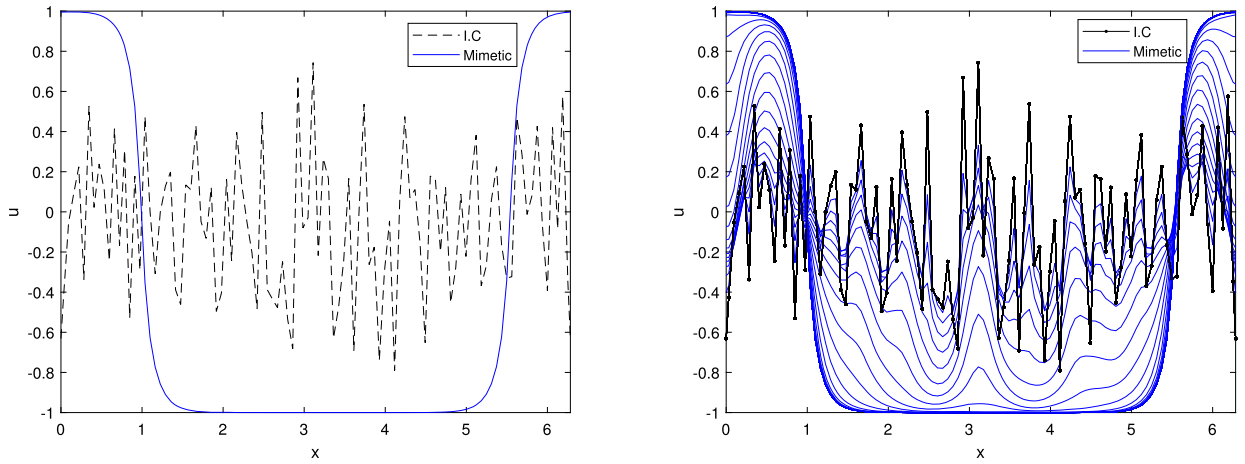

Fig. 10. Allen-Cahn equation: initial condition and final configuration (left). Right: dynamic evolution as t reaches $t_f = 1$. Curves correspond to $h = 0.01$ increments of time. We use $N = 100$ on $\Omega = [0, 2\pi]$ with $\epsilon = 0.15$.

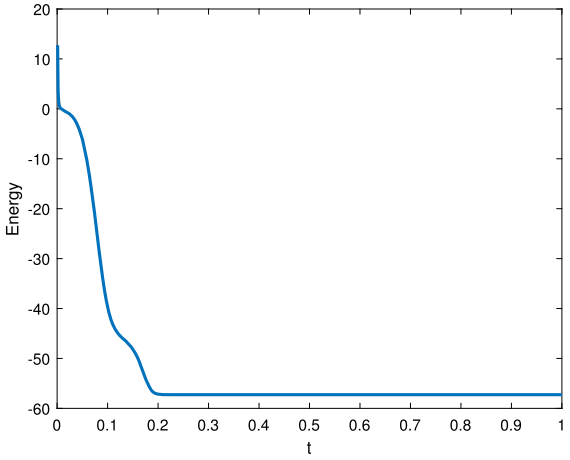

Fig. 11. Energy evolution corresponding to the numerical simulation from Fig. 10.

of random initial conditions to exhibit phase separation, but it takes longer for $\phi(x) = \cos(2x)$ to exhibit the same phenomena. The mimetic approach is able to evolve these two initial conditions to their respective final time while preserving the energy decreasing property during the entire duration of the simulation (see Fig. 9 and Fig. 11).

4.2. 2D dynamics

Having tested the proposed methods in 1D, we now illustrate the adaptability of the mimetic finite difference operator with an energy stable splitting method. We extend our methods to the Allen-Cahn (AC) equation in 2D which reads

$$u_t = \Delta u - \frac{1}{\epsilon^2} f(u), \tag{24}$$

$$u(x, y, 0) = \phi(x, y), \tag{25}$$

and illustrate their performance. A series of time stepping schemes that were based on the convexity splitting (CS) [39] scheme were presented in [38] to solve phase field models. Using the ideas of CS approach coupled with a mimetic finite difference method, leads to the following scheme for the Allen-Cahn equation

$$\frac{U^{n+1} - U^n}{h} = \Delta U^{n+1} - \frac{a}{\epsilon^2} U^{n+1} - \frac{1}{\epsilon^2}\left[U^{n^3} - (1 + a)U^n\right], \tag{26}$$

where $U^n_{i,j} \approx u(t_n, x_i, y_j)$ and $a > 2$ is the CS splitting parameter [38] and our computational domain is a box $\Omega = [0, 2\pi]^2$. The grid spacing for the two space variables x and y follows a direct extension from the 1D space discretization. We solve the Allen-Cahn equation using the convexity splitting approach coupled with the mimetic finite difference (CSMFD) method using a random initial

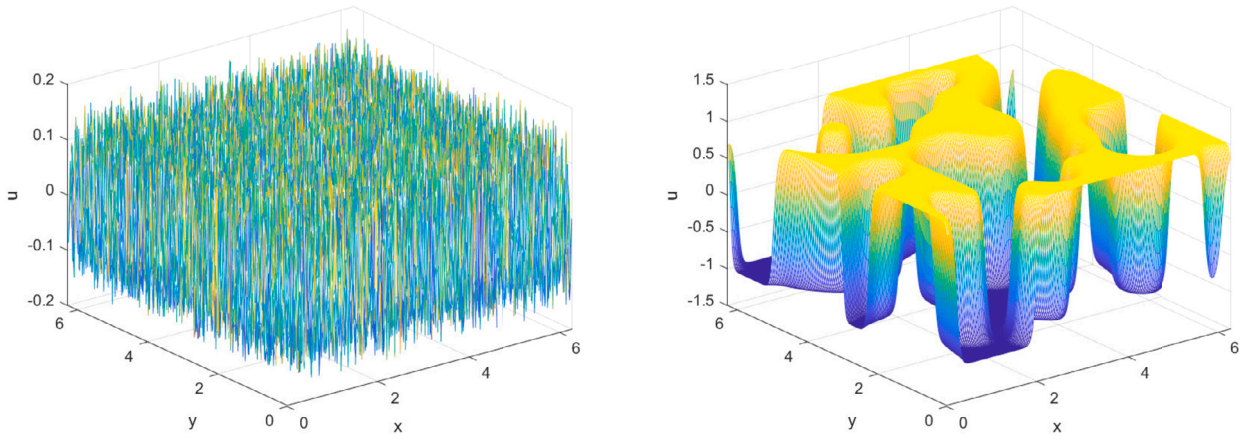

Fig. 12. Allen-Cahn equation: initial condition (left). Right: dynamic evolution illustrating phase separation in 2D at $t_f = 1$. We use $N = 256$ on $\Omega = [0, 2\pi]^2$ with $\epsilon = 0.05$ and $h = 0.01$.

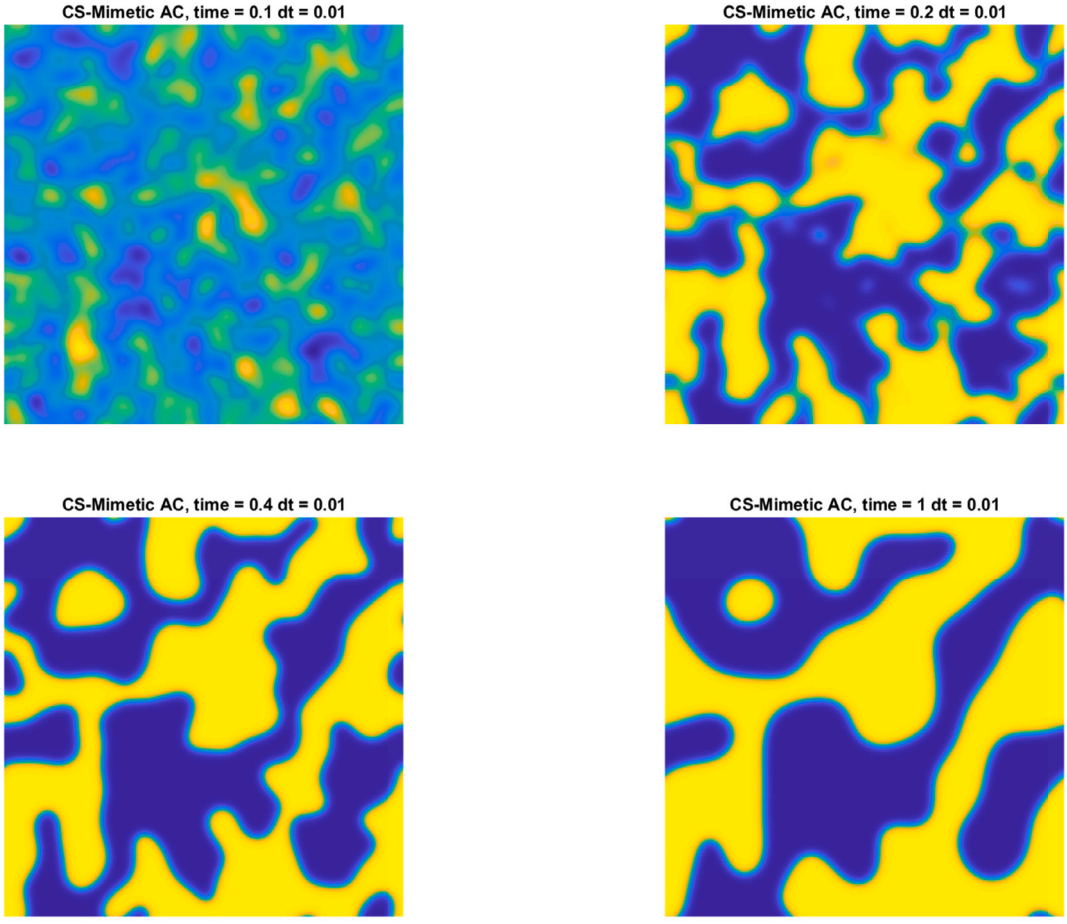

Fig. 13. Allen-Cahn equation: same initial condition and parameters from the simulation from Fig. 12. Snapshots taken at $t = 0.1, 0.2, 0.4, 1$ capturing the phase separation process of materials in a binary mixture.

state which is composed of values ranging from $(-0.15, 0.15)$. Simulation results are presented in Fig. 12. This simulation shows the phase separation process of two materials that initially started at a mixed state. Here the yellow color represents a pure state ($u = +1$) of the first material and blue represents the second material in the mixture ($u = -1$).

In the literature for phase field models, it is far more common to illustrate the phase separation process of materials and pattern formation in 2D problems by using level curves. We illustrate this by re-running the simulation presented in Fig. 12 with a more

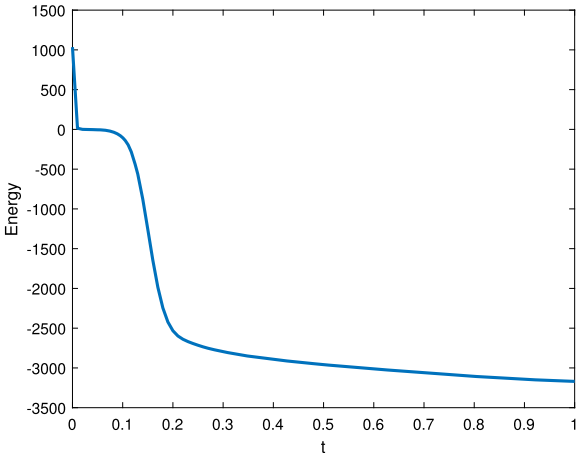

Fig. 14. Energy evolution corresponding to the numerical simulation from Fig. 13.

traditional view of the 2D dynamics of the Allen-Cahn equation. Simulation snapshots are taken at different values of $t = 0.1, 0.2, 0.4, 1$ and presented in Fig. 13. Figures illustrate the phase separation process in which domains of pure states (blue or yellow) began to segregate and grow as time progresses. CSMFD method is able to capture the correct dynamics of the phase separation process while preserving the energy decreasing property as shown in Fig. 14.

5. Conclusion

In this paper, we investigated and implemented a numerical method that is based on a mimetic finite difference operator for solving different variants of the nonlinear Allen–Cahn equation with periodic and non-periodic boundary conditions. In addition, we also analyzed the performance of the mimetic-based numerical method by applying it to the classical heat equation with a variety of boundary conditions. We evaluated the performance of the mimetic-based numerical method by comparing the errors of its solutions with those obtained by a classical finite difference method and the pdepe built-in Matlab function. We computed the L_1 errors of the numerical solutions by using the exact solutions when they are available or with the reference solutions obtained for a refined mesh. We adapted the mimetic-based numerical method by using the MOLE library. Some few adaptations were needed in order to deal with the Allen-Cahn equation in one dimension due to the specific boundary conditions of the examples. We presented several results with regard to errors and numerical convergence tests in order to provide insight into the reliability and accuracy of the mimetic-based numerical method. The numerical results show that the method based on the mimetic difference operator is a reliable method for solving the Allen–Cahn and heat equations with both periodic and non-periodic boundary conditions. Moreover, the mimetic-based method shows an order of convergence as good as the finite difference method and better than the pdepe Matlab function. On the other hand, the numerical results show that the pdepe Matlab function solutions seem to preserve a fixed level accuracy for small time step values, but the associated errors don't decrease for smaller time step sizes. We also found that the solutions generated by the mimetic-based method are more accurate than the ones generated by the pdepe Matlab solver provided that the timestep is small enough ($h \approx 10^{-3}$). Similar errors between the mimetic and finite difference approach were found when considering the second order approximation. However, when considering high order approximation the mimetic approach was superior in accuracy and ease of implementation (while requiring small N) since reaching higher order is accomplished very efficiently due to the optimized MOLE libraries.

We also combined the mimetic finite difference operator with an energy stable splitting method to solve the Allen-Cahn equation in 2D. In particular, we utilize the convexity splitting approach coupled with the mimetic finite difference method using random initial states. This initial condition enables us to model the problem of phase separation process of two materials. The results showed that this approach is numerically reliable and is capable to describe the correct dynamics for the Allen–Cahn equation by ensuring the energy decreasing property. Finally, the results presented in this study show that the mimetic finite-difference operator or the MOLE library is a reliable tool to solve the Allen-Cahn equation and it is promising for other phase field models. The MOLE library is an open source and can be modified to adapt to the particular phase field model and can be easily implemented with higher order accuracy. Further investigations will include the study of computational aspects to other phase field models including extended versions of Allen-Cahn equations related to bio-medical applications [40], Cahn-Hilliard (CH) equation [41,42] and the Phase Field Crystal (PFC) equation [38,43].

Data availability

No data was used for the research described in the article.

Acknowledgements

S.O is grateful for the start-up funds provided by New Mexico Tech. S.O thanks Thomas Wilteski (Duke University) for hosting a research visit during summer 2023. G.G.P thanks Universitat Politècnica de Valencia for a year long visit under grant María Zambrano (UPV, funding from the Spain Ministry of Universities funded by the European Union—Next Generation EU). Research reported in this publication was supported by an Institutional Development Award (IDeA) from the National Institute of General Medical Sciences of the National Institutes of Health under grant number P20GM103451.

Appendix A

In this section we give the expression for the divergence operator for the mimetic operator in 1D:

$$D^n_{1d} = \frac{1}{dx}\begin{pmatrix} 0 & 0 & 0 & \dots & 0 \\ -1 & 1 & 0 & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & -1 & 1 & 0 \\ 0 & \dots & 0 & 0 & 0 \end{pmatrix}_{(n+2)\times(n+1)},$$

where the first and last zero rows are associated with the boundary nodes where the divergence is not defined. For more details about the mimetic operators and some recent applications, we refer interested readers to the following references [17,27,28,32].

References

[1] S.M. Allen, J.W. Cahn, A microscopic theory for antiphase boundary motion and its application to antiphase domain coarsening, Acta Metall. 27 (6) (1979) 1085–1095.
[2] J.W. Cahn, On spinodal decomposition, Acta Metall. 9 (9) (1961) 795–801, https://doi.org/10.1016/0001-6160(61)90182-1, https://www.sciencedirect.com/science/article/pii/0001616061901821.
[3] A. Miranville, The Cahn–Hilliard equation and some of its variants, AIMS Math. 2 (3) (2017) 479–544, https://doi.org/10.3934/Math.2017.2.479, https://www.aimspress.com/article/doi/10.3934/Math.2017.2.479.
[4] J. Wang, Z. Shi, Multi-reconstruction from points cloud by using a modified vector-valued Allen–Cahn equation, Mathematics 9 (12) (2021) 1326.
[5] D. Wang, Y. Li, H. Jia, A two-grid finite element method for the Allen-Cahn equation with the logarithmic potential, Numer. Methods Partial Differ. Equ. 39 (2) (2023) 1251–1265.
[6] B. Inan, M.S. Osman, T. Ak, D. Baleanu, Analytical and numerical solutions of mathematical biology models: the Newell-Whitehead-Segel and Allen-Cahn equations, Math. Methods Appl. Sci. 43 (5) (2020) 2588–2600.
[7] D. Jeong, J. Kim, An explicit hybrid finite difference scheme for the Allen–Cahn equation, J. Comput. Appl. Math. 340 (2018) 247–255.
[8] Y. Kim, D. Lee, Numerical investigation into the dependence of the Allen–Cahn equation on the free energy, Adv. Comput. Math. 48 (3) (2022) 36.
[9] C. Schwarzmeier, M. Holzer, T. Mitchell, M. Lehmann, F. Häusl, U. Rüde, Comparison of free-surface and conservative Allen–Cahn phase-field lattice Boltzmann method, J. Comput. Phys. 473 (2023) 111753.
[10] S. Aihara, T. Takaki, N. Takada, Multi-phase-field modeling using a conservative Allen–Cahn equation for multiphase flow, Comput. Fluids 178 (2019) 141–151.
[11] S. Aihara, N. Takada, T. Takaki, Highly conservative Allen–Cahn-type multi-phase-field model and evaluation of its accuracy, Theor. Comput. Fluid Dyn. 37 (5) (2023) 639–659.
[12] Y. Khan, I. Ali, S. Islam, Q. Biao Wu, Some new computational methods for the Allen-Cahn equation with non-periodic boundary conditions arising in computational fluid dynamics, Int. J. Numer. Methods Heat Fluid Flow 23 (4) (2013) 588–597.
[13] R. Guo, L. Ji, Y. Xu, High order local discontinuous Galerkin methods for the Allen-Cahn equation: analysis and simulation, J. Comput. Math. (2016) 135–158.
[14] H. Liu, J. Yan, A local discontinuous Galerkin method for the Korteweg–de Vries equation with boundary effect, J. Comput. Phys. 215 (1) (2006) 197–218.
[15] S. Jafarzadeh, L. Wang, A. Larios, F. Bobaru, A fast convolution-based method for peridynamic transient diffusion in arbitrary domains, Comput. Methods Appl. Mech. Eng. 375 (2021) 113633.
[16] J. Corbino, J.E. Castillo, High-order mimetic finite-difference operators satisfying the extended Gauss divergence theorem, J. Comput. Appl. Math. 364 (2020) 112326.
[17] J. Brzenski, J.E. Castillo, Solving Navier–Stokes with mimetic operators, Comput. Fluids 254 (2023) 105817.
[18] K. Lipnikov, G. Manzini, D. Svyatskiy, Analysis of the monotonicity conditions in the mimetic finite difference method for elliptic problems, J. Comput. Phys. 230 (7) (2011) 2620–2642.
[19] K. Lipnikov, G. Manzini, M. Shashkov, Mimetic finite difference method, J. Comput. Phys. 257 (2014) 1163–1227.
[20] H.-O. Kreiss, G. Scherer, Finite element and finite difference methods for hyperbolic partial differential equations, in: Mathematical Aspects of Finite Elements in Partial Differential Equations, Elsevier, 1974, pp. 195–212.
[21] Mole: mimetic operators library enhanced, https://raw.githubusercontent.com/openjournals/joss-papers/joss.06005/joss.06005/10.21105.joss.06005.pdf. (Accessed 11 January 2022), 2022.
[22] pdepe built-in Matlab function, https://www.mathworks.com/help/matlab/ref/pdepe.html. (Accessed 11 January 2022).
[23] D. Kumar, V. Kumar, V. Singh, Mathematical modeling of brown stock washing problems and their numerical solution using MATLAB, Comput. Chem. Eng. 34 (1) (2010) 9–16.
[24] D. Yudianto, X. Yuebo, A comparison of some numerical methods in solving 1-D steady-state advection dispersion reaction equation, Civ. Eng. Environ. Syst. 27 (2) (2010) 155–172.
[25] L. Córdova, O. Rojas, B. Otero, J. Castillo, Compact finite difference modeling of 2-D acoustic wave propagation, J. Comput. Appl. Math. 295 (2016) 83–91.
[26] J. Villamizar, L. Mendoza, G. Calderón, O. Rojas, J.E. Castillo, High order mimetic differences applied to the convection–diffusion equation: a matrix stability analysis, GEM Int. J. Geomath. 14 (1) (2023) 26.
[27] J.E. Castillo, M. Yasuda, Linear systems arising for second-order mimetic divergence and gradient discretizations, J. Math. Model. Algorithms 4 (2005) 67–82.
[28] F. Solano-Feo, J. Guevara-Jordan, O. Rojas, B. Otero, R. Rodriguez, A new mimetic scheme for the acoustic wave equation, J. Comput. Appl. Math. 295 (2016) 2–12.
[29] F. Guillén-González, G. Tierra, Energy-stable and boundedness preserving numerical schemes for the Cahn-Hilliard equation with degenerate mobility, Appl. Numer. Math. 196 (2024) 62–82.

[30] T. Chu, O.T. Schmidt, RBF-FD discretization of the Navier-Stokes equations on scattered but staggered nodes, J. Comput. Phys. 474 (2023) 111756.
[31] M. Sohaib, A. Shah, Numerical solution of coupled Cahn–Hilliard Navier–Stokes equations for two-phase flows having variable density and viscosity, Math. Methods Appl. Sci.
[32] A. Srinivasan, M. Dumett, C. Paolini, G.F. Miranda, J.E. Castillo, Mimetic finite difference operators and higher order quadratures, GEM Int. J. Geomath. 14 (1) (2023) 19.
[33] J.E. Castillo, R. Grone, A matrix analysis approach to higher-order approximations for divergence and gradients satisfying a global conservation law, SIAM J. Matrix Anal. Appl. 25 (1) (2003) 128–142.
[34] R.D. Skeel, M. Berzins, A method for the spatial discretization of parabolic equations in one space variable, SIAM J. Sci. Stat. Comput. 11 (1) (1990) 1–32.
[35] L.F. Shampine, M.W. Reichelt, The MATLAB ODE suite, SIAM J. Sci. Comput. 18 (1) (1997) 1–22.
[36] L.F. Shampine, M.W. Reichelt, J.A. Kierzenka, Solving index-1 DAEs in MATLAB and Simulink, SIAM Rev. 41 (3) (1999) 538–552.
[37] Y. Yan, W. Chen, C. Wang, S.M. Wise, A second-order energy stable bdf numerical scheme for the Cahn-Hilliard equation, Commun. Comput. Phys. 23 (2) (2018) 572–602.
[38] K. Glasner, S. Orizaga, Improving the accuracy of convexity splitting methods for gradient flow equations, J. Comput. Phys. 315 (2016) 52–64, https://doi.org/10.1016/j.jcp.2016.03.042, http://www.sciencedirect.com/science/article/pii/S0021999116001947.
[39] D.J. Eyre, Unconditionally Gradient Stable Time Marching the Cahn-Hilliard Equation, MRS Proceedings, vol. 529, Cambridge Univ Press, 1998, p. 39.
[40] S. Wise, J. Lowengrub, H. Frieboes, V. Cristini, Three-dimensional multispecies nonlinear tumor growth—I: model and numerical method, J. Theor. Biol. 253 (3) (2008) 524–543, https://doi.org/10.1016/j.jtbi.2008.03.027, https://www.sciencedirect.com/science/article/pii/S0022519308001525.
[41] J. Shen, X. Yang, Numerical approximations of Allen-Cahn and Cahn-Hilliard equations, Discrete Contin. Dyn. Syst. 28 (4) (2010) 1669–1691.
[42] Y. He, Y. Liu, T. Tang, On large time-stepping methods for the Cahn-Hilliard equation, Appl. Numer. Math. 57 (5) (2007) 616–628.
[43] H. Gomez, X. Nogueira, An unconditionally energy-stable method for the phase field crystal equation, in: Higher Order Finite Element and Isogeometric Methods, Comput. Methods Appl. Mech. Eng. 249–252 (2012) 52–61, https://doi.org/10.1016/j.cma.2012.03.002, https://www.sciencedirect.com/science/article/pii/S0045782512000692.