

Novel energy dissipative method on the adaptive spatial discretization for the Allen-Cahn equation

Jing-Wei Sun(孙竟巍), Xu Qian(钱旭), Hong Zhang(张弘), and Song-He Song(宋松和)
Citation: Chin. Phys. B, 2021, 30 (**7**): 070201. DOI: 10.1088/1674-1056/abe37b
Journal homepage: http://cpb.iphy.ac.cn; http://iopscience.iop.org/cpb

What follows is a list of articles you may be interested in

Constructing reduced model for complex physical systems via interpolation and neural networks

Xuefang Lai(赖学方), Xiaolong Wang(王晓龙, and Yufeng Nie(聂玉峰)
Chin. Phys. B, 2021, 30 (**3**): 030204. DOI: 10.1088/1674-1056/abd92e

A meshless algorithm with the improved moving least square approximation for nonlinear improved Boussinesq equation

Yu Tan(谭渝) and Xiao-Lin Li(李小林)
Chin. Phys. B, 2021, 30 (**1**): 010201. DOI: 10.1088/1674-1056/abaed7

Direct immune-SCIR public-opinion propagation model based on real-time online users

Yun-Ming Wang(王运明), Tian-Yi Guo(郭天一)†, Wei-Dong Li(李卫东)‡, and Bo Chen(陈波)
Chin. Phys. B, 2020, 29 (**10**): 100204. DOI: 10.1088/1674-1056/aba9c0

Lump and interaction solutions to the (3+1)-dimensional Burgers equation

Jian Liu(刘健), Jian-Wen Wu(吴剑文)
Chin. Phys. B, 2020, 29 (**3**): 030201. DOI: 10.1088/1674-1056/ab6964

Average vector field methods for the coupled Schrödinger–KdV equations

Zhang Hong, Song Song-He, Chen Xu-Dong, Zhou Wei-En
Chin. Phys. B, 2014, 23 (**7**): 070208. DOI: 10.1088/1674-1056/23/7/070208

Novel energy dissipative method on the adaptive spatial discretization for the Allen–Cahn equation*

Jing-Wei Sun(孙竟巍)[1], Xu Qian(钱旭)[1,†], Hong Zhang(张弘)[1], and Song-He Song(宋松和)[1,2]

[1] Department of Mathematics, National University of Defense Technology, Changsha 410073, China
[2] State Key Laboratory of High Performance Computing, National University of Defense Technology, Changsha 410073, China

(Received 25 December 2020; revised manuscript received 23 January 2021; accepted manuscript online 5 February 2021)

We propose a novel energy dissipative method for the Allen–Cahn equation on nonuniform grids. For spatial discretization, the classical central difference method is utilized, while the average vector field method is applied for time discretization. Compared with the average vector field method on the uniform mesh, the proposed method can involve fewer grid points and achieve better numerical performance over long time simulation. This is due to the moving mesh method, which can concentrate the grid points more densely where the solution changes drastically. Numerical experiments are provided to illustrate the advantages of the proposed concrete adaptive energy dissipative scheme under large time and space steps over a long time.

Keywords: moving mesh, energy dissipative, average vector field method, Allen–Cahn equation

PACS: 02.60.Cb, 05.70.Fh, 02.70.Bf **DOI:** 10.1088/1674-1056/abe37b

1. Introduction

Phase-field models are usually constructed to reproduce a given interfacial dynamics. For instance, in solidification problems, the front dynamics is given by a diffusion equation for either concentration or temperature in the bulk and some boundary conditions at the interface, which constitutes the sharp interface model. In the past few decades, phase-field models have been widely applied in image analysis, material science, engineering, fluid mechanics, and life science, etc. (e.g., Refs. [6,8,13–15]) Among them, one fundamental model is the Allen–Cahn model, which is originally introduced to describe the non-conservative phase variables in the phase separation process. The Allen–Cahn model is recognized as gradient flow systems, for which there exists a Lyapunov function, known as the free energy. From a modeling view, given a specified Lyapunov function or a free energy function, the Allen–Cahn type equations can be derived as the L^2 gradient flow. This generality makes the Allen–Cahn type equations extremely useful in modeling many interfacial or multiphase problems. In this paper, we are concerned with numerical solutions of the following Allen–Cahn equation:

$$\begin{cases} u_t = \varepsilon^2 \Delta u + f(u), & x \in \Omega,\ t \in (0,T], \\ u(\cdot,t)\ \text{is}\ \Omega\ \text{periodic}, & t \in [0,T], \\ u(x,0) = u_0(x), \end{cases} \tag{1}$$

where Ω is a domain in $\mathbb{R}$, Δ is the Laplacian operator, and $u, \varepsilon, f(u) = u - u^3$ represent the difference between the concentrations of the two mixtures' components, the interfacial width, and a polynomial double-well potential, respectively. The Allen–Cahn equation was originally introduced by Allen and Cahn in Ref. [1] to describe the motion of anti-phase boundaries in crystalline solids. After that, it has been widely extended to model various phenomena in nature, such as phase transitions,[2] image analysis,[3] mean curvature flows,[4] and crystal growth.[5] It is well known that the Allen–Cahn equation possesses nonlinear stability due to its time-dependent dissipation, i.e., it can be viewed as the L^2-gradient flow of the Ginzburg–Landau free energy functional

$$E(u) = \int_\Omega \frac{1}{2}\varepsilon^2 |\nabla u|^2 + F(u)\,\mathrm{d}x, \tag{2}$$

where $F(u) = \frac{1}{4}(u^2-1)^2$ is typically called the double-well type potential. Figure 1 shows the function $F(u)$. Differentiating it with respect to time t, we obtain the following equation:

$$\frac{\mathrm{d}}{\mathrm{d}t}E(u) = \left(\frac{\delta E(u)}{\delta u}, g(u)\right) = -\left|\left|\frac{\partial u}{\partial t}\right|\right|^2, \quad \forall t > 0, \tag{3}$$

where $g(u) = \varepsilon^2 \Delta u + f(u)$, the symbols $(\cdot,\cdot)$ and $||\cdot||$ are the standard L^2 inner product and norm defined as

$$(u,v) = \int_\Omega uv\,\mathrm{d}x, \quad ||u|| = \left(\int_\Omega |u|^2\,\mathrm{d}x\right)^{\frac{1}{2}}. \tag{4}$$

Besides the dissipation property, another is that the Allen–Cahn equation admits the maximum principle preserving (MPP) property:[9,16] if the initial value and the boundary

*Project supported by the National Key R&D Program of China (Grant No. 2020YFA0709800), the National Natural Science Foundation of China (Grant Nos. 11901577, 11971481, 12071481, and 12001539), the Natural Science Foundation of Hunan, China (Grant Nos. S2017JJQNJJ0764 and 2020JJ5652), the fund from Hunan Provincial Key Laboratory of Mathematical Modeling and Analysis in Engineering (Grant No. 2018MMAEZD004), the Basic Research Foundation of National Numerical Wind Tunnel Project, China (Grant No. NNW2018-ZT4A08), and the Research Fund of National University of Defense Technology (Grant No. ZK19-37).

†Corresponding author. E-mail: qianxu@nudt.edu.cn

© 2021 Chinese Physical Society and IOP Publishing Ltd

conditions are bounded by constant β ($\beta = 1$ for Eq. (1)), then the entire solution is also bounded by β, i.e., $||u(\cdot,t)||_{L_\infty} \leq \beta, \forall t > 0$.

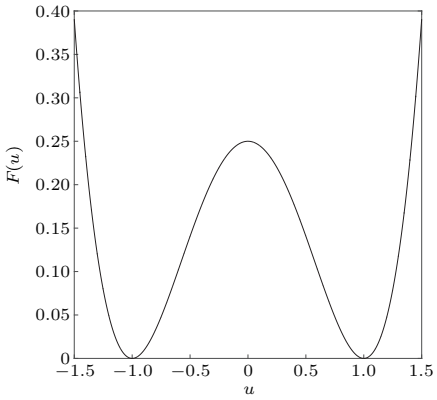


Fig. 1. A double well potential $F(u) = 0.25(u^2 - 1)^2$.

Since available theoretical solutions for the Allen–Cahn equation are still very limited, numerical investigation is an important tool to understand the physical behavior of this system. In particular, the average vector field (AVF) method[17] was proposed to simulate Hamiltonian ordinary differential equations (ODEs) due to their long time simulation and good preservation properties of conservative quantities of the original problem. It is a kind of B-series method which can be used to analyze the accuracy of the scheme by using the root series theory. There have been a number of works solving various PDEs by using the AVF method in time discretization. Gong, Jiang *et al.*[18,19] extended the original second-order AVF scheme to the higher-order schemes. This is undoubted because of the satisfactory performance of the AVF method in maintaining the high-order accuracy of energy.

There are many ways to solve the problem of energy conservation, including the symplectic and multisymplectic method (e.g., Refs. [7,35,36]), the high-order Runge–Kutta method (e.g., Refs. [10,34]), the wavelet method (e.g., Refs. [11,12]), and especially the AVF method for time-dependent PDE systems (e.g., Ref. [38]), etc. These systems usually have a structure that has significantly evolved as the integration of PDEs proceed, including interfaces, shocks, changes of phase, and high vorticity or regions of complexity, etc. They are further associated with small length or time scales, rapid movement of the solution feature, and the possibility of blow-up of solution. Although researchers have adapted the finite element, finite volume, finite difference, or collocation method and achieved satisfactory numerical results, one problem is inevitable, that is, small enough time and space steps are required to achieve the desired accuracy, which takes a lot of computing costs. The adaptive moving mesh method has a favorable characteristic that could overcome this problem. However, as far as we know, there is no related work on how to extend the AVF method to preserve energy dissipative property on the nonuniform meshes.

Adaptive methods for solving PDEs can be roughly divided into three fundamental categories: h-refinement method, p-refinement method, and r-refinement method. For h-refinement method, it starts with an initially uniform mesh and then locally coarsens or refines it by the inclusion or deletion of mesh points. For p-refinement method, finite element discretization of the PDE is used with local polynomials of some particular order. And for r-refinement method, it starts with a uniform mesh and then moves the mesh points, keeping the mesh points fixed as the solution evolves. The mesh points are then concentrated into regions where the solution has individual behavior, usually typified by a rapid variation of either the solution or one of its higher derivatives. Compared with other methods, there are several significant advantages of r-refinement methods in various applications such as computational fluid mechanics, phase-field models, and convective heat transfer, etc. (e.g., Refs. [20,21]) For instance, it is convenient to work with the same number of mesh points. This makes linear algebra rather easier because the matrices considered have a constant sparsity structure. Moreover, the discretization strategy on the mesh is also easier. The third advantage of r-refinement methods is that under certain conditions, the mesh movement strategy coupled with the discrete PDE can be regarded as a large dynamical system that would be easier for theoretical analysis.

Our main proposal is to achieve a balance among effectiveness, high efficiency, and structure preservation. We are committed to extending the AVF method to nonuniform meshes via the r-refinement method. The extension is based on the "mapping method" which is the most popular one among the moving mesh method, and the change of coordinates plays an important role. Recently, Yaguchi extended the discrete variational derivative method to nonuniform grids[22] and proved that this method could likewise do the structure-preserving on nonuniform grids. A summation by parts formula is introduced on one-dimensional nonuniform grids since it plays a very important role in the discrete variational method. We also use this method as a reference to prove the theorem. In a recent study,[23] an hr adaptive method is proposed to get the spatial and temporal adaptation effect and then to solve the nonlinear Schrödinger equation (NSE). Nevertheless, the proposed method only involves r-refinement and explores the adaptation in space.

The rest of this paper is organized as follows. In Section 2, we review the idea of the mapping method and two approaches of discretization for the PDE. In Section 3, we introduce our mesh movement strategy, including the mesh density function and its corresponding smoothing method. The dissipative moving mesh scheme based on the classical AVF method is defined in Section 4, and relevant theorems and proofs are given. Section 5 contains numerical experiments

and results of the application of the scheme proposed. Conclusion and outlook are discussed in Section 6.

2. Discretization and mesh adaptation

Notice that the Allen–Cahn equation as one of the Hamiltonian PDEs

$$\frac{\partial u}{\partial t} = \mathcal{S}\frac{\delta \mathcal{H}}{\delta u}, \tag{5}$$

where $\mathcal{S}$ is a negative semi-definite differential operator, and $\mathcal{H}$ denotes a certain energy functional, such as the Hamiltonian or free energy. Equation (5) can be rewritten as the following form:[24]

$$\frac{\partial u}{\partial t} = (-1)^{s+1}\left(\frac{\partial}{\partial x}\right)^{2s}\frac{\delta \mathcal{H}}{\delta u}, \quad s = 0, \tag{6}$$

which will be a general form for later discretization and theorems.

2.1. Spatial discretization

For simplicity of notation, we define the following finite difference operators:

$$\delta^{+}u_j = u_{j+1} - u_j,\ \delta^{-}u_j = u_j - u_{j-1},\ \delta^{c}u_j = \frac{u_{j+1}-u_{j-1}}{2},$$

$$\delta_x^{+}u_j = \frac{u_{j+1}-u_j}{x_{j+1}-x_j},\ \delta_x^{-}u_j = \frac{u_j-u_{j-1}}{x_j-x_{j-1}},\ \delta_x^{c}u_j = \frac{u_{j+1}-u_{j-1}}{x_{j+1}-x_{j-1}},$$

where $0 \le j \le N-1$. Furthermore, we define the second order discrete derivative as

$$\delta_x^2 u_j = \frac{2}{x_{j+1}-x_{j-1}}\left[\frac{(u_{j+1}-u_j)}{(x_{j+1}-x_j)} - \frac{(u_j-u_{j-1})}{(x_j-x_{j-1})}\right].$$

In addition, for any u and v, we define the discrete L_2 inner product and notions as follows:

$$\langle \boldsymbol{u}, \boldsymbol{v}\rangle_{\kappa} = \boldsymbol{u}^{\mathrm{T}} D(\kappa)\boldsymbol{v} = \sum_{j=0}^{N-1}\kappa_j u_j v_j, \tag{7}$$

where $D(\kappa) = \mathrm{diag}(\kappa_0, \ldots, \kappa_{N-1})$ is an N by N diagnoal matrix.

In order to introduce mesh movement, Eq. (6) is rewritten with respect to a moving reference frame, and the problem is recast in terms of the independent variables ξ and t, by using the transformation

$$x = x(\xi, t),\ \ \xi \in \Omega_{\mathrm{c}} \equiv [0,1],\ \ t \in (0,T], \tag{8}$$

from computational space $\Omega_{\mathrm{c}} \times (0,T]$ to physical space $\Omega_{\mathrm{p}} \times (0,T]$. A uniform mesh covering Ω_{c} is given by

$$\xi_i = \frac{i}{N},\ \ i = 0,1,\ldots,N, \tag{9}$$

and the corresponding nonuniform mesh on Ω_{p} is

$$x_{\mathrm{l}} = x_0 < x_1(t) < \cdots < x_{N-1}(t) < x_N = x_{\mathrm{r}}, \tag{10}$$

where

$$x_i(t) = x(\xi_i, t),\ \ i = 0,1,\ldots,N. \tag{11}$$

Equation (6) is rewritten in the computational domain Ω_{c}

$$\frac{\partial u}{\partial t} = (-1)^{s+1}\left(\frac{\mathrm{d}\xi}{\mathrm{d}x}\frac{\mathrm{d}}{\mathrm{d}\xi}\right)^{2s}\left(\frac{\delta \mathcal{H}}{\delta u}\right)_{\mathrm{cs}}, \tag{12}$$

where $\left(\frac{\delta \mathcal{H}}{\delta u}\right)_{\mathrm{cs}}$ is the transformed variational derivative

$$\left(\frac{\delta \mathcal{H}}{\delta u}\right)_{\mathrm{cs}} = \frac{\partial \mathcal{H}}{\partial u} - \frac{\mathrm{d}\xi}{\mathrm{d}x}\frac{\partial}{\partial \xi}\left(\frac{\mathrm{d}x}{\mathrm{d}\xi}\frac{\mathrm{d}\xi}{\mathrm{d}x}\frac{\partial \mathcal{H}}{\partial u_x}\right). \tag{13}$$

This is a natural form of the variational derivative in the computational space. Yaguchi *et al.*[22] have proved that the variational structure is retained in the computational space. To make it convenient for the theoretical analysis, we decompose Eq. (12) to the following form:

$$\begin{aligned}\frac{\partial u}{\partial t} =& -\left(-\frac{\mathrm{d}\xi}{\mathrm{d}x}\frac{\mathrm{d}}{\mathrm{d}\xi}\right)\left(-\frac{\mathrm{d}\xi}{\mathrm{d}x}\frac{\mathrm{d}}{\mathrm{d}\xi}\right)^{s-1}\left(\frac{\mathrm{d}\xi}{\mathrm{d}x}\frac{\mathrm{d}x}{\mathrm{d}\xi}\frac{\mathrm{d}}{\mathrm{d}\xi}\right)\\ &\times\left(\frac{\mathrm{d}\xi}{\mathrm{d}x}\frac{\mathrm{d}}{\mathrm{d}\xi}\right)^{s-1}\left(\frac{\delta \mathcal{H}}{\delta u}\right)_{\mathrm{cs}}, \quad s = 0,1,2,3,\ldots.\end{aligned} \tag{14}$$

The total energy of Eq. (6) is defined by

$$H(t) := \int_{\Omega}\mathcal{H}(u, u_x)\,\mathrm{d}x. \tag{15}$$

As is widely known, under certain boundary conditions, Eq. (6) has the dissipation property

$$\frac{\mathrm{d}H}{\mathrm{d}t} \le 0. \tag{16}$$

The following theorem explains that in the computational domain, Eq. (12) also has the dissipation property. It is worth mentioning that although the theorem gives special boundary conditions, it also applies to periodic boundary conditions. It can refer to Ref. [22].

Theorem 1[22] Suppose that the boundary condition satisfies

$$\left[\frac{\partial u}{\partial t}\frac{\partial \mathcal{H}}{\partial u_x}\right]_{\xi=0}^{\xi=1} = 0. \tag{17}$$

Suppose also that

$$\begin{aligned}&\left[\left(\left(\frac{\mathrm{d}\xi}{\mathrm{d}x}\frac{\partial}{\partial \xi}\right)^{s-p}\frac{\mathrm{d}\xi}{\mathrm{d}x}\frac{\mathrm{d}x}{\mathrm{d}\xi}\frac{\mathrm{d}\xi}{\mathrm{d}x}\frac{\partial}{\partial \xi}\left(\frac{\mathrm{d}\xi}{\mathrm{d}x}\frac{\partial}{\partial \xi}\right)^{s-1}\left(\frac{\delta \mathcal{H}}{\delta u}\right)_{\mathrm{cs}}\right)\right.\\ &\left.\times\left(\left(\frac{\mathrm{d}\xi}{\mathrm{d}x}\frac{\partial}{\partial \xi}\right)^{p-1}\left(\frac{\delta \mathcal{H}}{\delta u_x}\right)_{\mathrm{cs}}\right)\right]_{\xi=0}^{\xi=1} = 0, \quad p = 1,\ldots,s,\end{aligned} \tag{18}$$

if $s \ge 1$. Then the solutions of Eq. (12) have the dissipation property

$$\frac{\mathrm{d}H_{\mathrm{cs}}}{\mathrm{d}t} \le 0, \quad H_{\mathrm{cs}}(t) = \int_0^1 H(u, u_x)\frac{\mathrm{d}x}{\mathrm{d}\xi}\,\mathrm{d}\xi.$$

The effect of mesh movement in the time discretization of the physical PDE system can be treated with either the quasi-Lagrange approach or the rezoning approach. We would give the quasi-Lagrange approach as a reference. The latter one will be highlighted and applied to the adaptive method.

2.1.1. Quasi-Lagrange approach

With the quasi-Lagrange approach, the mesh points are considered to move continuously in time, and physical time derivatives are transformed into time derivatives along mesh trajectories supplemented with a convective term reflecting mesh movement. This approach mainly use the chain rule, i.e., denoting the transformed variable $v(\xi,t) = u(x(\xi,t),t)$, we have

$$\frac{\partial v}{\partial t} = \frac{\partial u}{\partial t} + \frac{\partial x}{\partial t}\frac{\partial u}{\partial x},$$

where $\frac{\partial x}{\partial t} = \frac{\partial x}{\partial t}(\xi,t)$ determines the mesh speed. In the new coordinates (ξ,t), we can transform the PDE and discrete it.

2.1.2. Rezoning approach

The rezoning approach, as its name implies, is going to update the mesh at each time step by using certain mesh equations or generators, and the mesh points are considered to move intermittently in time. It is an integrated process that involves solving alternately for the physical solution and the mesh. The physical solution is interpolated from the old mesh to the new one, and the physical PDE is then discretized on the new mesh, which is held fixed for the current time step. This is the main difference from the quasi-Lagrange approach, and Fig. 2 gives an intuitive display. As can be seen, the mesh points are considered to move continuously for the quasi-Lagrange approach of temporal discretization of physical PDEs, while the mesh is considered to vary only at time instants for the rezoning approach of temporal discretization of physical PDEs. Interpolation of the physical solution determines the success of this method, and it is often necessary to use a conservative interpolation scheme that preserves some solution quantities and properties. We will show the detail in later chapters.

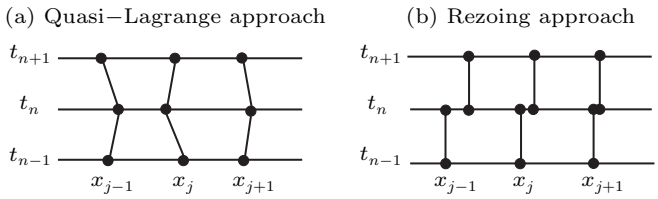


Fig. 2. Two different mesh movement approaches with time varying in Ref. [25].

In fact, for a moving mesh method, the discrete physical PDE and the mesh equation give a coupled system. Moreover, on account of high coupling, the mesh can respond promptly to any change occurring in the physical solution. Unfortunately, coupling leads to a highly nonlinear equation that is difficult to solve due to its more complicated structure. Therefore we choose the rezoning approach to strive for maximum decoupling. Since the time lag effect brought and the computational cost by the rezoning approach cannot be ignored, we have to take a smaller time step.

3. Mesh movement strategy

Since only the one-dimensional problem is concerned in this paper, the de Boor's equidistributing algorithm could not be more appropriate. Given a continuous function $\rho(x)$, typically depending on the first or second order derivatives of the underlying solution u to be adapted, the number of grid points $N-1$ and the bounded domain $[0,L]$ where u is defined, equidistribution needs to find a grid $X_h : x_0 = 0 < x_1 < \cdots < x_N = L$ such that

$$\int_{x_0}^{x_1} \rho(x)\,\mathrm{d}x = \int_{x_i}^{x_{i+1}} \rho(x)\,\mathrm{d}x, \quad \text{for } i = 1,\ldots,N-1, \tag{19}$$

where the function $\rho(x)$ is to minimize the error of the numerical approximation, which is currently called the mesh density function (e.g., Refs. [25,26]). To ensure that the equidistribution is unique, assume that the mesh density function is strictly positive, i.e,

$$\rho(x) > 0, \quad \forall x \in [0,L].$$

A typical and simple choice of $\rho(x)$ is the arc-length mesh density function

$$\rho(x) = \sqrt{1 + \left(\frac{\partial u}{\partial x}\right)^2}. \tag{20}$$

Its semi-discrete form is

$$\rho_{i+\frac{1}{2}} = \sqrt{1 + \left(\frac{u_{i+1} - u_i}{x_{i+1} - x_i}\right)^2}, \tag{21}$$

where $\rho_{i+\frac{1}{2}}$ represents a semi-discrete approximation to $\rho(\frac{x_i + x_{i+1}}{2})$. Typically, u_i is only given by numerical approximations evaluated on a previously evaluated grid and u_{i+1} can be found by interpolation over the new grid, which will be interpreted in more detail later.

In this case, the meaning of the equidistribution principle is that the weighted arc length of u over each interval is equal. That means that both u and ρ are approximate values. In addition, notice that if u is not smooth, the discrete mesh density function computed this way can often change abruptly and unnecessarily slow down the computation. To obtain a smoother mesh and also make Eq. (19) easier to integrate, a common practice is to smooth the mesh density function $\rho(x)$. A simple but effective smoothing method is given as follows:[25]

$$\begin{aligned}
\bar{\rho}_1 &:= \frac{1}{2}\rho_1 + \frac{1}{2}\rho_N, \\
\bar{\rho}_i &:= \frac{1}{4}\rho_{i-1} + \frac{1}{2}\rho_i + \frac{1}{4}\rho_{i+1}, \quad j = 2,\ldots,N-1, \\
\bar{\rho}_N &:= \frac{1}{2}\rho_{N-1} + \frac{1}{2}\rho_N.
\end{aligned} \tag{22}$$

Combining Eq. (19) with Eq. (11), a natural formula is given

$$\int_{x_0}^{x(\xi)} \rho(x)\,\mathrm{d}x = \xi \int_{x_0}^{x_M} \rho(x)\,\mathrm{d}x. \tag{23}$$

After twice differential with respect to ξ, a classical equation yields

$$\frac{\mathrm{d}}{\mathrm{d}\xi}\left(\rho(x)\frac{\mathrm{d}x}{\mathrm{d}\xi}\right)=0. \tag{24}$$

Meanwhile, we can get the following system of $(N-1)$ equations by discretizing Eq. (19)

$$(x_i-x_{i-1})\,\rho_{i-\frac{1}{2}}=(x_{i+1}-x_i)\,\rho_{i+\frac{1}{2}},\ \ i=1,\ldots,N-1, \tag{25}$$

with the boundary condition $x_0=0, x_N=L$. Furthermore, when $\xi_{i+1}-\xi_i$ is a constant, Eqs. (21) and (23) can be obtained on the computational grid ξ_h

$$\frac{2}{\xi_{i+1}-\xi_{i-1}}\left(\rho_{i+\frac{1}{2}}\frac{x_{i+1}-x_i}{\xi_{i+1}-\xi_i}-\rho_{i-\frac{1}{2}}\frac{x_i-x_{i-1}}{\xi_i-\xi_{i-1}}\right)=0. \tag{26}$$

4. Adaptive energy dissipative method

Lemma 1 in Ref. [27] indicated that the operator ∇ as the standard gradient could replace the variational derivative in a finite number of dimensions, i.e.,

$$\frac{\delta\bar{\mathcal{H}}'}{\delta\bar{u}}\equiv\nabla\bar{\mathcal{H}}, \tag{27}$$

where the notations $\bar{\mathcal{H}}$ and $\bar{u}$, which are different from the notation $\bar{\rho}$ as mentioned earlier, denote the discrete values of $\mathcal{H}$ and u and $\bar{\mathcal{H}}'=\bar{\mathcal{H}}D(\kappa)$. The following two lemmas are given for later proof.

Lemma 1[22] Suppose that a solution of Eq. (6) satisfies the condition

$$\left[\frac{\partial u}{\partial t}\frac{\mathrm{d}x}{\mathrm{d}\xi}\frac{\mathrm{d}\xi}{\mathrm{d}x}\frac{\partial\mathcal{H}}{\partial u_x}\right]_{\xi=0}^{\xi=1}=0, \tag{28}$$

then

$$\frac{\mathrm{d}H_{\mathrm{cs}}}{\mathrm{d}t}=\int_0^1\frac{\partial u}{\partial t}\left(\frac{\delta\mathcal{H}}{\delta u}\right)_{\mathrm{cs}}\frac{\mathrm{d}x}{\mathrm{d}\xi}\mathrm{d}\xi, \tag{29}$$

and $\left(\frac{\delta\mathcal{H}}{\delta u}\right)_{\mathrm{cs}}$ satisfies

$$\left(\frac{\delta\mathcal{H}}{\delta u}\right)_{\mathrm{cs}}=\frac{\delta\mathcal{H}}{\delta u}.$$

Lemma 2[22] Suppose that the periodic boundary condition is satisfied. Then

$$\begin{aligned}\frac{1}{\Delta t}\left(H^{n+1}-H^n\right)&=\sum_{j=0}^{N}\kappa_j\left(\frac{U_j^{n+1}-U_j^n}{\Delta t}\right)\\&\quad\times\left(\frac{\delta\bar{\mathcal{H}}}{\delta\left(U^{n+1},U^n\right)}\right)_j.\end{aligned} \tag{30}$$

4.1. Design of schemes for dissipative equations

Motivated by Yaguchi,[22] we design a scheme that serves for dissipative equations and proves the energy dissipation property. Firstly, we recall the classical AVF scheme as follows. One thing should be noticed that since the variational derivative, which is of continuous form or not, is the same in the physical space and the computational space and in the discrete case gradient operator is equal to variational derivative, as shown in Eq. (27), therefore in the discrete case $\left(\nabla\bar{\mathcal{H}}\right)=\left(\nabla\bar{\mathcal{H}}\right)_{\mathrm{cs}}$.

Theorem 2 Let $\bar{\mathcal{S}}$ be any negative-definite matrix approximation to $\mathcal{S}$ and $\bar{\mathcal{H}}$ be any finite difference approximation to $\mathcal{H}$. Let

$$f(u):=\left(\bar{\mathcal{S}}\nabla\bar{\mathcal{H}}(u)\right)_{\mathrm{cs}}, \tag{31}$$

then we obtain an AVF scheme in the computational space

$$\frac{u^{n+1}-u^n}{\Delta t}=\int_0^1 f\left((1-\tau)u^n+\tau u^{n+1}\right)\mathrm{d}\tau. \tag{32}$$

This theorem is obvious. Furthermore, we rewrite the above equation by adding the component of Eq. (14) and get its discrete counterpart

$$\begin{aligned}\frac{U_j^{n+1}-U_j^n}{\Delta t}&=\left(\kappa_j^{-1}\delta_k^-\right)\left((x_\xi)_{k,j}^{-1}\delta_{k-1}^-\right)\cdots\\&\quad\times\left((x_\xi)_{2,j}^{-1}\delta_1^-\right)\left((x_\xi)_{1,j}^{-1}\delta^c\right)\left((x_\xi)_{1,j}^{-1}\delta_1^+\right)\cdots\\&\quad\times\left((x_\xi)_{k-1,j}^{-1}\delta_{k-1}^+\right)\left((x_\xi)_{k,j}^{-1}\delta_k^+\right)\\&\quad\times\int_0^1\nabla\left((1-\tau)U_j^n+\tau U_j^{n+1}\right)_{\mathrm{cs}}\mathrm{d}\tau,\end{aligned} \tag{33}$$

where $(x_\xi)_{k,j}$ and δ_k can be chosen arbitrarily $(k=1,\ldots,s)$, and s is equivalent to s of Eq. (6).

Theorem 3 If the periodic boundary condition is satisfied, then Eq. (33) has an energy dissipation law, i.e.,

$$\frac{1}{\Delta t}\left(H^{n+1}-H^n\right)\le 0.$$

Proof For simplicity, replace

$$\int_0^1\nabla\left((1-\tau)U_j^n+\tau U_j^{n+1}\right)_{\mathrm{cs}}\mathrm{d}\tau$$

with the notation $\left(\bar{\nabla}\right)_{\mathrm{cs}}$. Furthermore, replace discrete variational derivative term with $\left(\bar{\nabla}\right)_{\mathrm{cs}}$, then we get

$$\frac{1}{\Delta t}\left(H^{n+1}-H^n\right)=\sum_{j=0}^{N}\kappa_j\left(\frac{U_j^{n+1}-U_j^n}{\Delta t}\right)\left(\bar{\nabla}\right)_{\mathrm{cs}}. \tag{34}$$

Substituting Eq. (33) and applying the summation by parts in Ref. [22] gives

$$\begin{aligned}&=-\sum_{j=0}^{N}\kappa_j\left[\left(-\kappa_j^{-1}\delta_k^-\right)\left(-(x_\xi)_{k,j}^{-1}\delta_{k-1}^-\right)\cdots\left(-(x_\xi)_{2,j}^{-1}\delta_1^-\right)\left((x_\xi)_{1,j}^{-1}\kappa_j(x_\xi)_{1,j}^{-1}\delta_1^+\right)\cdots\left((x_\xi)_{k,j}^{-1}\delta_k^+\right)\left(\bar{\nabla}\right)_{\mathrm{cs}}\right]\left(\bar{\nabla}\right)_{\mathrm{cs}}\\&=-\sum_{j=0}^{N}\kappa_j\left[\left(-(x_\xi)_{k,j}^{-1}\delta_{k-1}^-\right)\cdots\left(-(x_\xi)_{2,j}^{-1}\delta_1^-\right)\left((x_\xi)_{1,j}^{-1}\kappa_j(x_\xi)_{1,j}^{-1}\delta_1^+\right)\cdots\left((x_\xi)_{k,j}^{-1}\delta_k^+\right)\left(\bar{\nabla}\right)_{\mathrm{cs}}\right]\left(\kappa_j^{-1}\delta_k^+\left(\bar{\nabla}\right)_{\mathrm{cs}}\right)\end{aligned}$$

$$= -\sum_{j=0}^{N} \kappa_j \left[\left(-\kappa_j^{-1} \delta_{k-1}^{-} \right) \cdots \left(-(x_\xi)_{2,j}^{-1} \delta_1^{-} \right) \left((x_\xi)_{1,j}^{-1} \kappa_j (x_\xi)_{1,j}^{-1} \delta_1^{+} \right) \cdots \left((x_\xi)_{k,j}^{-1} \delta_k^{+} \right) \left(\bar{\nabla} \right)_{\mathrm{cs}} \right] \left((x_\xi)_{k,j}^{-1} \delta_k^{+} \left(\bar{\nabla} \right)_{\mathrm{cs}} \right).$$

Continue in this fashion to obtain

$$= -\sum_{j=0}^{N} \left[\left(-\kappa_j^{-1} \delta_{k-2}^{-} \right) \cdots \left(-(x_\xi)_{2,j}^{-1} \delta_1^{-} \right) \left((x_\xi)_{1,j}^{-1} \kappa_j (x_\xi)_{1,j}^{-1} \delta_1^{+} \right) \cdots \left((x_\xi)_{k,j}^{-1} \delta_k \right) \left(\bar{\nabla} \right)_{\mathrm{cs}} \right] \left[\left((x_\xi)_{k-1}^{-1} \delta_{k-1}^{+} \right) \left((x_\xi)_{k,j}^{-1} \delta_k^{+} \right) \left(\bar{\nabla} \right)_{\mathrm{cs}} \right]$$

$$\vdots$$

$$= -\sum_{j=0}^{N} \kappa_j \left[\left(-\kappa_j^{-1} \delta_1^{-} \right) \left((x_\xi)_{1,j}^{-1} \kappa_j (x_\xi)_{1,j}^{-1} \delta_1^{+} \right) \cdots \left((x_\xi)_{k,j}^{-1} \delta_k^{+} \right) \cdots \left((x_\xi)_{k,j}^{-1} \delta_k^{+} \right) \left(\bar{\nabla} \right)_{\mathrm{cs}} \right] \left[\left((x_\xi)_{2,j}^{-1} \delta_2^{+} \right) \cdots \left((x_\xi)_{k,j}^{-1} \delta_k^{+} \right) \left(\bar{\nabla} \right)_{\mathrm{cs}} \right]$$

$$= -\sum_{j=0}^{N} \kappa_j \left[\left((x_\xi)_{1,j}^{-1} \kappa_j (x_\xi)_{1,j}^{-1 \delta_1^{+}} \right) \cdots \left((x_\xi)_{k,j}^{-1} \delta_k^{+} \right) \left(\bar{\nabla} \right)_{\mathrm{cs}} \right] \left[\left(\kappa_j \delta_1^{+} \right) \left((x_\xi)_{2,j}^{-1} \delta_2^{+} \right) \cdots \left((x_\xi)_{k,j}^{-1} \delta_k^{+} \right) \left(\bar{\nabla} \right)_{\mathrm{cs}} \right]$$

$$= -\sum_{j=0}^{N} \kappa_j \left[\left((x_\xi)_{1,j}^{-1} \delta_1^{+} \right) \cdots \left((x_\xi)_{k,j}^{-1} \delta_k^{+} \right) \left(\bar{\nabla} \right)_{\mathrm{cs}} \right]^2 \leq 0.$$

4.2. A complete r-algorithm based on rezoning approach

Due to the use of the rezoning approach and the constant number of grid points, the solution at each new time t^{n+1} is calculated by three main steps. Firstly, the grid X_j^{n+1} is computed based on the solution $\boldsymbol{u}^n$ on X_j^n, which is calculated in the previous time step. Then $\boldsymbol{u}^n$ is interpolated over the new grid X_j^{n+1} by a solution procedure that preserves the approximation of the energy dissipative property. Subsequently, the solution $\boldsymbol{u}^{n+1}$ at the time t^{n+1} can be found by Eq. (33). The solution procedure is illustrated by the following pseudo-code specifically.

Algorithm 1 The r-algorithm based on rezoning approach

1: Initialize variables; select all parameters including $\Delta \boldsymbol{x}$ and Δt, etc.
2: Give a uniform mesh over the physical domain and determine $\boldsymbol{u}^0$ based on initial condition φ_0
3: Utilize the mesh density function $\omega(\boldsymbol{x})$ and de Boor's equidistribution principle to compute an initial grid $\boldsymbol{x}^0$.
4: for $n = 0 : NT - 1$ do
5: Set reltol $= 1.0e-10$ and abstol $= 1$;
6: while abstol $>$ reltol do
7: Compute the approximately equidistributing grid $\boldsymbol{x}^{n+1}$ from $\boldsymbol{x}^n$ and $\boldsymbol{u}^n$
8: Compute $\tilde{\boldsymbol{u}} = \psi_{(\boldsymbol{x}^{n+1}, \boldsymbol{x}^n)} \boldsymbol{u}^n$, for a given interpolation operator $\psi_{(\boldsymbol{x}^{n+1}, \boldsymbol{x}^n)}$ mapping from $\boldsymbol{x}^n$ to $\boldsymbol{x}^{n+1}$.
9: Solve Eq. (33) where $U_j^{(n)}$ is replaced by $\tilde{U}_j$ on the new mesh $\boldsymbol{x}^{n+1}$
10: abstol $= \max \left(\mathrm{abs} \left(U_j^{(n+1)} - U_j^{(n)} \right) \right)$
11: end while
12: Set $n := n + 1$
13: end for

As mentioned before, the key step is to select the appropriate interpolation operator ψ. In Ref. [28], Huang gave the detailed process and error analysis of linear interpolation on the adaptive moving mesh. However, in the actual calculation, we observe that the cubic interpolation operator (the MATLAB built-in function **pchip**) works best with respect to the error between the numerical solution and the exact solution.

5. Numerical experiments

According to Eq. (33), we obtain the full-discrete scheme of the Allen–Cahn equation on the nonuniform mesh

$$\begin{aligned} \frac{U_j^{n+1} - U_j^n}{\Delta t} = & \frac{q}{4} \left(\left(\kappa_j^{-1} \delta^{-} \right) \left((x_\xi)_j^{-1} \kappa_j (x_\xi)_j^{-1} \delta^{+} \right) \left(U_j^{n+1} + U_j^n \right) \right. \\ & \left. + \left(\kappa_j^{-1} \delta^{+} \right) \left((x_\xi)_j^{-1} \kappa_j (x_\xi)_j^{-1} \delta^{-} \right) \left(U_j^{n+1} + U_j^n \right) \right) \\ & + \frac{p}{2} \left(U_j^{n+1} + U_j^n \right) + \frac{r}{4} \left(\left(U_j^n \right)^3 + \left(U_j^{n+1} \right)^3 \right. \\ & \left. + \left(U_j^n \right)^2 U_j^{n+1} + U_j^n \left(U_j^{n+1} \right)^2 \right). \end{aligned} \tag{35}$$

Next, we choose two pairs of initial solution $u_0(x)$ and parameter ε to verify the validity of the scheme.

Example 1 Consider the following initial value

$$u_0(x) = \cos(\pi x), \quad x \in [-2, 2], \quad t \in [0, 10].$$

We also set $\varepsilon = 0.01$, $\Delta t = 0.01$ and $\Delta x = 0.02$. In Fig. 3, two intuitive top views are given at $t = 0$ s and $t = 10$ s to show that the phase has been completely separated. The mesh trajectories and energy change are shown in Fig. 4 to indicate that the r-adaptive component of the adaptive algorithm performs well tracking the movement of the front throughout the time integration period, and the discrete energy drops very quickly in the first two seconds, and at $t = 3$ s it is at a stable state.

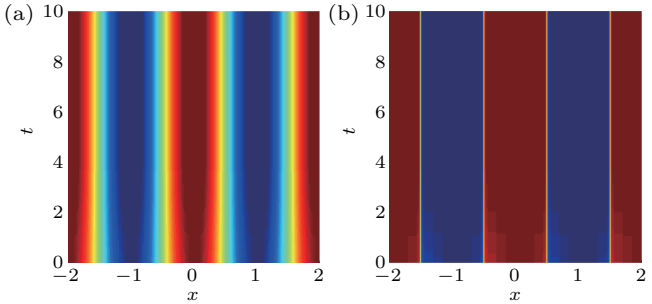


Fig. 3. Snapshots of (a) initial time $T = 0$ s and (b) terminus time $T = 10$ s.

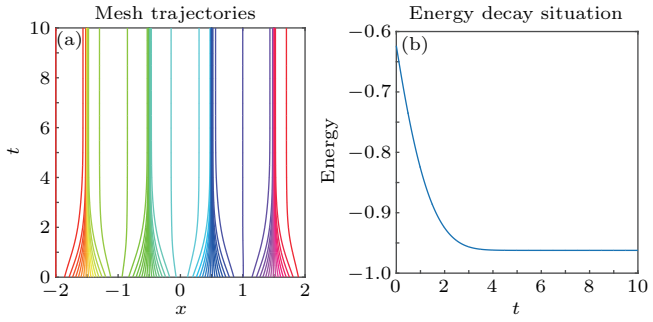


Fig. 4. (a) Grid movement - each line represents the path of one grid point in time. (b) AC equation's energy decay case.

Next, we test the accuracy of the proposed method by choosing a reference solution with a small time step $\Delta t = 5\times10^{-4}$. The L^2-norms and L^∞-norms of the errors of numerical solutions are presented respectively in Table 1. It indicates that the moving mesh method can achieve almost the same accuracy as the uniform method. Fixing space step $\Delta x = 0.02$ and setting time step Δt half each time, we obtain the numerical results shown in Table 2. Note that in the uniform mesh method, L^2 error can be kept the second-order convergence while it can not be in the moving mesh method. Referring to the previous literature, there is no convergence analysis for the moving mesh method. This is because even if the same number of grids are taken each time, the grid position will also bring a small change due to the r-algorithm. We also notice that in Table 2, there is no numerical result for the uniform mesh method when taking $\Delta t = 0.25$ or 0.125. This is because the traditional uniform mesh method is limited by the step ratio $\Delta x/\Delta t$. However, the moving mesh method can break through this limit and tend to perform well in a large step, although there is no theory to verify whether it is unconditionally stable. Another point we have to mention is that using the moving mesh method costs more computational time, especially when the space or time step is very small. This phenomenon is also caused by the r-refinement method. In the beginning, the physical domain Ω_p is mapped to Ω_c, and it needs to solve alternately between two domains all the time. In addition, we compare the total energy with different time steps with the reference solution for the L^∞ error, and the line chart is shown in Fig. 5. It is worth noting that when $\Delta t = 0.5$, the energy error reaches around magnitude 10^{-2}, and $\Delta t = 0.001$ corresponds to magnitude 10^{-6}. In addition, the error decreases very quickly before the time grid numbers reach 500. It again verifies the solution stability of large time steps and the advantage of the proposed scheme.

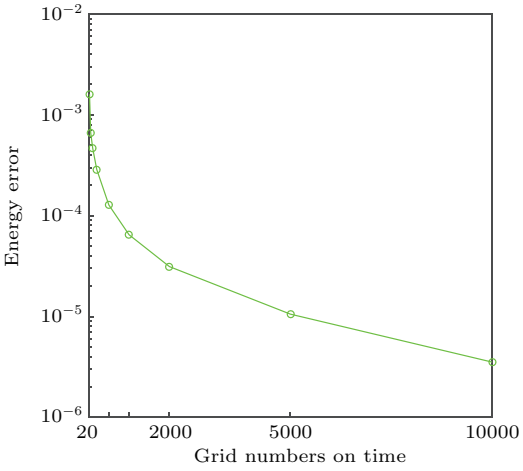


Fig. 5. Different time step's energy error with reference solution.

Table 1. Numerical results for the AC equation (example 1), $\Delta x = 0.01$.

Δt		0.02	0.01	0.005	0.0025	0.00125
Moving mesh	L_2	3.8199×10^{-8}	3.6399×10^{-8}	2.9785×10^{-9}	1.9676×10^{-9}	3.1184×10^{-10}
	L_∞	4.3083×10^{-9}	4.1181×10^{-9}	3.5176×10^{-10}	2.0455×10^{-10}	9.2236×10^{-11}
Uniform mesh	L_2	3.0717×10^{-8}	7.7141×10^{-9}	1.9288×10^{-9}	4.4559×10^{-10}	6.1532×10^{-11}
	L_∞	8.2032×10^{-9}	2.0601×10^{-9}	5.1509×10^{-10}	1.1900×10^{-10}	1.6429×10^{-11}

Table 2. The L^2 error and L^∞ error between the current time step and the previous time step (example 1), $\Delta x = 0.02$.

Δt		0.25	0.125	2.5×10^{-3}	1.25×10^{-3}	6.25×10^{-4}
Moving mesh	L_2	–	4.3664×10^{-5}	9.4351×10^{-6}	5.5875×10^{-6}	6.6842×10^{-7}
	L_∞	–	6.9082×10^{-6}	8.5375×10^{-7}	1.2860×10^{-7}	1.1814×10^{-8}
Uniform mesh	L_2	NaN	NaN	2.0421×10^{-7}	5.1419×10^{-8}	1.2963×10^{-8}
	L_∞	NaN	NaN	4.2206×10^{-8}	1.0627×10^{-8}	2.6790×10^{-9}

Example 2 Consider the following initial value:

$$u_0(x) = \sin^3(x),\ \ x\in[-4,4],\ \ t\in[0,20].$$

Four different parameters $\varepsilon^2 = 0.001,\ 0.002,\ 0.003$ and 0.004 are tested for this example. We also set $\Delta x = 0.02 = \Delta t$. Figure 6 shows the phase cases at the terminal time $T = 20$ s. Furthermore, in order to reflect the difference in phase change caused by different parameters, the magnified results of the four local areas are shown in Fig. 7. It can be seen that as the parameter ε becomes larger, the width of the interfacial becomes longer, and phase separation gets coarser. This is consistent with the actual phase separation. Figure 8 illustrates intuitively the parameter ε affects the steady-state energy value and energy dissipation rate and verifies Eqs. (2) and (3).

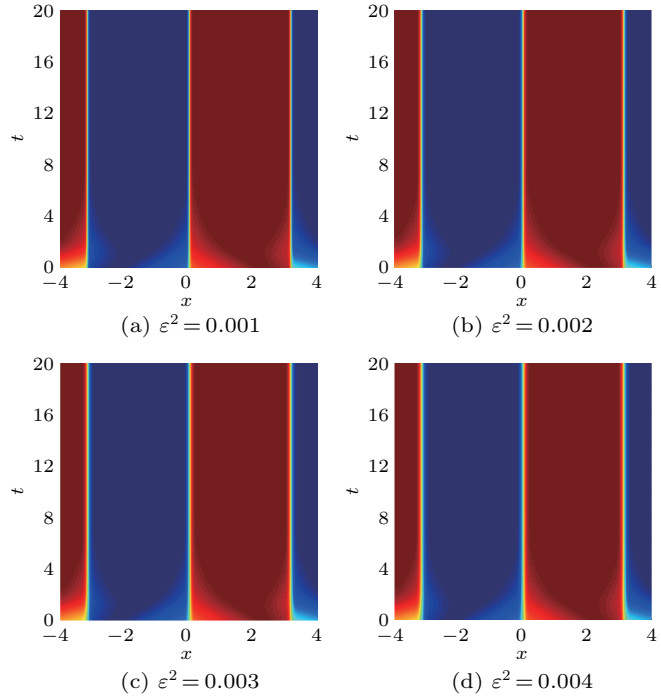


Fig. 6. Different parameter ε^2s' snapshots at $T = 20$ s.

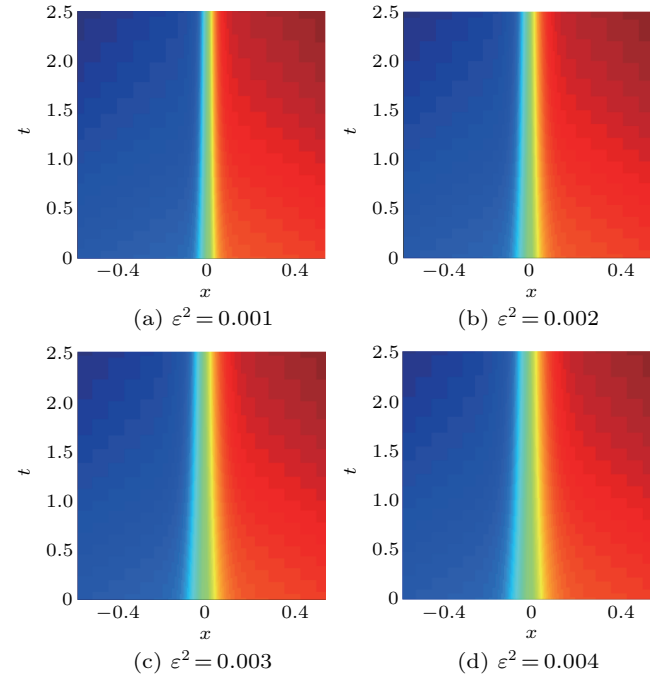


Fig. 7. Enlarged images in part of the phase area, with $x \in [-0.4, 0.4]$ and $T = 20$ s.

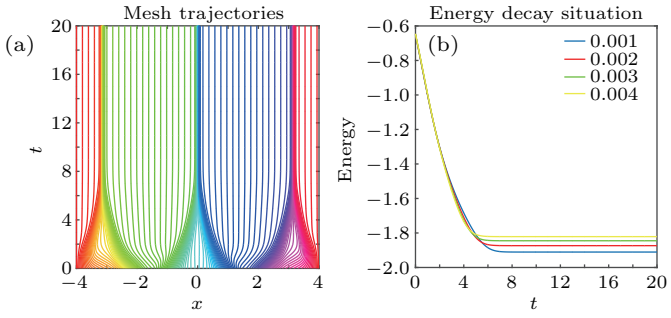


Fig. 8. (a) Grid movement with $\varepsilon^2 = 0.001$ and (b) different ε^2s' energy decay cases.

6. Conclusion and outlook

In this paper, we introduce a dissipative scheme and extend the average vector field method to nonuniform meshes. Our extension borrows from Yaguchi's thought that the dissipation properties obtained from the variational structure still holds after the change of coordinates. The de Boor's equidistribution principle and appropriate monitor function make the grid reasonably clustered in the region where the solution needs to get the effect of adaptive mesh. By using the rezoning approach, we get the discretization process of PDE solving. The numerical experiments confirm that the scheme can work well under the large step and step ratio $\Delta x/\Delta t$ that much less than 1. Although we have considered the only one-dimensional case, the scheme can also be derived for two-dimensional case by transforming the differential operators to

$$\frac{\partial u}{\partial x} = J^{-1}\left(\frac{\partial}{\partial \xi}\left(\frac{\partial y}{\partial \eta}u\right) - \frac{\partial}{\partial \eta}\left(\frac{\partial y}{\partial \xi}u\right)\right),$$
$$\frac{\partial u}{\partial y} = J^{-1}\left(\frac{\partial}{\partial \eta}\left(\frac{\partial x}{\partial \xi}u\right) - \frac{\partial}{\partial \xi}\left(\frac{\partial x}{\partial \eta}u\right)\right).$$

Since the quasi-Lagrange approach and rezoning approach each has its advantage, in the near future, we may propose a two-dimensional scheme by using the quasi-Lagrange approach and try to apply it to other phase-field models such as the Cahn-Hilliard equation and molecular beam epitaxy, etc.

References

[1] Allen S M and Cahn J W 1979 *Acta Metallurgica* **27** 1085
[2] Gokieli M and Marcinkowski L 2005 *Nonlinear Analysis: Theory, Methods & Applications* **63** e1143
[3] Benes M, Chalupecky V and Mikula K 2004 *Appl. Numer. Math.* **51** 187
[4] Feng X B and Prohl A 2003 *Numer. Math.* **94** 33
[5] Wheeler A A, Boettinger W J and McFadden G B 1992 *Phys. Rev. A* **45** 7424
[6] Wang S Y, Duan W G and Yin X Z 2013 *Chin. Phys. Lett.* **30** 110502
[7] Cai W J, Jiang C L, Wang Y S and Song Y Z 2019 *J. Comput. Phys.* **395** 166
[8] Zhu C S, Hu Z and Wang K M 2020 *Chin. Phys. B* **29** 034702
[9] Zhang H, Yan J Y, Qian X and Song S H 2021 *Appl. Numer. Math.* **161** 372
[10] Zhang H, Qian X, Yan J Y and Song S H 2020 *J. Comput. Phys.* **418** 109598
[11] Qian X, Fu H and Song S H 2017 *Appl. Math. Comput.* **307** 1
[12] Qian X, Fu H and Song S H 2017 *Adv. Appl. Math. Mech.* **9** 964
[13] Wang Y Y, Ding J H, Liu W B, Huang S S, Ke X Q, Wang Y Z, Zhang C and Zhao J J 2017 *Chin. Phys. B* **26** 026102
[14] Zhu C S, Han D and Xu S 2018 *Chin. Phys. B* **27** 094704
[15] Song Y H, Wang M T, Ni J, Jin J F and Zong Y P 2020 *Chin. Phys. B* **29** 128201
[16] Evans L C, Soner H M and Souganidis P E 1992 *Communications on Pure and Applied Mathematics* **45** 1097
[17] Quispel G R W and McLaren D I 2008 *J. Phys. A Math. Theor.* **41** 045206
[18] Gong Y Z, Cai J X and Wang Y S 2014 *J. Comput. Phys.* **279** 80
[19] Jiang C L, Sun J Q, Li H C and Wang Y F 2017 *Appl. Math. Comput.* **313** 144
[20] Zahr M J and Persson P O 2020 *Advanced Computational Methods for Aerodynamics 1* 6-10 January 2020, Orlando, FL, 0537
[21] Alharbi A and Naire S 2019 *J. Comput. Appl. Math.* **356** 219

[22] Yaguchi T, Matsuo T and Sugihara M 2010 *J. Comput. Phys.* **229** 4382
[23] Mackenzie J A and Mekwi W R 2020 *J. Comput. Appl. Math.* **364** 112320
[24] Furihata D and Matsuo T 2010 (New York: Chapman and Hall&CRC) p.103
[25] Huang W and Russell R D 2011 (New York: Springer) p. 83
[26] Huang W and Russell R D 1997 *SIAM J. Numer. Anal.* **34** 1106
[27] Celledoni E, Grimm V, McLachlan R I, McLaren D I, O'Neale D, Owren B and Quispel G R W 2012 *J. Comput. Phys.* **231** 6770
[28] Dahlby M and Owren B 2011 *SIAM J. Sci. Comput.* **33** 2318
[29] Matsuo T 2007 *J. Comput. Appl. Math.* **203** 32
[30] Eidnes S, Owren B and Ringholm T 2018 *Adv. Comput. Math.* **44** 815
[31] Mackenzie J A and Robertson M L 2000 *J. Comput. Phys.* **161** 537
[32] Miyatake Y and Matsuo T 2015 *J. Comput. Appl. Math.* **274** 79
[33] Eidnes S and Li L 2020 *SIAM J. Sci. Comput.* **42** 2865
[34] Zhang H, Qian X and Song S H 2020 *Appl. Math. Lett.* **102** 106091
[35] Zhu B B, Hu Z X, Tang Y F and Zhang R L 2016 *International Journal of Modeling, Simulation, and Scientific Computing* **7**1650008
[36] Ma Y P, Kong L H, Hong J L and Cao Y 2011 *Computers & Mathematics with Applications* **61** 319
[37] Shen J and Yang X F 2010 *Discrete & Continuous Dynamical Systems-A* **28** 1669
[38] Kong L H, Wei P, Hong Y Q, Zhang P and Wang P 2019 *Mathematical Methods in the Applied Sciences* **42** 3222