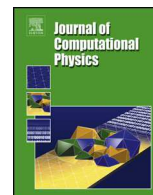




Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp



Numerical comparison of modified-energy stable SAV-type schemes and classical BDF methods on benchmark problems for the functionalized Cahn-Hilliard equation

Chenhui Zhang^a, Jie Ouyang^a, Cheng Wang^b, Steven M. Wise^{c,*}^a Department of Applied Mathematics, Northwestern Polytechnical University, Xi'an 710129, China^b Department of Mathematics, The University of Massachusetts, North Dartmouth, MA 02747, United States of America^c Department of Mathematics, The University of Tennessee, Knoxville, TN 37996, United States of America

ARTICLE INFO

Article history:

Received 9 March 2020

Received in revised form 4 August 2020

Accepted 11 August 2020

Available online 25 August 2020

Keywords:

FCH equation

SAV methods

Unconditional modified-energy stability

Fully implicit schemes

Meandering instability

Pearling instability

ABSTRACT

In this paper, we construct and test a class of linear numerical schemes for the Functionalized Cahn-Hilliard (FCH) equation with a symmetric double-well potential function by using a stabilized scalar auxiliary variable (SAV) method. To get a fair assessment of these new SAV-type schemes, we compare output with numerical solutions obtained by the classical, fully-implicit BDF1 and BDF2 schemes. We prove the unconditional unique solvability of the SAV systems and demonstrate the detailed steps used for finding the solutions to these systems. Two sixth-order constant-coefficient linear equations need to be solved at each time step for every SAV scheme. We also provide a theoretical analysis of the unconditional modified-energy stability for the schemes using the usual tools. The Fourier pseudo-spectral method is used as the spatial discretization. Several numerical tests are performed to verify the theoretical analyses and to compute some interesting problems that are physically relevant. Simulations of phase separation in 2D and 3D show the schemes can capture the correct qualitative dynamical behavior and, at the same time, the original physical FCH free energy is dissipated. The classical BDF1 (backward Euler) and BDF2 fully implicit methods, which have significantly smaller local truncation errors (LTEs), are used to repeat several numerical calculations and give a more objective measure of the accuracy and efficiency of the SAV schemes. To keep things simple and fair, for this preliminary battery of comparison tests, we use only fixed, uniform time step sizes. In this setting, the SAV schemes often have an advantage in terms of computational efficiency, being up to three times faster in CPU time when a relatively large time step size is used. However, when accuracy is counted in the measures of computational efficiency, the classical BDF methods often perform better than the linear SAV methods, with an advantage of up to three digits of precision. If the final target of a computation is a relatively high global accuracy, then the method with the least computational time to achieve that accuracy is very often classical BDF2. But, these conclusions are not universal; some test results are subtle and ambiguous. In any case, while SAV methods can be constructed in such a way that they are both energy stable and accurate, they are, however, not always a good choice

* Corresponding author.

E-mail addresses: chenhuizhang@mail.nwpu.edu.cn (C. Zhang), jieouyang@nwpu.edu.cn (J. Ouyang), cwang1@umassd.edu (C. Wang), swise1@utk.edu (S.M. Wise).

in practical, real-world computations, because their large LTEs can severely limit their true efficiency.

© 2020 Elsevier Inc. All rights reserved.

1. Introduction

An amphiphilic molecule is a compound whose head has a high affinity for a polar fluid (such as water) and whose tail has a high affinity for a non-polar fluid (like oil). Amphiphilic molecules can be added to an emulsion (a mixture of two immiscible fluids) to give the substance new properties, for example, to stabilize the mixture against coarsening. In this three-component system, the amphiphilic substance is often termed as an *emulsifier*. A common example is sodium stearate, a soap molecule, which when added to an oil-water emulsion, can trap oil droplets inside roughly spherical structures called micelles. This effectively makes the oil droplets water soluble, so that they can be washed away [11]. In this setting, each amphiphilic molecule, like the sodium stearate soap of the last example, resides on the interface separating the two immiscible fluids with its body aligned so that its head is adjacent to the polar fluid (water, e.g.) and its tail is adjacent to the non-polar fluid (oil, e.g.). Near equilibrium, the molecules pack tightly together to form a closed (edgeless), connected, single-layer sheet separating two immiscible fluids. Let us call this the *monolayer case*, for short.

Certain amphiphilic molecules can give rise to even more exotic structures. Some can produce a bilayer structure separating two miscible fluids. In particular, the fluids can be the same on either side of the interface, in which case the background fluid is simply termed as *the solvent*. This binary mixture of solvent and amphiphilic molecules is what we will be interested in modeling in the present paper, the *bilayer case*, for short. A prominent example is a vesicle, a biological structure where a lipid bilayer encloses the cytoplasm (internal fluid), separating it from the external fluid, which in our simple thought experiment is comprised of the same fluid.

In the bilayer case, the amphiphilic molecules form two distinct sheets or layers. The tail sides of the two sheets attract each other and stick together forming a two-layer sandwich, the bilayer. In contrast to the monolayer case, it is not necessary that the amphiphilic-molecule-populated interface is closed in the bilayer case. It can have edges or holes, at least in some reasonable sense. In general, the phase separation of an amphiphilic mixture can present a rich family of interface morphologies [32].

The functionalized Cahn-Hilliard (FCH) equation is a popular and important model for describing the diffusion and self-assembly of amphiphilic molecules in the bilayer case. The free energy for the model originated in [27] as a tool for characterizing phase separation of amphiphilic compounds in a background fluid (the solvent). The model was extended in [26,37] for capturing the dynamic formation of amphiphilic interface structures in a solvent. The FCH equation is the H^{-1} gradient flow with respect to the appropriate free energy, like that proposed in [27], so that the number of amphiphilic molecules is conserved and the free energy is dissipated as the simulation advances in time.

The FCH equation is a sixth-order nonlinear parabolic equation. Naive explicit or semi-implicit time stepping schemes may suffer from a truly severe time step restriction for the sake of numerical stability, in the worst case, $\delta t \leq Ch^6$, for some constant $C > 0$, where δt is the time step size and h is the space step size. For such schemes, an extremely small time step must often be used, even if a much larger time step would be allowed based on the consideration of accuracy alone.

In order to overcome this difficulty, many numerical schemes have been proposed for solving the FCH equation. A simple but efficient linear stabilization technique [12,34,36,41,43] was employed in [5] for solving the FCH equation. That method, which was also adopted in [31], was designed with two tunable stabilization parameters that must be selected through error trial. Choosing the stabilization parameters too large can lead to significant local truncation error, but the resulting semi-discrete system is a linear system that is easy to solve. Moreover, there is no theoretical justification to demonstrate this scheme is energy stable, although typically the total free energy from numerical calculations decays monotonically [5,31]. A fully implicit (essentially backward Euler) approach was adopted in [8]. The difficulty with methods of this type (that is, fully implicit BDF methods) is that a large nonlinear system of equations must be solved at each time step, and they require very sophisticated, efficient nonlinear solvers. Moreover, it is difficult to characterize the energy stability of such schemes rigorously, unless the time step size is very small. On the plus side, the schemes are very simple to state and, typically, the local truncation error is much smaller for classical (fully implicit) BDF methods than for semi-implicit methods of the same order of accuracy.

Another attractive technique is the so-called convex splitting approach [2,20,29,30,42,44], which is widely used for solving a variety of bistable gradient equations. In [21], the authors utilize this method to design a first-order convex splitting scheme for the FCH equation; it was proved that the method is both unconditionally energy stable and unconditionally uniquely solvable. Moreover, the energy stability in [21] is primitive, in the sense that it is with respect to the original (primitive) free energy of the system. This allows for the establishment of an optimal-order convergence result, which has also been proven in the paper. The downsides of the convex splitting method are (1) the local truncation error is significantly larger than that for BDF1 (backward Euler) and (2) a large nonlinear system must again be solved at each time step. An efficient solver was established in [21] for the scheme, based on the preconditioned steepest descent (PSD) method [22]. We will use a version of that method here but with the classical BDF time discretizations. In addition, we will show how to accelerate the performance of the solver compared to the computations in [21].

In this paper, we consider the scalar auxiliary variable (SAV) [28,38,40,39] method to construct some efficient, unconditionally modified-energy stable, linear schemes so that relatively large time steps can be used in the computation. The SAV method was first introduced in [39]. The main idea is to introduce a new (auxiliary) variable by adding an appropriate positive constant to the nonlinear part of the energy. This requires that the nonlinear energy part must be bounded from below so that the sum is not less than zero. The advantages of the SAV method have been pointed out in [40], including (1) it can be used to deal with a large class of gradient flows, (2) the resulting schemes (first- or second-order in time) are unconditionally modified-energy stable and linear, and (3) BDF1-like (BDF1-SAV), BDF2-like (BDF2-SAV), and Crank-Nicolson-like (CN-SAV) schemes are available in the methodology. The numerical error will become large if we use large time steps in the actual calculation, although we can choose any time step for the unconditionally modified-energy stable schemes. In this situation, the primal free energy of the system may not monotonically decrease, even though the modified free energy is dissipated. Fortunately, this numerical difficulty can be effectively overcome by adding stabilization terms, which will not cause any essential difficulties to the theoretical analysis of stability. An introduction of stabilization terms requires a delicate balance; they should be large enough to effect energy stability, but not so large that the local truncation error (LTE) is unacceptably inflated.

Speaking of local truncation error, the LTE for an SAV scheme is expected to be significantly larger than that of the corresponding purely implicit scheme, due to the auxiliary variables introduced. In other words, the LTE for BDF2-SAV is much greater than of the pure BDF2 method, et cetera. Thus, an important question arises: Is an SAV scheme more efficient than the corresponding purely implicit method to achieve a certain level of desired accuracy? On one hand the SAV scheme is linear, but it has a large LTE. The purely implicit scheme has a relatively smaller LTE, but requires a (potentially) complicated nonlinear equation to solve. In the case of the FCH problem, the nonlinear terms can be very complicated, which makes this problem a good one for computing such benchmarks. Coupled with a very efficient solver, like that used herein, it is possible that, contrary to expectations, the pure BDF schemes can win out. In fact, we will show some cases where this is the case. Here we focus only on the case of fixed, uniform time stepping. This gives a fair starting point for comparisons, since the SAV schemes are derived under such assumptions. However, since FCH problems have solutions that exhibit a rich diversity of time scales, ultimately adaptive time stepping algorithms should be compared. This is the subject of current work that is reported elsewhere.

We will observe that the SAV schemes often have an advantage in terms of computational efficiency, being up to three times faster in CPU time when a relatively large time step size is used. However, when accuracy is counted in the measures of computational efficiency, the classical BDF methods will often perform better than the linear SAV methods, with an advantage of up to three digits of precision. If the final target of a computation is relatively high global accuracy, then the method (of those tested) with the least computational time to achieve that desired accuracy is typically classical BDF2. But, these conclusions are not universal; some test results are subtle and ambiguous, as the reader will see. Still, our preliminary results should serve as a caution to those looking for the “best” methods. Stable and efficient does not always mean accurate.

The rest of the paper is organized as follows. In Section 2, we introduce the FCH free energy and the FCH equation. In Section 3, we reformulate the FCH energy using an auxiliary variable. This allows us to present a formally equivalent FCH dynamical system, with one additional equation for the auxiliary variable. In Section 4, we construct three unconditionally modified-energy stable schemes with first and second-order accuracy and show that each scheme is unconditionally uniquely solvable and unconditionally modified-energy stable. In Section 5, we apply our schemes to perform some numerical tests. First we demonstrate the expected temporal accuracies are achieved with some toy problems. Then we carry out some benchmark computations and make detailed comparisons of the solutions computed by the SAV schemes with the classical nonlinear, fully-implicit BDF1 and BDF2 schemes. Some conclusions are given in Section 6. The proofs of the modified-energy stabilities for the SAV schemes are left for Appendices A, B, and C.

2. The functionalized Cahn-Hilliard equation

A detailed derivation of the Functionalized Cahn-Hilliard equation can be found in [33]. Here, we give a brief recapitulation in the simplest setting, namely, where the homogeneous free energy density, F below, is symmetric, with equal well depths. It helps to begin with the classical Cahn-Hilliard (CH) equation, which models the phase separation and coarsening of a conserved binary mixture. This is the conserved (H^{-1}) gradient flow of the free energy functional

$$E_{CH}[\phi] = \int_{\Omega} \left(\frac{1}{2} \epsilon^2 |\nabla \phi|^2 + F(\phi) \right) d\mathbf{x}, \quad (2.1)$$

where F is the symmetric double-well function

$$F(\phi) = \frac{1}{4} (\phi^2 - 1)^2 \quad \text{and, therefore,} \quad F'(\phi) = \phi^3 - \phi; \quad (2.2)$$

$\phi(\mathbf{x})$ is the phase field variable, which represents the scaled volume fraction of one component; and $\epsilon > 0$ is a parameter that characterizes the thickness of the diffuse interface. The minima of F are, of course, located at $\phi = \pm 1$, which represent the pure states of the binary mixture. The classical Cahn-Hilliard equation is

$$\dot{\phi} = \Delta \mu_{\text{CH}},$$

where μ_{CH} is the chemical potential, that is, the variational derivative of E_{CH} with respect to ϕ :

$$\mu_{\text{CH}} = -\epsilon^2 \Delta \phi + F'(\phi). \quad (2.3)$$

Here, and throughout the text, we assume that ϕ satisfies periodic boundary conditions on the rectangular domain $\Omega \subset \mathbb{R}^d$, $d = 2, 3$. The FCH free energy takes the form [27,33]

$$E[\phi] = \int_{\Omega} \left\{ \frac{1}{2} \mu_{\text{CH}}^2 - \eta \left(\frac{\epsilon^2}{2} |\nabla \phi|^2 + F(\phi) \right) \right\} d\mathbf{x}, \quad (2.4)$$

where η is a parameter which describes the properties of amphiphilic polymer phase at the interface. Here, the phase field variable $\phi(\mathbf{x}, t)$ is used to distinguish different components of a binary mixture. More specifically, $\phi(\mathbf{x}, t) = +1$ is associated to the solvent phase and $\phi(\mathbf{x}, t) = -1$ is associated to the amphiphilic polymer phase. For the parameter η , when $\eta > 0$, (2.4) is the so-called FCH energy; when $\eta < 0$, (2.4) is the well-known Cahn-Hilliard-Willmore (CHW) energy (see a more recent work [7] for the numerical analysis of the energy stability for the CHW model). In this paper, we focus only on the FCH energy, i.e., $\eta > 0$.

The FCH equation is the H^{-1} gradient flow of the free energy $E[\phi]$:

$$\dot{\phi} = M \Delta \mu, \quad (2.5)$$

$$\mu = \epsilon^2 \Delta \omega - F''(\phi) \omega + \eta \omega, \quad (2.6)$$

$$\omega = \epsilon^2 \Delta \phi - F'(\phi), \quad (2.7)$$

where μ is the chemical potential; M is the mobility constant; and

$$F''(\phi) = 3\phi^2 - 1. \quad (2.8)$$

While we assume that periodic boundary conditions are enforced, for simplicity, other boundary conditions may be used. In the present case, the FCH equation is free energy dissipated, that is,

$$\dot{E}[\phi] = (M \Delta \mu, \mu)_{L^2} = -M \|\nabla \mu\|_{L^2}^2 \leq 0. \quad (2.9)$$

Our desire is to construct numerical schemes that can inherit this property, at least in some sense.

3. An auxiliary variable reformulation for the FCH equation

In order to construct unconditionally energy stable numerical schemes for the FCH equation by using the standard SAV method, the first thing that we need to do is to introduce an auxiliary variable in the free energy. Let us rewrite the free energy functional $E[\phi]$ in the following equivalent way:

$$E[\phi] = \int_{\Omega} \left\{ \frac{1}{2} \epsilon^4 (\Delta \phi)^2 - \left(1 + \frac{\eta}{2} \right) \epsilon^2 |\nabla \phi|^2 + G(\phi) \right\} d\mathbf{x}, \quad (3.1)$$

where

$$G(\phi) := 3\epsilon^2 \phi^2 |\nabla \phi|^2 + \frac{1}{2} (F'(\phi))^2 - \eta F(\phi).$$

Then we have

Lemma 3.1. *There is a finite real number $C_1 = C_1(\eta)$, such that, for every $\phi \in H_{\text{per}}^1$,*

$$G(\phi(\mathbf{x})) \geq C_1, \quad \forall \mathbf{x} \in \mathbb{R}^d.$$

There is a finite real number $C_2 = C_2(\eta)$, such that, for every $\phi \in H_{\text{per}}^2$,

$$E[\phi] \geq C_2.$$

In short, both $G(\phi)$ and $E[\phi]$ are bounded from below.

Proof. By using Young's inequality, we have

$$-\epsilon^2 \Delta \phi \cdot \left(1 + \frac{\eta}{2}\right) \phi \leq \frac{1}{2} (\epsilon^2 \Delta \phi)^2 + \frac{1}{2} \left(1 + \frac{\eta}{2}\right)^2 \phi^2. \quad (3.2)$$

Then, rewriting $E[\phi]$, we have

$$\begin{aligned} E[\phi] &= \int_{\Omega} \left\{ \frac{1}{2} \epsilon^4 (\Delta \phi)^2 + \left(1 + \frac{\eta}{2}\right) \epsilon^2 \phi \Delta \phi + G(\phi) \right\} d\mathbf{x} \\ &\geq \int_{\Omega} \left\{ G(\phi) - \frac{1}{2} \left(1 + \frac{\eta}{2}\right)^2 \phi^2 \right\} d\mathbf{x}. \end{aligned}$$

Defining

$$H(\phi) := \frac{1}{2} (F'(\phi))^2 - \eta F(\phi) \quad \text{and} \quad S(\phi) := H(\phi) - \frac{1}{2} \left(1 + \frac{\eta}{2}\right)^2 \phi^2,$$

and using equation (2.2), we find that

$$H(\phi) = \frac{1}{2} \phi^6 - \left(1 + \frac{1}{4} \eta\right) \phi^4 + \left(\frac{1}{2} + \frac{1}{2} \eta\right) \phi^2 - \frac{1}{4} \eta \geq C_1, \quad (3.3)$$

$$S(\phi) = \frac{1}{2} \phi^6 - \left(1 + \frac{1}{4} \eta\right) \phi^4 - \frac{\eta^2}{8} \phi^2 - \frac{1}{4} \eta \geq C_3, \quad (3.4)$$

for some finite constants C_1 and C_3 that clearly depend upon η . Finally,

$$G(\phi) = 3\epsilon^2 \phi^2 |\nabla \phi|^2 + H(\phi) \geq C_1, \quad (3.5)$$

$$E[\phi] \geq \int_{\Omega} \left\{ 3\epsilon^2 \phi^2 |\nabla \phi|^2 + S(\phi) \right\} d\mathbf{x} \geq \int_{\Omega} C_3 d\mathbf{x} = |\Omega| C_3 =: C_2, \quad (3.6)$$

and the results are proven. $\square$

We now introduce an auxiliary variable

$$U = \sqrt{\int_{\Omega} G(\phi) d\mathbf{x} + B} = \sqrt{\int_{\Omega} \left\{ 3\epsilon^2 \phi^2 |\nabla \phi|^2 + \frac{1}{2} (F'(\phi))^2 - \eta F(\phi) \right\} d\mathbf{x} + B}, \quad (3.7)$$

where B is a positive constant that ensures that

$$\int_{\Omega} G(\phi) d\mathbf{x} + B > 0.$$

This allows one to redefine the total free energy (3.1) as

$$E[\phi, U] = \int_{\Omega} \left\{ \frac{1}{2} \epsilon^4 (\Delta \phi)^2 - \left(1 + \frac{\eta}{2}\right) \epsilon^2 |\nabla \phi|^2 \right\} d\mathbf{x} + U^2 - B. \quad (3.8)$$

We call this the modified free energy. The original FCH dynamical equation (2.5)–(2.7) transforms into the following equivalent PDE system:

$$\dot{\phi} = M \Delta \mu, \quad (3.9)$$

$$\mu = \epsilon^4 \Delta^2 \phi + (2 + \eta) \epsilon^2 \Delta \phi + U X(\phi), \quad (3.10)$$

$$\dot{U} = \frac{1}{2} \int_{\Omega} X(\phi) \dot{\phi} d\mathbf{x}, \quad (3.11)$$

where

$$X(\phi) = \frac{6\epsilon^2 (\phi |\nabla \phi|^2 - \nabla \cdot (\phi^2 \nabla \phi)) + F'(\phi) F''(\phi) - \eta F'(\phi)}{\sqrt{\int_{\Omega} \left\{ 3\epsilon^2 \phi^2 |\nabla \phi|^2 + \frac{1}{2} (F'(\phi))^2 - \eta F(\phi) \right\} d\mathbf{x} + B}}. \quad (3.12)$$

The corresponding initial conditions of the new system are

$$\phi(t=0) = \phi^0 \quad \text{and} \quad U(t=0) = U(\phi^0), \quad (3.13)$$

with the periodic boundary conditions imposed. The scalar variable U requires no boundary condition.

This new system also admits an energy law, as we now show. Observe that

$$\dot{E}[\phi, U] = \int_{\Omega} \left\{ \epsilon^4 \Delta \phi \Delta \dot{\phi} - 2 \left(1 + \frac{\eta}{2} \right) \epsilon^2 \nabla \phi \nabla \dot{\phi} \right\} d\mathbf{x} + 2U\dot{U},$$

and

$$\begin{aligned} \dot{U} &= \frac{1}{2} \frac{1}{U} \int_{\Omega} \partial_t G(\phi) d\mathbf{x} \\ &= \frac{1}{2} \frac{1}{U} \int_{\Omega} \left\{ 6\epsilon^2 \phi \dot{\phi} |\nabla \phi|^2 + 6\epsilon^2 \phi^2 \nabla \phi \cdot \nabla \dot{\phi} + F'(\phi) F''(\phi) \dot{\phi} - \eta F'(\phi) \dot{\phi} \right\} d\mathbf{x} \\ &= \frac{1}{2} \int_{\Omega} X(\phi) \dot{\phi} d\mathbf{x}. \end{aligned}$$

Therefore, we conclude that

$$\dot{E}[\phi, U] = \int_{\Omega} \mu \dot{\phi} d\mathbf{x} = \int_{\Omega} \mu M \Delta \mu d\mathbf{x} = -M \|\nabla \mu\|_{L^2}^2 \leq 0, \quad (3.14)$$

where $E[\phi, U]$ is the modified free energy (3.8).

Remark 3.2. At the continuous time and continuous space levels, the energy law (3.14) of the auxiliary variable system is equivalent to the energy law (2.9) for the original system. The original physical energy of the system will be guaranteed to be dissipated (non-increasing) in time using either system. We can use the present auxiliary variable formulation to construct first- and second-order, linear, constant-coefficient, unconditionally modified-energy stable numerical schemes. But, while the SAV numerical schemes will admit some forms of energy dissipation laws, we will not be able to guarantee that the original physical (primal) energy (the energy written only in the primal variable ϕ) will be dissipated in time. This is one notable downside of the SAV methodology.

Computational experience has shown that the dissipation property for the original energy functional is preserved for the SAV schemes when the time step and stabilization parameters are chosen correctly. Meanwhile, many other standard and classical numerical schemes, such as linear implicit-explicit (IMEX) schemes, with appropriately chosen time step sizes, could also be able to preserve such a property. Moreover, it would be expected that the IMEX schemes would also have significantly smaller LTEs compared to the corresponding SAV schemes.

Remark 3.3. The auxiliary variable is introduced into the new equivalent PDE system (3.9)–(3.11) in order to overcome the difficulty of dealing directly with the nonlinear part within the FCH equation. However, this comes at a price. The discretized version of equation (3.11) will give rise, potentially, to extra terms in the local truncation error.

4. SAV and fully implicit schemes for the FCH equation

In this section, we present three unconditionally modified-energy stable numerical schemes for solving the FCH equation by using the SAV method. We also present the simple, but elegant, classical, fully implicit BDF1 (Backward-Euler) and BDF2 schemes. These fully implicit schemes are nonlinear, requiring sophisticated solvers to make these schemes efficient. Luckily, such nonlinear solvers are available, and we give the details of one, the PSD algorithm, herein.

Like the Invariant Energy Quadratization (IEQ) method, the SAV schemes will be linear, and will be stable in terms of some consistent modified energy. The advantage of the SAV over IEQ is that the equations defining the scheme will be linear with constant coefficients, and thus efficiently invertible using FFT methods. In contrast with convex splitting methods, such an energy stability is not primal. The energy that is dissipated for the SAV schemes is a modified, but consistent form of the free energy. On the other hand, SAV schemes are defined by linear constant-coefficient operators, whereas the convex splitting schemes are generally nonlinear and generally of low order.

In general, one can use the SAV method to design the first- and second-order unconditionally energy stable schemes for the phase field equations. And higher-order schemes can also be designed by using this method in some cases, and we refer the readers to [40] for more details. Here we use the SAV method to construct a first-order scheme and two different numerical schemes with second-order-in-time accuracy.

It should be noted that we add stabilization terms in each of the SAV schemes, which means that these schemes are not of standard SAV type. These additional stabilization terms will not cause any difficulty in the theoretical analysis of stability. We will explain the reasoning for adding the stabilization terms in the discussions of the numerical results in Section 5. No stabilization terms are needed for the fully implicit BDF schemes.

4.1. A BDF1-SAV scheme

Now we present an SAV scheme that is related to the backward Euler (BDF1) method. We refer to this method by the acronym BDF1-SAV. Assuming that ϕ^n and U^n are known, we update the value of ϕ^{n+1} and U^{n+1} according to the following first order numerical scheme:

$$\frac{\phi^{n+1} - \phi^n}{\delta t} = M \Delta \mu^{n+1}, \quad (4.1a)$$

$$\mu^{n+1} = \epsilon^4 \Delta^2 \phi^{n+1} + (2 + \eta) \epsilon^2 \Delta \phi^n + X^n U^{n+1} - S \Delta (\phi^{n+1} - \phi^n), \quad (4.1b)$$

$$\frac{U^{n+1} - U^n}{\delta t} = \frac{1}{2} \int_{\Omega} X^n \frac{\phi^{n+1} - \phi^n}{\delta t} d\mathbf{x}, \quad (4.1c)$$

where $S \geq 0$ is a stabilization parameter.

There are two main advantages of the SAV method: (1) this method is easy to implement; (2) systems to be inverted in the scheme are linear, positive, and have constant coefficients. Now we will explain the main points involved in solving the semi-discrete linear system (4.1). For more details, see [39] and [40]. The method described here to solve the scheme (4.1) can be modified for the other two forthcoming second-order schemes that we will propose in Sections 4.2 and 4.3.

First, (4.1c) implies that

$$U^{n+1} = U^n + \frac{1}{2} (X^n, \phi^{n+1}) - \frac{1}{2} (X^n, \phi^n). \quad (4.2)$$

Substituting (4.2) into equation (4.1b) and eliminating the auxiliary variable U^{n+1} , we get

$$\begin{aligned} \mu^{n+1} &= \epsilon^4 \Delta^2 \phi^{n+1} - S \Delta (\phi^{n+1} - \phi^n) + (2 + \eta) \epsilon^2 \Delta \phi^n \\ &\quad + U^n X^n + \frac{1}{2} (X^n, \phi^{n+1}) X^n - \frac{1}{2} (X^n, \phi^n) X^n. \end{aligned} \quad (4.3)$$

In combination with (4.1a) and (4.3) and regrouping the result, we are left with

$$\frac{\phi^{n+1}}{M \delta t} - \epsilon^4 \Delta^3 \phi^{n+1} + S \Delta^2 \phi^{n+1} - \frac{1}{2} (X^n, \phi^{n+1}) \Delta X^n = f^n, \quad (4.4)$$

where

$$f^n := \frac{\phi^n}{M \delta t} + S \Delta^2 \phi^n + (2 + \eta) \epsilon^2 \Delta^2 \phi^n + U^n \Delta X^n - \frac{1}{2} (X^n, \phi^n) \Delta X^n. \quad (4.5)$$

Next, defining a linear, positive, constant-coefficient differential operator

$$\mathcal{L}\psi := \frac{\psi}{M \delta t} - \epsilon^4 \Delta^3 \psi + S \Delta^2 \psi,$$

assuming periodic boundary conditions, as usual, we find

$$\mathcal{L}\phi^{n+1} - \frac{1}{2} (X^n, \phi^{n+1}) \Delta X^n = f^n. \quad (4.6)$$

By applying operator $\mathcal{L}^{-1}$ to the above equation, we get

$$\phi^{n+1} = \mathcal{L}^{-1} f^n + \frac{1}{2} (X^n, \phi^{n+1}) \mathcal{L}^{-1} (\Delta X^n). \quad (4.7)$$

The most straightforward way to calculate (X^n, ϕ^{n+1}) is to take the L^2 inner product of (4.7) with X^n :

$$(X^n, \phi^{n+1}) - \frac{1}{2} (X^n, \phi^{n+1}) (X^n, \mathcal{L}^{-1} (\Delta X^n)) = (X^n, \mathcal{L}^{-1} f^n). \quad (4.8)$$

We then obtain from (4.8) that

$$(X^n, \phi^{n+1}) = \frac{(X^n, \mathcal{L}^{-1} f^n)}{1 + \frac{1}{2} (X^n, \mathcal{L}^{-1} (-\Delta) X^n)}, \quad (4.9)$$

which is well-defined since $\mathcal{L}^{-1} (-\Delta)$ is a positive, symmetric operator. By substituting formula (4.9) into equation (4.7), we finally get an explicit expression for ϕ^{n+1} . We have proven the following:

Theorem 4.1. *The scheme (4.1) is unconditionally uniquely solvable.*

We now recap the implementation procedure to solve the linear and coupled system (4.1). If the values of ϕ^n and U^n have been known, we will determine ϕ^{n+1} in the following steps:

- Step 1. Compute X^n via (3.12) and f^n via (4.5).
- Step 2. Compute $(X^n, \mathcal{L}^{-1} f^n)$ and $(X^n, \mathcal{L}^{-1} \Delta X^n)$. This can be done by solving two sixth-order equations with constant coefficients.
- Step 3. Compute (X^n, ϕ^{n+1}) . This can be done easily by using (4.9).
- Step 4. Compute ϕ^{n+1} by (4.7).

The procedure described above for solving ϕ^{n+1} is straightforward to implement, though it does involve a number of FFT/IFFT applications in practice. Specifically, at each time step, we invert two linear, positive, sixth-order operators with constant coefficients and periodic boundary conditions.

Theorem 4.2. *The scheme (4.1) is unconditionally modified-energy stable in the sense that it satisfies the following discrete modified-energy law:*

$$E_1(\phi^{n+1}, U^{n+1}) \leq E_1(\phi^n, U^n), \quad \forall n \geq 0, \quad (4.10)$$

where

$$E_1(\phi^{n+1}, U^{n+1}) := \frac{1}{2} \epsilon^4 \|\Delta \phi^{n+1}\|^2 - \frac{2+\eta}{2} \epsilon^2 \|\nabla \phi^{n+1}\|^2 + |U^{n+1}|^2 - B$$

is the modified energy at time level t^{n+1} .

The proof is given in Appendix A. See our earlier remark, Remark 3.2, regarding the context of this “energy stability” result.

4.2. The BDF2-SAV scheme

One second-order accurate scheme that we describe is based on a combination of the second-order backwards-difference formula (BDF2) approximation for the time derivative and the SAV methodology. We refer to this method by the acronym BDF2-SAV. Assuming that ϕ^{n-1} , ϕ^n , U^{n-1} and U^n are known, we update ϕ^{n+1} and U^{n+1} according to the following BDF2-SAV scheme:

$$\frac{3\phi^{n+1} - 4\phi^n + \phi^{n-1}}{2\delta t} = M \Delta \mu^{n+1}, \quad (4.11a)$$

$$\mu^{n+1} = \epsilon^4 \Delta^2 \phi^{n+1} + (2+\eta) \epsilon^2 \Delta \phi^{*,n+1} + X^{*,n+1} U^{n+1} - S \Delta (\phi^{n+1} - 2\phi^n + \phi^{n-1}), \quad (4.11b)$$

$$\frac{3U^{n+1} - 4U^n + U^{n-1}}{2\delta t} = \frac{1}{2} \int_{\Omega} X^{*,n+1} \frac{3\phi^{n+1} - 4\phi^n + \phi^{n-1}}{2\delta t} d\mathbf{x}, \quad (4.11c)$$

where the extrapolated variables are defined via

$$\phi^{*,n+1} := 2\phi^n - \phi^{n-1} \quad \text{and} \quad X^{*,n+1} := 2X^n - X^{n-1};$$

and $S \geq 0$ is a stabilization parameter.

Theorem 4.3. *The scheme (4.11) is unconditionally uniquely solvable and unconditionally modified-energy stable in the sense that it satisfies the following discrete modified-energy law:*

$$E_2(\phi^{n+1}, \phi^n, U^{n+1}, U^n) \leq E_2(\phi^n, \phi^{n-1}, U^n, U^{n-1}), \quad \forall n \geq 0, \quad (4.12)$$

where

$$\begin{aligned} E_2(\phi^{n+1}, \phi^n, U^{n+1}, U^n) := & \frac{1}{2} \epsilon^4 \frac{\|\Delta \phi^{n+1}\|^2 + \|2\Delta \phi^{n+1} - \Delta \phi^n\|^2}{2} + S \frac{\|\nabla \phi^{n+1} - \nabla \phi^n\|^2}{2} \\ & - \frac{2+\eta}{2} \epsilon^2 \frac{\|\nabla \phi^{n+1}\|^2 + \|2\nabla \phi^{n+1} - \nabla \phi^n\|^2}{2} \\ & + \frac{|U^{n+1}|^2 + |2U^{n+1} - U^n|^2}{2} + \frac{2+\eta}{2} \epsilon^2 \|\nabla \phi^{n+1} - \nabla \phi^n\|^2 - B \end{aligned}$$

is the modified energy at time level t^{n+1} .

The theorem, whose proof is given in Appendix B, guarantees that the modified energy is dissipated. But, recalling our earlier remark, Remark 3.2, it may be that the original primal energy can actually increase if the time step is large enough. In fact, our own experiments show that this can be the case. Ultimately, the reader should keep in mind that, while stability is important, stability without accuracy, is typically numerically meaningless. Accuracy and stability are what is desired.

4.3. The CN-SAV scheme

The next second-order accurate scheme is based on the Crank-Nicolson approach in combination with the SAV strategy. We refer to this method by the acronym CN-SAV. Assuming that ϕ^{n-1} , ϕ^n , U^{n-1} and U^n are known, we update ϕ^{n+1} and U^{n+1} according to the following CN-SAV scheme:

$$\frac{\phi^{n+1} - \phi^n}{\delta t} = M \Delta \mu^{n+\frac{1}{2}}, \quad (4.13a)$$

$$\mu^{n+\frac{1}{2}} = \epsilon^4 \Delta^2 \frac{\phi^{n+1} + \phi^n}{2} + (2 + \eta) \epsilon^2 \Delta \phi^{\circ, n+\frac{1}{2}} + X^{\circ, n+\frac{1}{2}} \frac{U^{n+1} + U^n}{2} - S \Delta (\phi^{n+1} - 2\phi^n + \phi^{n-1}), \quad (4.13b)$$

$$\frac{U^{n+1} - U^n}{\delta t} = \frac{1}{2} \int_{\Omega} X^{\circ, n+\frac{1}{2}} \frac{\phi^{n+1} - \phi^n}{\delta t} d\mathbf{x}, \quad (4.13c)$$

where the extrapolated variables are defined via

$$\phi^{\circ, n+\frac{1}{2}} := \frac{3}{2}\phi^n - \frac{1}{2}\phi^{n-1} \quad \text{and} \quad X^{\circ, n+\frac{1}{2}} := \frac{3}{2}X^n - \frac{1}{2}X^{n-1};$$

and $S \geq 0$ is a stabilization parameter.

The proof of the following result is similar to those already encountered and is given in Appendix C.

Theorem 4.4. *The scheme (4.13) is unconditionally uniquely solvable and unconditionally energy stable in the sense that it satisfies the following discrete modified-energy law:*

$$E_3(\phi^{n+1}, \phi^n, U^{n+1}) \leq E_3(\phi^n, \phi^{n-1}, U^n), \quad \forall n \geq 0, \quad (4.14)$$

where

$$\begin{aligned} E_3(\phi^{n+1}, \phi^n, U^{n+1}) = & \frac{1}{2} \epsilon^4 \|\Delta \phi^{n+1}\|^2 + S \frac{\|\nabla \phi^{n+1} - \nabla \phi^n\|^2}{2} + |U^{n+1}|^2 \\ & - \frac{2 + \eta}{2} \epsilon^2 \left(\|\nabla \phi^{n+1}\|^2 - \frac{1}{2} \|\nabla \phi^{n+1} - \nabla \phi^n\|^2 \right) - B \end{aligned}$$

is the modified energy at time level t^{n+1} .

4.4. Fully implicit BDF1 and BDF2 schemes

The Backward-Euler (BDF1) scheme and the pure second-order Backward Differentiation Formula (BDF2) scheme are very easy to construct. In general, they are not provably unconditionally energy stable nor are they provably unconditionally uniquely solvable, that is, these properties may not hold for all possible times step sizes. However, if the time step size, δt , is sufficiently small, both stability and solvability will hold. See, for example, the classical textbook [1] and a few other related works [3, 13–19]. This lack of unconditional solvability/stability is not a big issue in practice, since, in our experience, the time step size restrictions for these properties to hold are less “restrictive” than those for accuracy. In addition, a few modified BDF schemes have been proposed for various gradient flows, such as Cahn-Hilliard and epitaxial thin film growth [23, 35, 45], in which certain artificial regularization is added so that an unconditional energy stability could be theoretically proved.

For the FCH equation, the BDF1 scheme [8] is

$$\frac{\phi^{n+1} - \phi^n}{\delta t} = M \Delta \mu^{n+1}, \quad (4.15a)$$

$$\mu^{n+1} = (\epsilon^2 \Delta - F''(\phi^{n+1}) + \eta)(\epsilon^2 \Delta \phi^{n+1} - F'(\phi^{n+1})). \quad (4.15b)$$

The related fully implicit BDF2 scheme is

$$\frac{3\phi^{n+1} - 4\phi^n + \phi^{n-1}}{2\delta t} = M \Delta \mu^{n+1}, \quad (4.16a)$$

$$\mu^{n+1} = (\epsilon^2 \Delta - F''(\phi^{n+1}) + \eta)(\epsilon^2 \Delta \phi^{n+1} - F'(\phi^{n+1})). \quad (4.16b)$$

Both the BDF1 and BDF2 schemes result in highly nonlinear sixth-order elliptic equations that are difficult to solve. The preconditioned steepest descent (PSD) algorithm, which was introduced in [22], is employed in this paper to solve all the nonlinear equations and is very easy to implement. For more details about the PSD algorithm, refer to [21–23]. There are some other efficient methods that can also be used to solve these nonlinear equations. For example, the Newton-Raphson method is applied in [8] to solve the BDF1 scheme.

Here, we take the BDF1 scheme as an example to show the implementation of the PSD iterative algorithm. We first rewrite scheme (4.15) as

$$\mathcal{N}[\phi^{n+1}] := 0, \quad (4.17)$$

where

$$\mathcal{N}[\phi] := \frac{1}{M\delta t}(-\Delta)^{-1}(\phi - \phi^n) + (\epsilon^2 \Delta - F''(\phi) + \eta)(\epsilon^2 \Delta \phi - F'(\phi)). \quad (4.18)$$

The mean-zero projection operator, $\langle \cdot \rangle_0$, is defined as

$$\langle u \rangle_0 := u - \frac{1}{|\Omega|} \int_{\Omega} u \, d\mathbf{x}. \quad (4.19)$$

Clearly, $\langle u \rangle_0$ is a mean zero function for any u . We next choose a linear, positive, and symmetric *preconditioner* operator, $\mathcal{P}$, and we obtain the (mean zero) *search direction*, d_k by solving

$$\mathcal{P}[d_k] = \langle -\mathcal{N}[\phi_k^{n+1}] \rangle_0, \quad (4.20)$$

where ϕ_k^{n+1} ($k = 0, 1, \dots$) is the k -th iterative approximation of ϕ^{n+1} with $\phi_0^{n+1} = \phi^n$. For the FCH equation we choose the preconditioner

$$\mathcal{P}[\psi] := -\frac{1}{M\delta t} \Delta^{-1} \psi + \psi + \epsilon^4 \Delta^2 \psi, \quad (4.21)$$

which is well-defined for any mean zero function, ψ .

Note that equation (4.20) can be solved efficiently by using the FFT when periodic boundary conditions are enforced. Once we get the mean-zero search direction, d_k , we update the $(k+1)$ -st approximation of ϕ^{n+1} according to the formula

$$\phi_{k+1}^{n+1} = \phi_k^{n+1} + \alpha d_k. \quad (4.22)$$

Choosing α by line search [22] – in other words, line optimization – one obtains the preconditioned steepest descent (PSD) method. If the δt is sufficiently small, we can prove that $\lim_{k \rightarrow \infty} \phi_k^{n+1} = \phi^{n+1}$.

In order to speed up the PSD solver, we use a constant, invariant step size, $\alpha = 0.72$, in equation (4.22), which avoids a costly line search. This value is obtained after much numerical experimentation. It can be proven that the method still converges with a fixed step size α at a geometric rate, but slightly more slowly than with exact line search. However, the savings incurred by avoiding the line search more than makes up the difference, in most cases. The solver algorithm detailed here has the more appropriate name *preconditioned steepest descent with approximate line search (PSD-ALS)*. We will just refer to it as the PSD solver, for brevity, but see, for example, [6].

For the BDF2 scheme (4.16), we choose the preconditioner

$$\mathcal{P}[\psi] := \frac{3}{2M\delta t}(-\Delta)^{-1} \psi + \psi + \epsilon^4 \Delta^2 \psi, \quad (4.23)$$

and compute the search direction d_k by the counterpart of (4.20) appropriate for the BDF2 scheme. Finally, according to expression (4.22), one can get the solution of scheme BDF2 at time level $n+1$.

Remark 4.5. We wish to point out that we have not gone to great lengths in this manuscript to optimize the PSD solver. It is possible, we now know, to significantly improve the performance of the PSD solver for the present FCH problem by choosing a more general preconditioner (for the BDF1 case)

$$\mathcal{P}[\psi] := -\frac{1}{M\delta t} \Delta^{-1} \psi + \beta \psi + \epsilon^4 \Delta^2 \psi, \quad (4.24)$$

where $\beta \geq 0$ is a tunable parameter, and optimizing the performance of the solver (for particular fixed values of δt and M) over reasonable values of the PSD step size α and the preconditioning parameter β . This optimization/tuning will be reported in a subsequent work and will make the BDF1-PSD and BDF2-PSD algorithms even more competitive.

Table 1

The L^2 errors at $t = 0.1$ for the given exact solution (5.2) with parameters (5.1) calculated respectively by BDF1-SAV, BDF2-SAV and CN-SAV. The expected global accuracies are achieved, on the assumption that there is negligible spatial error.

δt	BDF1-SAV	Order	BDF2-SAV	Order	CN-SAV	Order
2.00000×10^{-2}	7.93e-4	-	2.41e-4	-	1.41e-4	-
1.00000×10^{-2}	4.38e-4	0.86	6.98e-5	1.79	2.69e-5	2.39
5.00000×10^{-3}	2.48e-4	0.82	2.23e-5	1.65	9.10e-6	1.56
2.50000×10^{-3}	1.40e-4	0.82	6.63e-6	1.75	3.00e-6	1.60
1.25000×10^{-3}	7.72e-5	0.86	1.78e-6	1.90	8.71e-7	1.78
6.25000×10^{-4}	4.14e-5	0.90	4.56e-7	1.96	2.32e-7	1.91
3.12500×10^{-4}	2.16e-5	0.94	1.15e-7	1.99	5.98e-8	1.96
1.56250×10^{-4}	1.11e-5	0.96	2.87e-8	2.00	1.51e-8	1.99
7.81250×10^{-5}	5.61e-6	0.98	7.18e-9	2.00	3.81e-9	1.99
3.90625×10^{-5}	2.82e-6	0.99	1.80e-9	2.00	9.55e-10	2.00
1.95313×10^{-5}	1.42e-6	0.99	4.51e-10	2.00	2.38e-10	2.00
9.76563×10^{-6}	7.10e-7	1.00	1.15e-10	1.97	5.76e-11	2.05

5. Numerical simulations

In this section, we test our SAV and classical BDF schemes. For simplicity and for minimizing the errors caused by spatial discretization, we adopt the highly accurate Fourier spectral method with N^2 (or N^3) Fourier modes to perform spatial discretization in 2D space (or in 3D space). Certainly, other spatial discretization methods, such as finite difference method [21], finite element method [24], and finite volume method [46], also can be used. We perform these simulations in $\Omega = [0, L]^d$, ($d = 2, 3$) and use periodic boundary conditions. The following parameters will be used for our numerical schemes in most cases,

$$M = 1, \epsilon = 0.1, \eta = \epsilon^2, L = 4\pi, N = 2^8, B = L^5. \quad (5.1)$$

For the SAV stabilization parameters, $S = \epsilon^2$ is used in the BDF1-SAV scheme and $S = 3\epsilon^2$ is used in the BDF2-SAV and CN-SAV schemes, for all tests in this paper. Some of the above parameters (5.1) may change for some special cases, which we mention, as appropriate.

5.1. SAV accuracy and stabilization tests

Since the SAV schemes are new, we test them on toy problems to confirm the expected orders of accuracy. In particular, these schemes need some calibration due to the use of the stabilization terms. We use the function

$$\phi(x, y, t) = \sin(x)\cos(y)\cos(t) \quad (5.2)$$

as the exact solution of the FCH equation (3.9)–(3.11) by adding appropriate forcing terms to the equations. In Table 1, we list the L^2 errors at $t = 0.1$ and convergence orders of the above three energy stable numerical schemes (i.e., BDF1-SAV, BDF2-SAV and CN-SAV). Table 1 shows that BDF1-SAV can attain first order accuracy and BDF2-SAV as well as CN-SAV can attain second order accuracy in time, respectively. Further, we find that the L^2 errors resulting from CN-SAV are slightly smaller than that of BDF2-SAV, as may be expected, although both of them are second-order accurate.

Table 1 shows us good accuracy and convergence order of each SAV scheme, which should be credited to the introduction of the appropriate stabilization parameter, S . The exact profile (5.2) is used to do some studies on the stabilization parameter S . In the BDF1-SAV scheme, we use the different values $S = 0, \epsilon^2, 3\epsilon^2, 5\epsilon^2, 10\epsilon^2$, to test the effect of the stabilization term on the accuracy for FCH equation. The L^2 errors calculated using different S in the BDF1-SAV scheme at $t = 0.1$ are presented in Fig. 1. Obviously, the stabilization term can improve the numerical stability of the BDF1-SAV scheme when the large time step is used. However, the effects are subtle for the BDF1-SAV scheme, because the pure un-stabilized BDF1-SAV scheme (i.e., $S = 0$) also has good accuracy when $\delta t \leq 1.0 \times 10^{-3}$.

Next, we apply the two second-order schemes, BDF2-SAV and CN-SAV, to perform the same test. In Fig. 2, the role of the stabilization term on accuracy is pretty obvious for both the BDF2-SAV and CN-SAV schemes. In the BDF2-SAV scheme with $S = 0$, we cannot get accurate numerical results for time step size $\delta t > 5.0 \times 10^{-4}$. But, the stabilization term can help us get good approximate solutions for the FCH equation, and it doesn't increase the errors significantly when small time steps are used. As can be seen from Fig. 2(c), the pure un-stabilized CN-SAV scheme ($S = 0$) doesn't work, and the approximations given by it are non-physical, even when a very small time step is used although the scheme itself is unconditionally energy stable. However, when the stabilization term is added to this scheme, its accuracy becomes satisfactory and is a little better than the BDF2-SAV scheme under the same conditions, according to Table 1.

We find that the BDF1-SAV scheme can show the best accuracy when $S = \epsilon^2$ and that both the BDF2-SAV and CN-SAV schemes can show the best accuracy when $S = 3\epsilon^2$. Hence, the following computations are going to be based on these values.

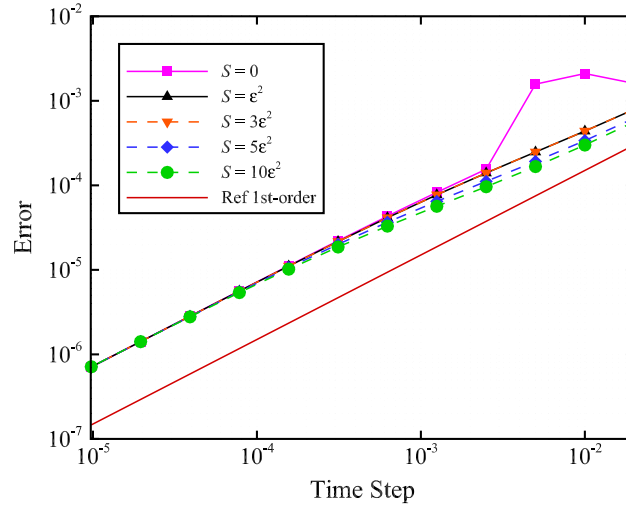


Fig. 1. The L^2 errors corresponding to different values of S at time $t = 0.1$ computed by using the BDF1-SAV scheme.

Table 2

The L^2 errors at $t = 4$ and $t = 10$ calculated by schemes BDF1-SAV, BDF2-SAV, CN-SAV, BDF1 and BDF2, and the corresponding total CPU time of each scheme, by using the fixed time step $\delta t = 1 \times 10^{-3}$ with the initial condition (5.3) and parameters (5.4). See Fig. 4.

Scheme	$t = 4$		$t = 10$	
	Error	CPU Time	Error	CPU Time
BDF1-SAV	4.99e-3	15.12	4.03e-3	37.21
BDF2-SAV	8.77e-5	15.36	7.42e-5	37.33
CN-SAV	5.26e-5	16.97	4.29e-5	41.68
BDF1	8.81e-5	56.73	7.65e-5	105.62
BDF2	5.70e-7	53.12	2.07e-7	99.21

Remark 5.1. The main reason that large time steps can be used in all the above SAV schemes is due to the introduction of the stabilization term. It is well known that we can not use any time step size δt we want in the actual calculations, because accuracy concerns limit the size. Larger numerical errors will be introduced by larger time steps. In any case, the stabilization term is quite useful for allowing us to use large time steps if needed. This becomes more of an issue in adaptive time stepping, which we have chosen to ignore in the present manuscript for the sake of simplicity and brevity.

Remark 5.2. Extra numerical error is introduced by adding the stabilization term in the SAV scheme. Large stabilization parameters generally lead to large local truncation errors (LTEs). Therefore, the user must be careful to strike a balance between stability and accuracy.

5.2. Sample computations and benchmark problems

We use a benchmark problem to demonstrate the efficiency and accuracy of our schemes, especially the second order schemes. The initial condition, which can be found in [4,8,21], is specified as follows,

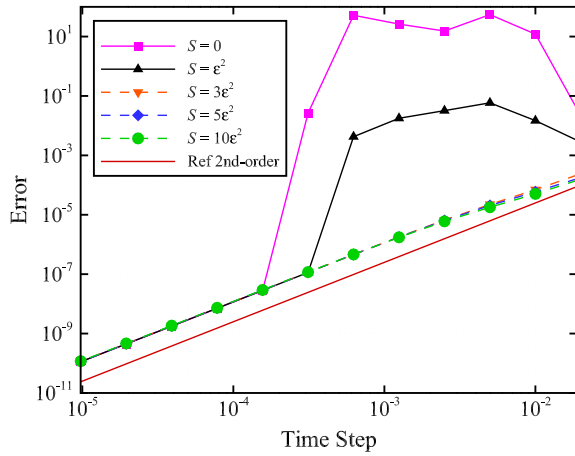
$$\phi(x, y, t = 0) = 2 \exp(\sin x + \sin y - 2) + 2.2 \exp(-\sin x - \sin y - 2) - 1, \quad (5.3)$$

and we use the following parameters for this problem:

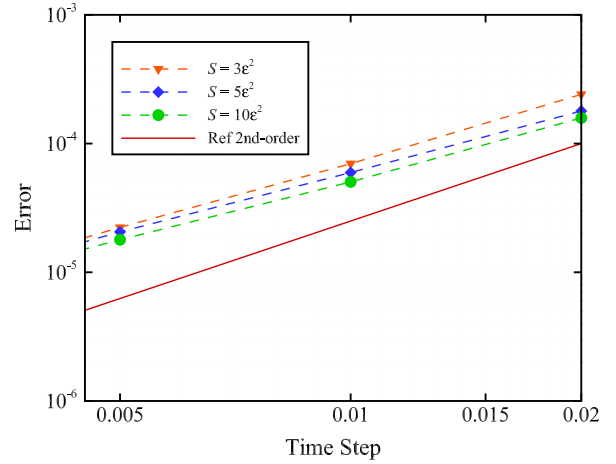
$$M = 1, \epsilon = 0.18, \eta = \epsilon^2, L = 2\pi, N = 2^7, B = L^5. \quad (5.4)$$

For the initial condition (5.3) and parameters (5.4), we compare the different calculation results at $t = 4$ and $t = 10$ by using our three schemes (i.e., BDF1-SAV, BDF2-SAV and CN-SAV), and the classical BDF1 and BDF2 schemes with the same time step sizes $\delta t = 1.0 \times 10^{-3}$ and $\delta t = 1.0 \times 10^{-4}$, respectively. Since the exact solution of this problem is unknown, we use the solution calculated by the BDF2 scheme (4.16) with a very small time step size $\delta t = 1.0 \times 10^{-5}$ as the exact solution at $t = 4$ and $t = 10$.

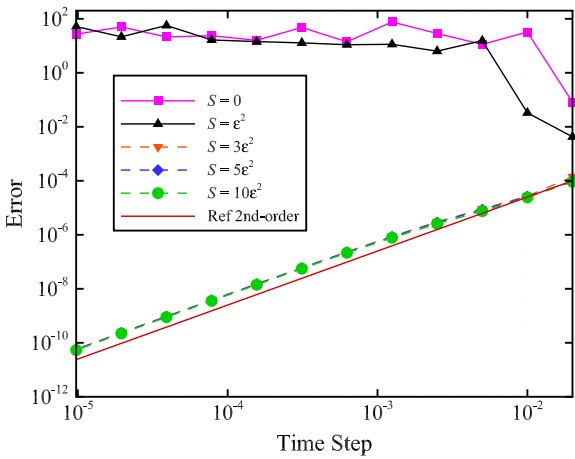
The L^2 errors and the total CPU usage time of all schemes are listed in Tables 2 and 3. We draw the following conclusions from this test.



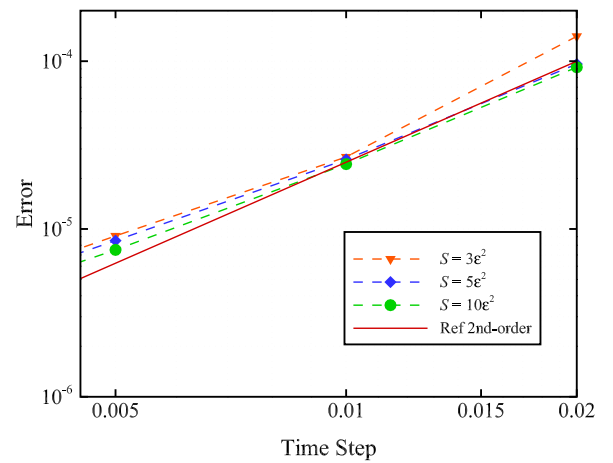
(a) BDF2-SAV



(b) BDF2-SAV



(c) CN-SAV



(d) CN-SAV

Fig. 2. The L^2 errors corresponding to different values of S at $t = 0.1$ computed by using the BDF2-SAV scheme (top) and CN-SAV scheme (bottom) respectively. The right-hand figure in each row is a zoom-in of the figure on the left.

Table 3

The L^2 errors at $t = 4$ and $t = 10$ calculated by schemes BDF1-SAV, BDF2-SAV, CN-SAV, BDF1 and BDF2, and the corresponding total CPU time of each scheme, by using the fixed time step $\delta t = 1 \times 10^{-4}$ with the initial condition (5.3) and parameters (5.4). See Fig. 4. In this case, the BDF1 and BDF2 computations (to $t = 10$) are slightly less expensive than the SAV computations. But, note that numerical saturation seems to have occurred for the BDF2 scheme, and, in fact, the errors seem to be slightly larger, inexplicably, than for the case with larger time step size.

Scheme	$t = 4$		$t = 10$	
	Error	CPU Time	Error	CPU Time
BDF1-SAV	5.07e-4	142.71	4.11e-4	357.53
BDF2-SAV	1.14e-6	143.45	1.06e-6	357.64
CN-SAV	7.68e-7	157.79	7.31e-7	390.28
BDF1	9.07e-6	200.80	7.80e-6	302.89
BDF2	8.36e-7	207.81	1.01e-6	314.37

- 1) The accuracy of BDF1-SAV is the worst, but the CPU time it takes is the least of all SAV schemes and is much less than BDF1 and BDF2 for $\delta t = 1 \times 10^{-3}$ in Table 2.

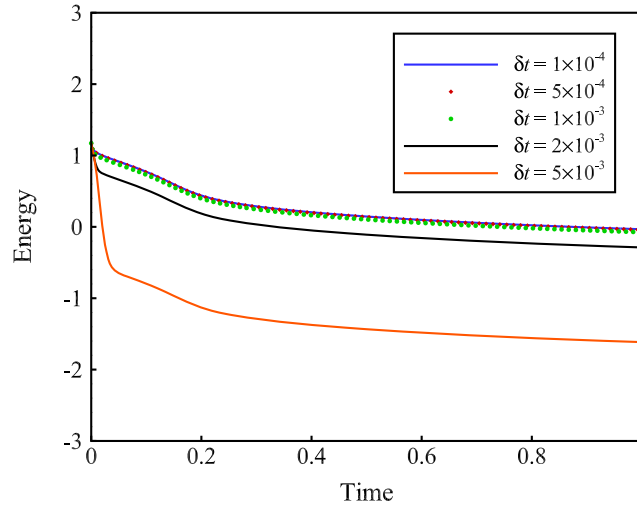


Fig. 3. The energy evolution curves of the modified energy for different time steps of $\delta t = 1 \times 10^{-4}$, 5×10^{-4} , 1×10^{-3} , 2×10^{-3} and 5×10^{-3} by using BDF2-SAV scheme along with the initial condition (5.3) and the parameters (5.4). (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

- 2) The L^2 errors of all the SAV schemes are larger than the pure implicit schemes, BDF1 and BDF2, but the CPU time consumed by them are less for $\delta t = 1 \times 10^{-3}$. See Table 2.
- 3) The L^2 errors of all the SAV schemes reduce when using the time step $\delta t = 1 \times 10^{-4}$. And the CPU time consumed by them are ten times as much as that of $\delta t = 1 \times 10^{-3}$ respectively for the same stopping time by comparing Table 2 with Table 3. The CPU time consumed should increase by a factor of 10 when the time step is reduced by a factor of 10, since the amount of work solving the SAV schemes is always fixed.
- 4) For all the second-order schemes, the error of BDF2 is obviously the best for large time step $\delta t = 1 \times 10^{-3}$ according to Table 2, however, the error of our CN-SAV scheme becomes the smallest for $\delta t = 1 \times 10^{-4}$ according to Table 3.
- 5) What's the most surprising is the accuracy of BDF1 scheme. The errors of BDF1 are of nearly the same order as those of the BDF2-SAV scheme for time step $\delta t = 1 \times 10^{-3}$ and $\delta t = 1 \times 10^{-4}$.
- 6) We can see that in Table 3 the CPU time spent by BDF2-SAV and CN-SAV are less than BDF1 and BDF2 at $t = 4$, however the situation is reversed at $t = 10$. The main reason is that the values of ϕ change slowly when $t > 4$ for this problem such that the solution resulting from BDF1 or BDF2 can be obtained by only one or two PSD iterations with time step $\delta t = 1 \times 10^{-4}$, which means only one six-order equation needs to be solved, while the SAV schemes still need to solve two six-order equations at each time step.

Although the BDF2-SAV scheme is unconditionally modified-energy stable in theory, it will introduce larger local truncation error (LTE) if a larger time step is used in the computation, as shown in Tables 2 and 3. Considering both accuracy and stability, it is necessary to find a reasonable range of the time step size. In Fig. 3, we plot the evolution curves of the modified energy by using different time step sizes $\delta t = 1 \times 10^{-4}$, 5×10^{-4} , 1×10^{-3} , 2×10^{-3} and 5×10^{-3} . We observe that all the energy evolution curves falling even for the largest time step size $\delta t = 5.0 \times 10^{-3}$, which means our schemes are energy stable. Furthermore, the curves for $\delta t = 2 \times 10^{-3}$ and $\delta t = 5 \times 10^{-3}$ are away from other curves, which means the larger time step will introduce larger error, even though the energy is dissipated. As a result, in order to obtain good accuracy, the time step size should not be more than $\delta t = 1 \times 10^{-3}$ for this problem. Here, we choose the time step $\delta t = 1 \times 10^{-3}$ to perform this simulation. In Fig. 4, we present the simulation results at a sequence of time instants. They are highly similar to those in [4]. As can be seen, the amphiphilic material will form the expected network morphology.

Of course, in addition to the fully implicit BDF schemes (utilizing a PSD-type solver) and the linear SAV schemes, other schemes are possible for the FCH problem. In particular, standard implicit-explicit (IMEX) schemes for which only the positive, constant-coefficient linear terms are treated implicitly would be natural to compare against the SAV schemes. Arguably, SAV schemes are of the linear IMEX class. In fact, in a recently submitted arXiv manuscript [9], benchmark computations of morphological complexity in the FCH flow have been conducted that compare in detail BDF2/PSD, SAV, and linear IMEX schemes, in an adaptive time step setting. The authors of [9] show that, like the SAV schemes, the linear IMEX schemes require subtle stabilization/preconditioning to achieve robust performance; such stabilization can enhance efficiency by several orders of magnitude. More importantly, it is discovered that the nonlinear BDF2/PSD scheme achieves the smallest global discretization error at "target" LTE. However, the linear IMEX scheme is the most computationally efficient as measured by the number of FFT calls required to achieve a desired global error. The performance of the SAV scheme mirrors that of linear IMEX, but with slightly higher LTE and roughly half the computational efficiency (due to the auxiliary variables introduced in the SAV approach). See [9] for details.

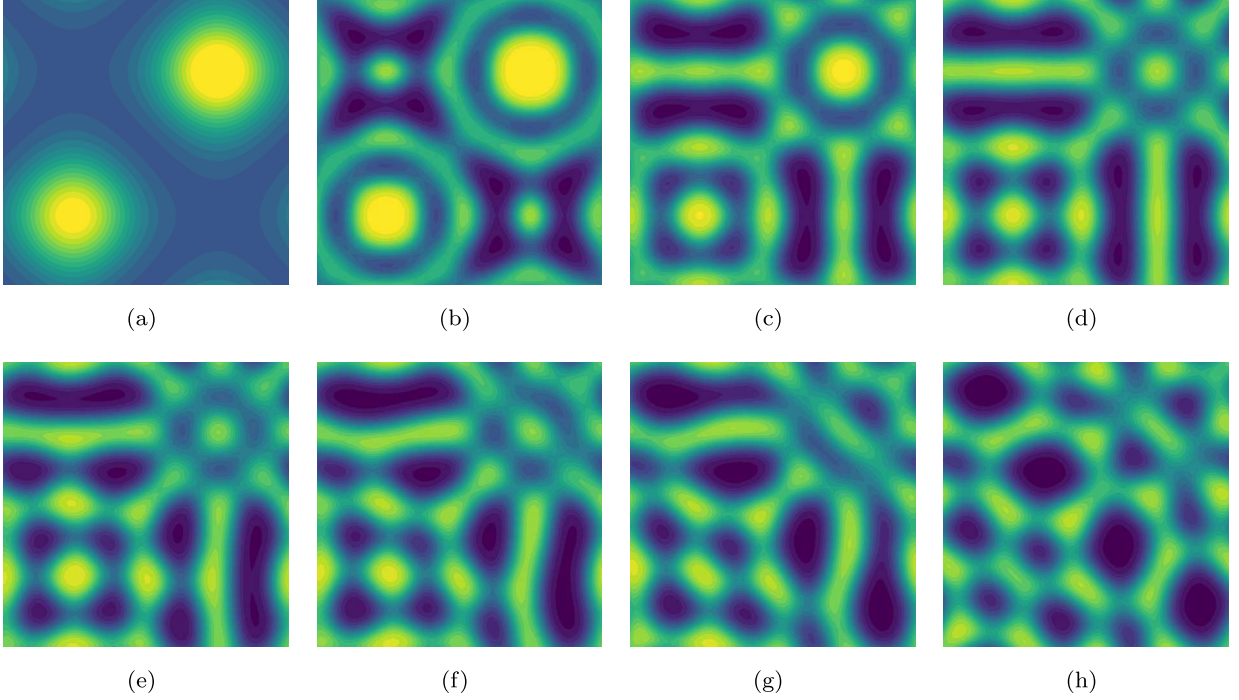


Fig. 4. The snapshots for the simulation of a benchmark problem, which starts with the initial condition (5.3) and parameters (5.4) by using the BDF2-SAV scheme with $\delta t = 1 \times 10^{-3}$, at $t = 0, 1, 5, 20, 400, 500, 600$ and 1000 .

5.3. Meandering instability

For a bilayer with any shape, the interface can be lengthened under the action of the meandering instability [10] when there is too much mass (thickness) thereupon. To verify the validity of our schemes, we simulate the meandering instability phenomenon by setting the initial condition to be a smooth approximation of

$$\varphi(x, y, t=0) = \begin{cases} -1, & x > \sin(y) + L/2 + 0.34, \\ -1, & x < \sin(y) + L/2 - 0.34, \\ 1, & \text{otherwise.} \end{cases} \quad (5.5)$$

We take $\eta = 0.2$ in the FCH equation (3.9)-(3.11) and the other parameters are the same as (5.1).

Typically, Fourier pseudo-spectral methods may have some difficulties in dealing with discontinuous functions, owing to the Gibbs phenomenon. Furthermore, to make fair comparisons for different methods with different spatial mesh sizes, h , we want to smooth this initial data in such a way that it is still periodic. Thus, we seek a filter $\mathcal{F} : L^2(\Omega) \rightarrow C_{\text{per}}^\infty(\Omega)$, which is defined as follows.

1. Suppose that $\varphi \in L^2(\Omega)$ is piecewise continuous, for simplicity. Project φ into a grid of size $\hat{N} \times \hat{N}$ with mesh spacing $\hat{h} = \frac{L}{\hat{N}}$:

$$\mathcal{P}_{\hat{h}}(\varphi)_{i,j} = \varphi(x_i, y_j), \quad 1 \leq i, j \leq \hat{N}.$$

2. Compute the discrete Fourier coefficients, $\hat{\varphi}$, using the $\hat{N} \times \hat{N}$ discrete Fourier transform of the grid function $\mathcal{P}_{\hat{h}}(\varphi)$.
3. Define “filtered” Fourier coefficients via

$$\hat{\varphi}' = \sigma(\vec{k})\hat{\varphi},$$

where $\sigma(\vec{k})$ is the Gaussian filter:

$$\sigma(k_1, k_2) := \exp\left(-\frac{4\lambda|\vec{k}|^2}{\hat{N}^2}\right), \quad (5.6)$$

and

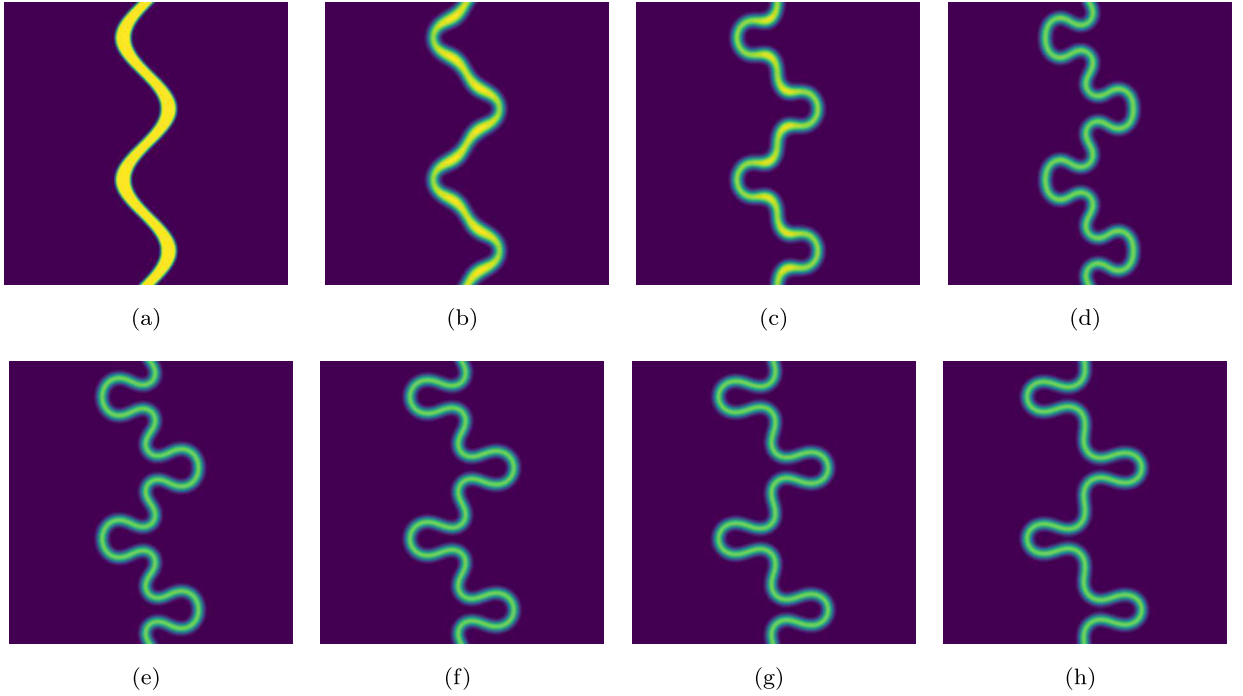


Fig. 5. The snapshots for the simulation of meandering instability at $t = 0, 10, 15, 20, 30, 50, 75$ and 100 . We start this simulation from the filtered version of (5.5), that is $\phi_0 = \mathcal{F}[\varphi]$, with $\hat{N} = 1024$ and $\lambda = 50 \ln(10)$ in (5.6). The other parameters are given by $M = 1$, $\epsilon = 0.1$ and $\eta = 0.2$. This simulation is performed by using the BDF2-SAV scheme with $\delta t = 1.0 \times 10^{-3}$, and 256^2 Fourier modes are adopted to discrete the spatial variables in domain $\Omega = [0, 4\pi] \times [0, 4\pi]$.

$$k_m \in \left\{ -\frac{\hat{N}}{2} + 1, -\frac{\hat{N}}{2} + 2, \dots, \frac{\hat{N}}{2} \right\} = I_{\hat{N}}, \quad m = 1, 2.$$

4. Finally, get the filtered approximation

$$\mathcal{F}[\varphi](x, y) = \sum_{k_1, k_2 \in I_{\hat{N}}} \hat{\varphi}'(k_1, k_2) \exp\left(\frac{2\pi i}{L}(xk_1 + yk_2)\right), \quad (i = \sqrt{-1}),$$

which is both smooth and periodic.

In Fig. 5(a), we plot the filtered initial data $\phi_0 = \mathcal{F}[\varphi]$ with $\hat{N} = 1024$ and $\lambda = 50 \ln(10)$ in (5.6). We apply the BDF2-SAV scheme in the numerical test with time step $\delta t = 1.0 \times 10^{-3}$. The results captured at the sequence of time instants, $t = 0, 10, 15, 20, 30, 50, 75$ and 100 , are plotted in Fig. 5. The meandering instability phenomenon can be observed by using our BDF2-SAV scheme. We also observe that the shape of the curve changes slowly in later stages, only gradually elongated in the horizontal direction.

To measure the accuracy of the new SAV schemes (BDF1-SAV, BDF2-SAV, and CN-SAV) along with large time step size, we take a uniform time step size $\delta t = 1.0 \times 10^{-3}$ and compare with the classical BDF methods. The numerical solution computed by the standard BDF2 scheme (4.16) with time step $\delta t = 1.0 \times 10^{-5}$ is treated as a reference solution at each moment, since an exact solution is not available for this problem. The comparison results of level set $\phi = 0$ between different schemes are plotted in Fig. 6. As can be seen, the curves of BDF2-SAV and CN-SAV fit very well with the reference solution, some parts even overlap perfectly. We have computed approximations of L^∞ and L^2 error norms of each scheme using the reference solution at the exact same time in every time frame of Fig. 6, so that we are able to visualize the difference according to the errors. We observe that the second-order schemes can show high numerical accuracy and the errors of the CN2-SAV scheme are smaller than that of the BDF2-SAV scheme, in both L^∞ and L^2 norms. We also find that the accuracy of the BDF2 scheme is extraordinarily high for $\delta t = 1 \times 10^{-3}$ in this problem, which is consistent with the results shown in Table 2.

Additionally, we list, in Table 4, the CPU time used by the above schemes for simulating the meandering instability. We observe that all the SAV schemes spend less CPU time than the BDF2 schemes if the large time step $\delta t = 1e-3$ is employed. Specifically, for $t \leq 30$, the BDF2 scheme spends three times as much CPU time as all SAV schemes do. In fact, in the early stage of this evolutionary process, i.e., $t \leq 30$, many drastic topological changes are involved in the process, so that the PSD solver requires more iterations to converge, i.e., using more CPU time; while in the SAV schemes only two

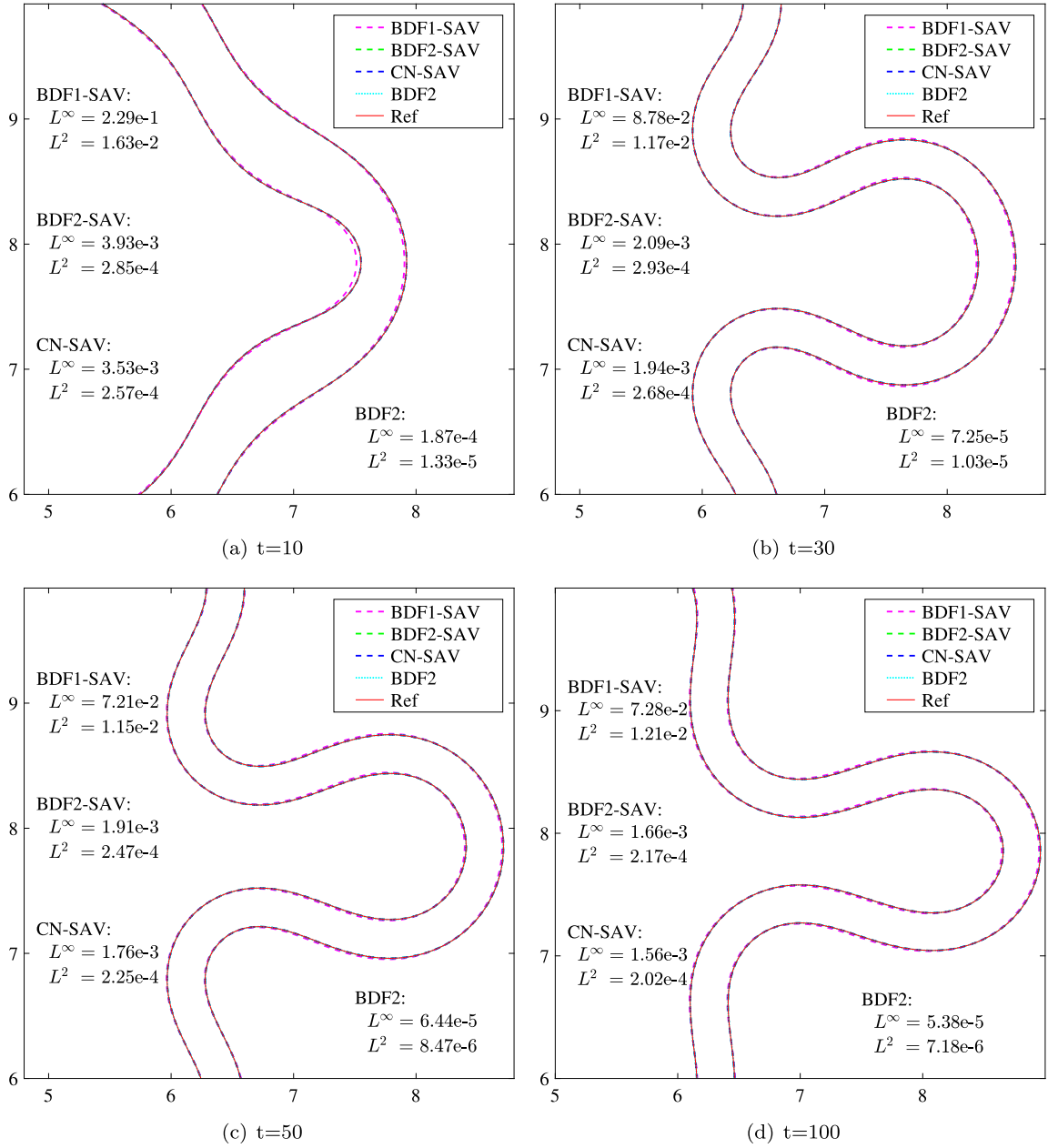


Fig. 6. The detailed comparison results for the BDF1-SAV, BDF2-SAV, CN-SAV and BDF2, at a sequence of time instants $t = 10, 30, 50$ and 100 , with the same time step $\delta t = 1.0 \times 10^{-3}$. The “Ref” represents the reference solution computed by the BDF2 scheme (4.16) with time step size $\delta t = 1.0 \times 10^{-5}$. The L^∞ and L^2 errors are plotted in these pictures.

sixth-order equations need to be solved for finding the numerical solution at each time step. By the end of the simulation, the evolution is slower so that the convergence speed of the PSD solver is accelerated. This makes the BDF2/PSD scheme more competitive in solve-time. In terms of error, the BDF2 method is the clear winner, at approximately one to two orders of magnitude more accurate.

5.4. Phase separation in 2D and 3D

One of the main purposes of the FCH equation is to investigate the process of the phase separation of an amphiphilic binary mixture. The FCH free energy considered so far is the most basic one, while it can simulate the formation of interesting structures (for example, the bi-layer structure). Here we use it to simulate the bi-layer network structure and set the initial condition as follows:

Table 4

The CPU time used by schemes BDF1-SAV, BDF2-SAV, CN-SAV, and BDF2 for different times in the simulation of meandering instability, see Fig. 6. The fixed time step $\delta t = 1 \times 10^{-3}$ is used for all the simulations.

Scheme	$t = 10$	$t = 30$	$t = 50$	$t = 100$
BDF1-SAV	124.36	368.85	612.73	1215.46
BDF2-SAV	125.65	376.58	624.91	1244.72
CN-SAV	133.73	399.31	666.24	1355.55
BDF2	488.39	1330.09	1611.76	2139.08

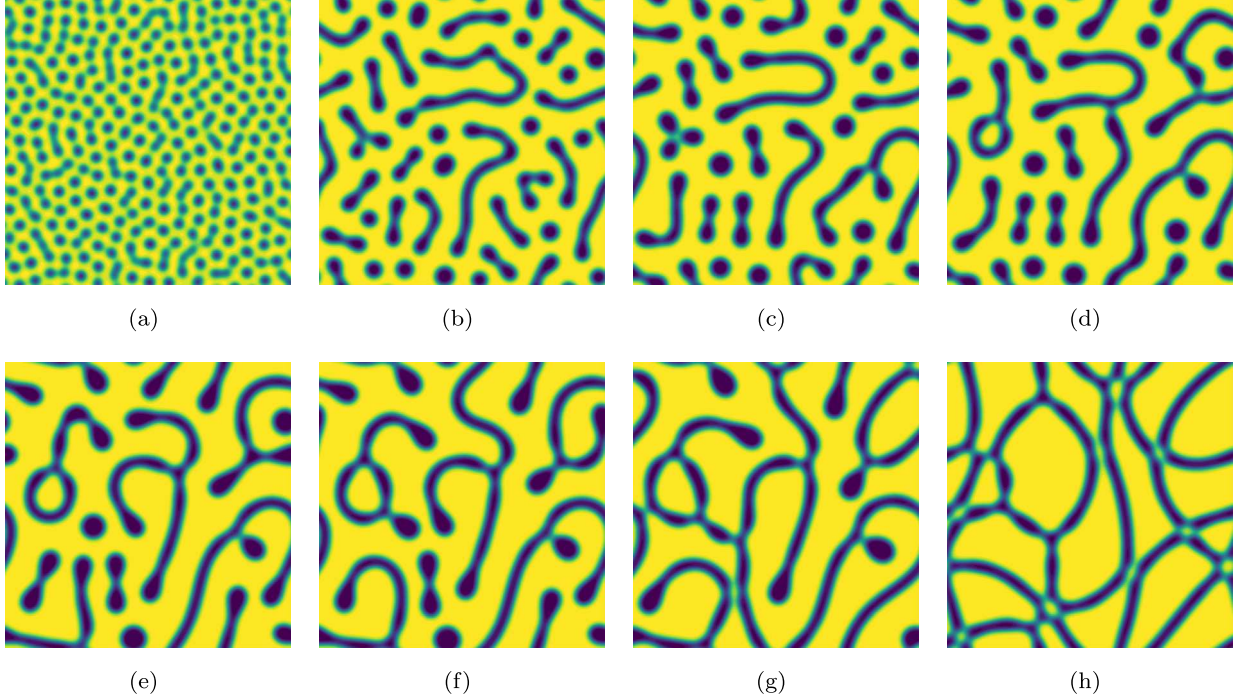


Fig. 7. The snapshots of the 2D phase separation computed by the BDF2-SAV scheme with initial condition (5.7) at a sequence of time instants, $t = 0.2, 10, 50, 100, 300, 500, 1000$ and 5000 . The parameters are given by $M = 1$, $\epsilon = 0.1$, $\eta = \epsilon^2$, $L = 4\pi$, and $N^2 = 256^2$.

$$\phi(x, y, t = 0) = 0.5 + 0.001 \times \text{rand}(x, y), \quad (5.7)$$

where the function $\text{rand}(x, y)$ generates random numbers uniformly distributed in $[-1, 1]$. The parameters involved are listed in (5.1) and the BDF2-SAV scheme with the time step $\delta t = 1.0 \times 10^{-3}$ is adopted to perform this simulation.

The 2D simulation results of the phase variable $\phi(x, y, t)$ at a sequence of time instants, $t = 0.2, 10, 50, 100, 300, 500, 1000$, and 5000 , are displayed in Fig. 7. The blue domains are the amphiphilic polymers corresponding to $\phi = -1$, while the yellow domains are the solvent phase corresponding to $\phi = +1$. Many pore structures rapidly formed at $t = 0.2$ and some of them then merged into worm-like structures over time. At $t = 10$, the narrow worm-like bi-layers merge to form enclosed regions. At $t = 10, 50$ and 100 , some interesting structures are captured, which are also observed by using the convex splitting method in [21], like Y-junctions, antennae, and so on. Finally, amphiphilic polymers form the bi-layer structures and the interfaces elongate and merge with each other to form a network structure after a long period evolution. The final near-equilibrium solution at $t = 5000$ shows us the network structure. See also results observed in [21,25,31].

The evolution curves of the modified energy and the original primal energy are displayed in Fig. 8. It is observed that both two energies decrease with respect to time for $\delta t = 1.0 \times 10^{-3}$, which demonstrate that the BDF2-SAV scheme, when stabilized, is energy stable even with $\delta t = 1.0 \times 10^{-3}$.

For 3D phase separation, we use the BDF2-SAV scheme with the same parameters (5.1) as that used in 2D, except now $N = 2^7$. The time step size $\delta t = 1.0 \times 10^{-3}$ is used to perform this simulation. The initial condition is given by

$$\phi(x, y, z, t = 0) = 0.5 + 0.001 \times \text{rand}(x, y, z), \quad (5.8)$$

where the function $\text{rand}(x, y, z)$ generates random numbers uniformly distributed in $[-1, 1]$. We present the simulation results at a sequence of time instants, $t = 20, 500$ and 1000 , in Fig. 9. The pictures of the first column in Fig. 9 are the iso-surface diagrams, in which the green and purple parts represent the level sets $\phi = 0.0$ and $\phi = 0.2$, respectively. The

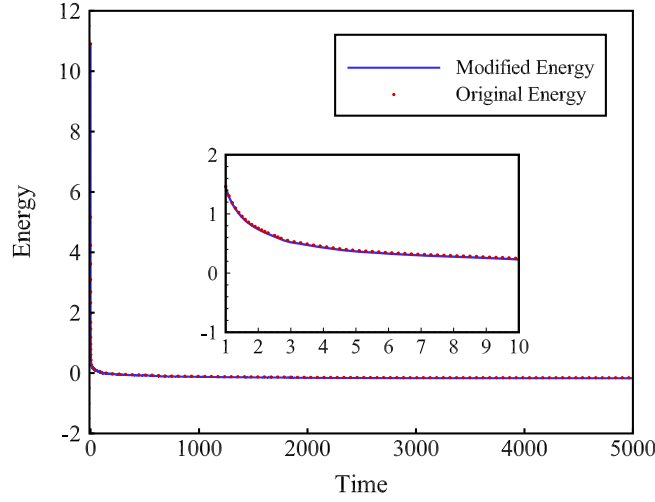


Fig. 8. The energy evolution curves of the modified energy and the original energy for the 2D simulation of phase separation, with $\delta t = 1.0 \times 10^{-3}$.

pictures of the second column in Fig. 9 correspond to slice diagrams. The pore and the bilayer structures could be clearly observed. We find that the phase separation behavior is similar to the 2D case. The basic structures are formed in the early stage of the phase separation process.

5.5. Pearling instability in annuli

Here, in order to demonstrate that our schemes can be used for other more complicated FCH models, we consider a more general form of the FCH energy

$$\tilde{E}[\phi] = \int_{\Omega} \frac{1}{2} \left(\epsilon^2 \Delta \phi - \tilde{F}'(\phi) \right)^2 - \left(\eta_1 \frac{\epsilon^2}{2} |\nabla \phi|^2 + \eta_2 \tilde{F}(\phi) \right) d\mathbf{x}, \quad (5.9)$$

in which, ϵ is a parameter that decides the thickness of the interface, and $\eta_1 > 0$ and $\eta_2 \in \mathbb{R}$ are also two small parameters related to the properties of amphiphilic materials. We obtain something similar to the original energy expression (2.4) when $\eta_1 = \eta_2 = \eta$.

We get the more general FCH equation, from the general FCH energy (5.9), in the H^{-1} gradient flow:

$$\dot{\phi} = M \Delta \mu, \quad (5.10)$$

$$\mu = (\epsilon^2 \Delta - \tilde{F}''(\phi) + \eta_1) \omega + (\eta_1 - \eta_2) \tilde{F}'(\phi), \quad (5.11)$$

$$\omega = \epsilon^2 \Delta \phi - \tilde{F}'(\phi). \quad (5.12)$$

Periodic boundary conditions are enforced for this equation, as before. In order to capture the dynamics associated to pearling bifurcation, we consider a double well potential function $\tilde{F}(\phi)$ with unequal well depths, $\tilde{F}(\phi = 1) < \tilde{F}(\phi = -1) = 0$. Specifically, we assume

$$\tilde{F}(\phi) = \frac{1}{2}(\phi + 1)^2 \left(\frac{1}{2}(\phi - 1)^2 + \frac{2}{3}\tau(\phi - 2) \right). \quad (5.13)$$

This implies that

$$\tilde{F}'(\phi) = (\phi + 1)(\phi + \tau)(\phi - 1), \quad (5.14)$$

and

$$\tilde{F}''(\phi) = (\phi + 1)(\phi - 1) + 2\phi(\phi + \tau). \quad (5.15)$$

We cannot directly apply our SAV schemes to deal with this special FCH equation (5.10)-(5.11) because there is a quadratic term in (5.14) that did not exist previously. However, we can use a transformation of ϕ to make the quadratic term in the derivative disappear. The easiest method is to implement the following affine change of variables:

$$\psi := \phi + \frac{\tau}{3}, \quad F(\psi) := \tilde{F}\left(\psi - \frac{\tau}{3}\right) = \tilde{F}(\phi).$$

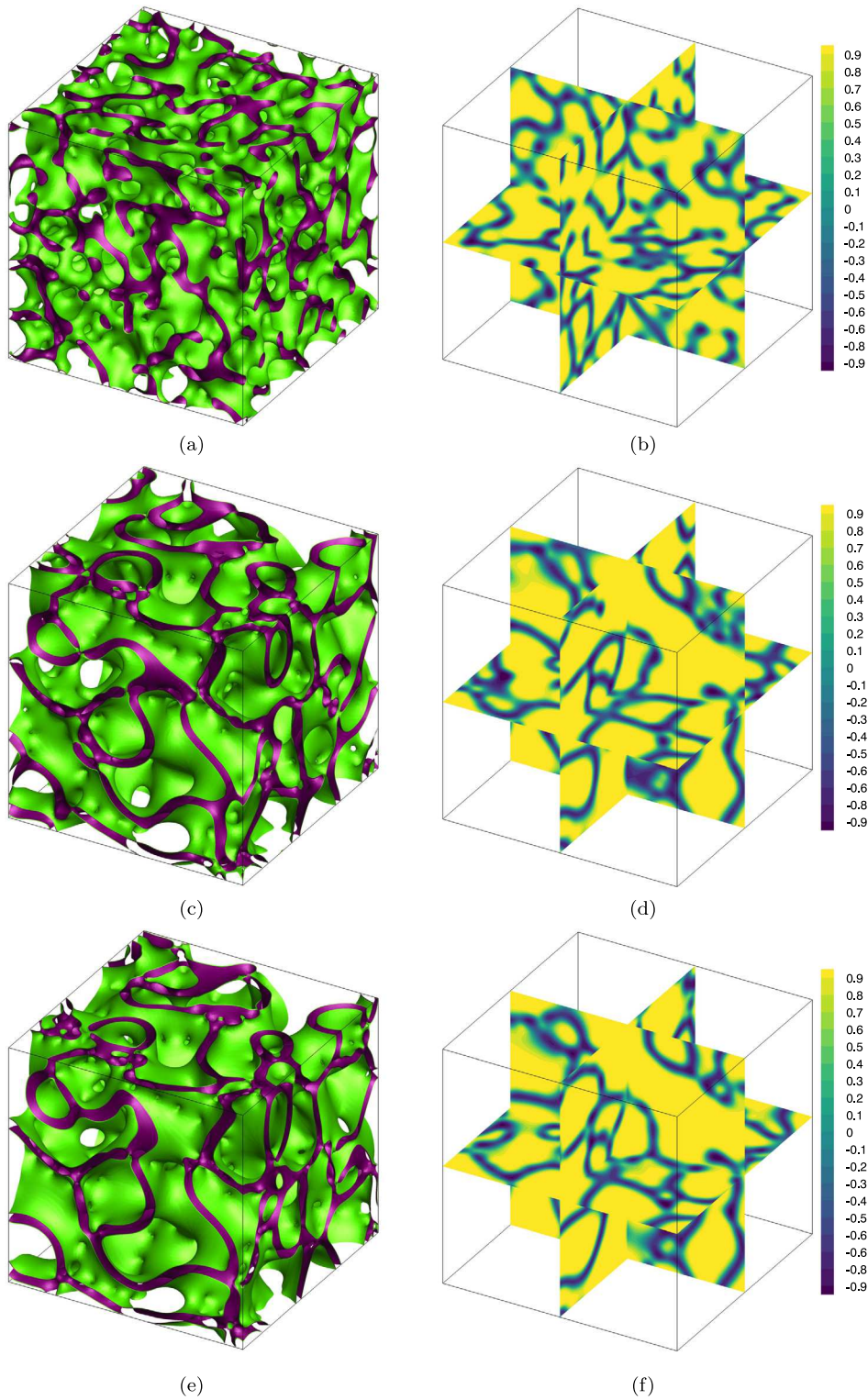


Fig. 9. The iso-surface diagrams (left) and the slice diagrams (right) of the 3D phase separation computed by the BDF2-SAV scheme with initial condition (5.7) at a sequence of time instants, $t = 20, 500$ and 1000 , respectively. This simulation is performed by using 128^3 Fourier modes, $\epsilon = 0.1$, $\eta = \epsilon^2$ and $M = 1$ in domain $\Omega = [0, 4\pi]^3$.

Then we have

$$F'(\psi) = \psi^3 - \left(1 + \frac{\tau^2}{3}\right)\psi - \frac{2\tau}{3}\left(1 + \frac{\tau}{3}\right)\left(1 - \frac{\tau}{3}\right). \quad (5.16)$$

We have successfully eliminated the quadratic term, and the SAV schemes designed before in this paper can be used for this more general FCH equation (5.10)–(5.11). Here, we briefly introduce the approach of applying our SAV schemes for solving the general FCH equation (5.10)–(5.11).

In terms of ψ , the FCH free energy (5.9) can be written as

$$\begin{aligned} E[\psi] = & \int_{\Omega} \left\{ \frac{1}{2}(\epsilon^2 \Delta \psi)^2 + 3\epsilon^2 \psi^2 |\nabla \psi|^2 - \left(\frac{\tau^2}{3} + 1\right) \epsilon^2 |\nabla \psi|^2 \right. \\ & \left. + \frac{1}{2}(F'(\psi))^2 - \left(\eta_1 \frac{\epsilon^2}{2} |\nabla \psi|^2 + \eta_2 F(\psi)\right) \right\} d\mathbf{x}. \end{aligned} \quad (5.17)$$

Then, setting

$$G(\psi) = \int_{\Omega} \left\{ 3\epsilon^2 \psi^2 |\nabla \psi|^2 + \frac{1}{2}(F'(\psi))^2 - \eta_2 F(\psi) \right\} d\mathbf{x}, \quad (5.18)$$

we have

$$U = \sqrt{G(\psi) + B}, \quad (5.19)$$

and the modified FCH free energy becomes

$$E[\psi, U] = \int_{\Omega} \left\{ \frac{1}{2}(\epsilon^2 \Delta \psi)^2 - \left(\frac{\tau^2}{3} + \frac{\eta_1}{2} + 1\right) \epsilon^2 |\nabla \psi|^2 \right\} d\mathbf{x} + U^2 - B. \quad (5.20)$$

Finally, we can rewrite the original general FCH equation (5.10)–(5.11) as follows

$$\dot{\psi} = M \Delta \mu, \quad (5.21)$$

$$\mu = \epsilon^4 \Delta \psi + (2 + \eta_1 + \frac{2\tau^2}{3}) \epsilon^2 \Delta \psi + U X(\psi), \quad (5.22)$$

$$\dot{U} = \frac{1}{2} \int_{\Omega} X(\psi) \dot{\psi} d\mathbf{x}, \quad (5.23)$$

where

$$X(\psi) = \frac{6\epsilon^2(\psi |\nabla \psi|^2 - \nabla \cdot (\psi^2 \nabla \psi)) + F'(\psi)F''(\psi) - \eta_2 F'(\psi)}{\sqrt{\int_{\Omega} \left\{ 3\epsilon^2 \psi^2 |\nabla \psi|^2 + \frac{1}{2}(F'(\psi))^2 - \eta_2 F(\psi) \right\} d\mathbf{x} + B}}. \quad (5.24)$$

Obviously, the system (5.21)–(5.23) and the system (3.9)–(3.11) are formally identical (at the PDE level), so that our SAV schemes can be directly applied.

5.5.1. An elliptical annulus

For the first example of the pearling bifurcation, we use a special initial value, a smoothed elliptical ring, which is obtained by applying the strategy in Section 5.3 to

$$\varphi(x, y, t=0) = \begin{cases} -1, & f(x, y) > L/4 + 0.2, \\ -1, & f(x, y) < L/4 - 0.2, \\ 1, & \text{otherwise,} \end{cases} \quad (5.25)$$

where $f(x, y) = \sqrt{(x - L/2)^2 + 0.5(y - L/2)^2}$. In order to get an initial value with good smoothness, we filter φ by choosing $\hat{N} = 1024$ and $\lambda = 50 \ln(10)$ in equation (5.6). Then, we first adopt the BDF2-SAV scheme to perform this simulation, where the other necessary parameters are given by

$$M = 1, \epsilon = 0.1, \eta_1 = 1.45\epsilon, \eta_2 = 2\epsilon, \tau = 0.125, L = 4\pi, N = 2^8, B = L^5. \quad (5.26)$$

The numerical simulation results from the BDF2-SAV are displayed in Fig. 10. We observe that the pearling bifurcation occurs by time $t = 2$ and the first place to change is located in the middle along the short axis of the elliptical ring. This is

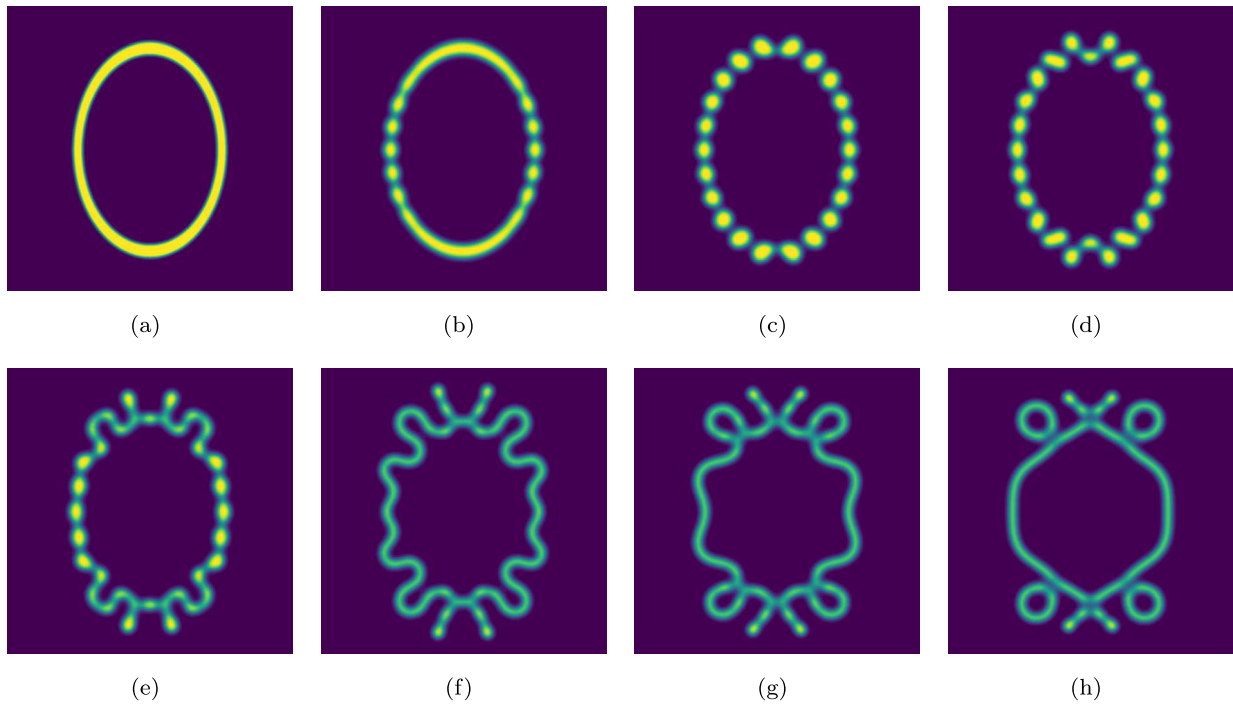


Fig. 10. The 2D dynamical evolution of phase variable ϕ simulated by the BDF2-SAV scheme along with $\delta t = 1 \times 10^{-3}$, which starts from the filtered version of (5.25) as initial data. The snapshots are the simulation results of pearling bifurcation at a sequence of time instants, $t = 0, 2, 10, 15, 20, 30, 60$ and 100 . The parameters are $M = 1$, $\epsilon = 0.1$, $\eta_1 = 1.45\epsilon$, $\eta_2 = 2\epsilon$ and $\tau = 0.125$.

Table 5

The total CPU time consumption by the BDF2-SAV and the BDF2 schemes, respectively, for $\delta t = 1 \times 10^{-3}$ and $\delta t = 1 \times 10^{-4}$ at $t = 20, t = 60$ and $t = 100$ with the initial value given as the smooth function $\phi_0 = \mathcal{F}[\varphi]$, where φ is given in (5.25).

δt	Scheme	CPU Time		
		t=20	t=60	t=100
1.0×10^{-3}	BDF2-SAV	251.57	754.05	1254.61
	BDF2	1094.41	2267.34	3061.94
1.0×10^{-4}	BDF2-SAV	2489.41	6661.05	10627.41
	BDF2	3952.33	7512.97	10248.57

likely because the bilayer is thinner in these regions. These little pearls reconnect into a line by $t = 20$ and, subsequently, a meandering instability occurs.

Just as before, we use the pure BDF2 numerical solution at a high temporal resolution $\delta t = 1.0 \times 10^{-5}$ as the reference solution for the comparison in terms of accuracy and computational speed. We present the comparison results by using two level sets $\phi = \pm 0.2$ for $\delta t = 1.0 \times 10^{-3}$ in Fig. 11 and $\delta t = 1.0 \times 10^{-4}$ in Fig. 12 at $t = 10, t = 20, t = 60$ and $t = 100$, respectively. The L^∞ error and L^2 error of each scheme (in comparison with the reference solution) are also displayed in each picture. We observe that for the larger time step $\delta t = 1.0 \times 10^{-3}$, the accuracy of the standard BDF2 is significantly higher than that of BDF2-SAV at any time. The accuracy of BDF2-SAV is obviously improved when the smaller time step $\delta t = 1.0 \times 10^{-4}$ is taken, and its error is in the same order as that of the standard BDF2. Interestingly, the L^2 (or L^∞) error of BDF2-SAV scheme is slightly smaller at $t = 60$ than that of BDF2 scheme. This reverses at $t = 100$.

In Table 5, we list the CPU time consumption by each numerical scheme with different time step sizes. We find that the BDF2-SAV takes much less CPU time than the BDF2 for $\delta t = 1.0 \times 10^{-3}$ at any time. However, for the smaller time step size $\delta t = 1.0 \times 10^{-4}$, the CPU time consumption by the BDF2-SAV is less than that of BDF2 at $t = 20$. As the convergence speed of the PSD solver is accelerated, the total CPU time consumption of BDF2 is less than that of BDF2-SAV.

Overall, the computational speed of BDF2-SAV is faster, while the accuracy is lower than that of BDF2 for the time step size $\delta t = 1.0 \times 10^{-3}$. Meanwhile, the BDF2 is more efficient and consumes less CPU time than BDF2-SAV, and their accuracy is roughly the same with the time step size $\delta t = 1.0 \times 10^{-4}$.

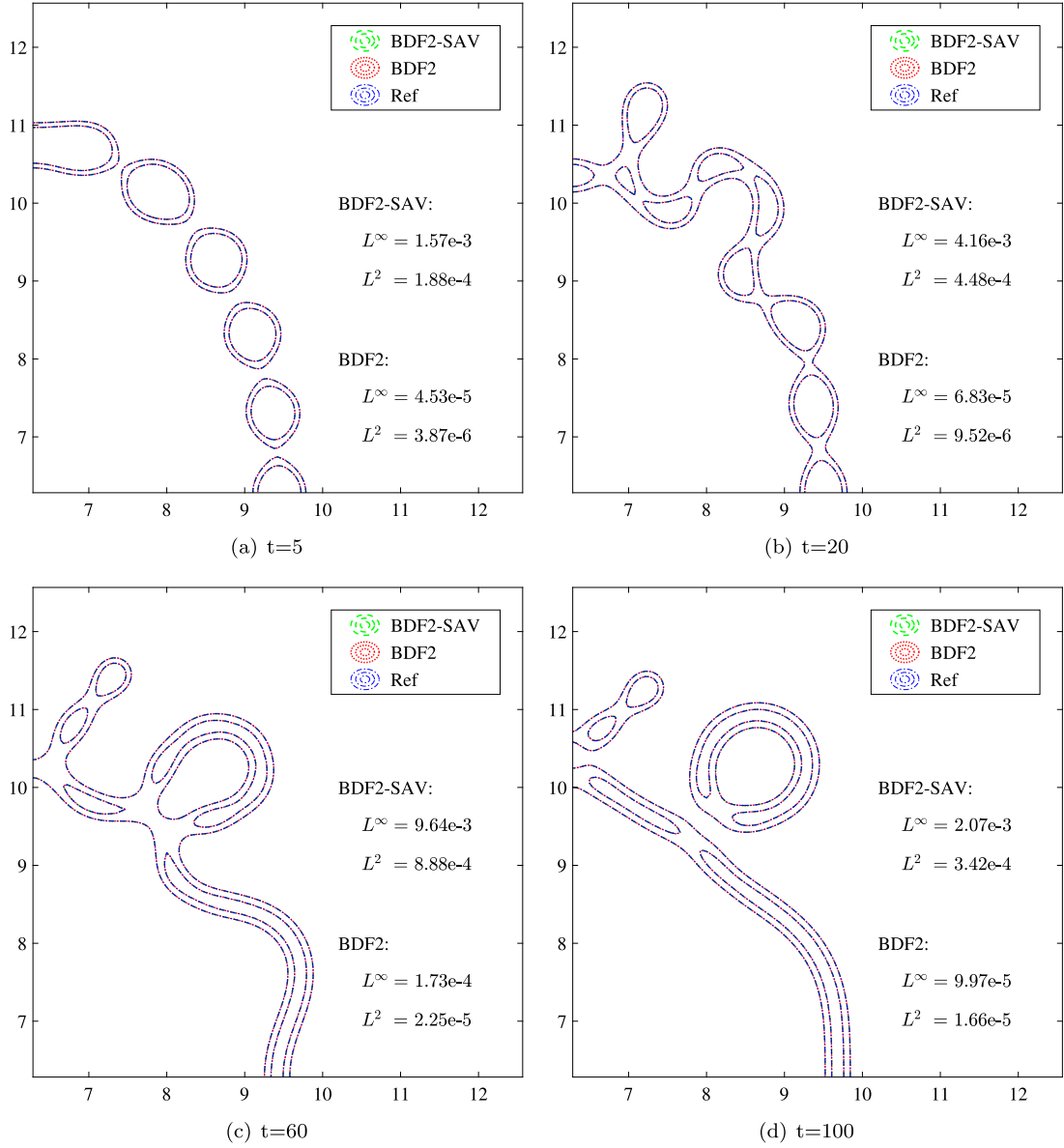


Fig. 11. The detailed comparison results for the BDF2-SAV and BDF2 schemes at a sequence of time instants, $t = 10, 20, 60,$ and 100 , with the same time step size $\delta t = 1.0 \times 10^{-3}$. The "Ref" represents the reference solution computed by the BDF2 scheme with time step $\delta t = 1.0 \times 10^{-5}$. The L^∞ and L^2 errors are presented in these pictures. The BDF2 solution has an error that is about 20 to 30 times smaller than that of the BDF2-SAV scheme.

5.5.2. Circular rings of varying thicknesses

Next we use a circular ring as the initial value:

$$\phi(x, y, t = 0) = 2 \cosh^{-1} \left(\frac{\sqrt{(x - 2\pi)^2 + (y - 2\pi)^2} - \pi}{\epsilon \cdot d} \right) - 1, \quad (5.27)$$

where $d > 0$ is a constant parameter which can control the interface width at the initial time. This initial condition is not smoothly periodic, but it is approximately so, provided d is not too large. We apply both the BDF2-SAV and fully implicit BDF2 schemes to solve the general FCH equation (5.21)–(5.23) by taking time step size $\delta t = 1.0 \times 10^{-3}$. The other parameters are the same as (5.26) except for $\eta_1 = 2\epsilon$. For the parameter d , we choose a couple of different values, $d = 1.3, 1.4, 1.5,$ and 2.0 , to observe the effect of the thickness of the initial interface on the final numerical results. The terminal time is set to be $t = 100$ for all the following simulations and the numerical results produced by the BDF2-SAV scheme are presented in Figs. 14, 16, 18, and 20.

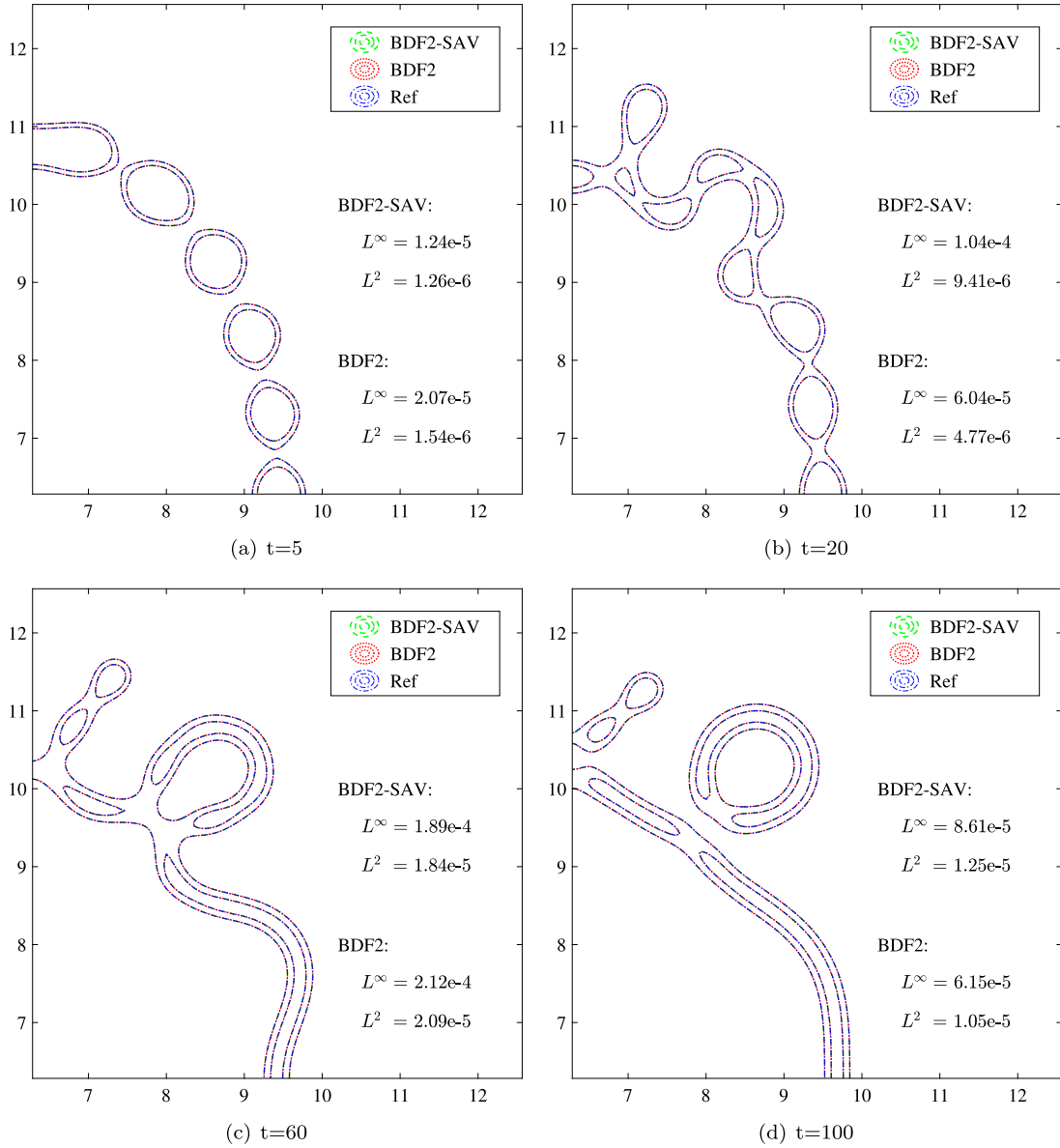


Fig. 12. The detailed comparison results of the BDF2-SAV and BDF2 schemes at time $t = 10, 20, 60$, and 100 , with the same time step $\delta t = 1.0 \times 10^{-4}$. The “Ref” represents the reference solution computed by the standard BDF2 scheme with time step $\delta t = 1.0 \times 10^{-5}$. The L^∞ and L^2 errors are presented in these pictures.

By comparing these figures, we find that different shapes form with different initial interface width, and the pearling bifurcation appears earlier when the thickness of the initial interface is smaller. Moreover, in each case, these small pearls on the ring appear almost at the same time. In the final time, all the shapes in Figs. 14, 16, 18, and 20 are different. For the BDF2-SAV scheme, we also present the energy evolution curves corresponding to different values of d in Fig. 13, and we find that every curve is monotonically decreasing.

We also take the numerical solution computed by the standard BDF2 scheme of the general FCH equation with the time step size $\delta t = 1.0 \times 10^{-3}$, to investigate the numerical accuracy of the BDF2-SAV scheme for the long time evolution. The simulation results produced by the standard BDF2 scheme are presented in Figs. 15, 17, 19 and 21. We observe, in the “eyeball norm”, that the results computed by these two numerical schemes are almost exactly the same.

For the above simulations, the BDF2-SAV and BDF2 schemes produce qualitatively identical numerical results for each of the initial conditions. In order to make a quantitative comparison between them, the simulation with $d = 1.4$ (see Figs. 16 and 17) is selected to compute error estimates. See Fig. 22. For CPU times, see Table 6. The reference solution is obtained by using the BDF2 scheme with a very small time step, $\delta t = 1.0 \times 10^{-5}$, to solve the FCH equation (5.10)–(5.12). Here, we

Table 6

The CPU time comparisons at $t = 10, 50$, and 100 between scheme BDF2-SAV and scheme BDF2 by using the same large time step $\delta t = 1.0 \times 10^{-3}$ and the same initial condition (5.27) where $d = 1.4$.

Scheme	$t = 10$	$t = 50$	$t = 100$
BDF2-SAV	129.60	643.73	1280.96
BDF2	317.73	1661.26	2688.28

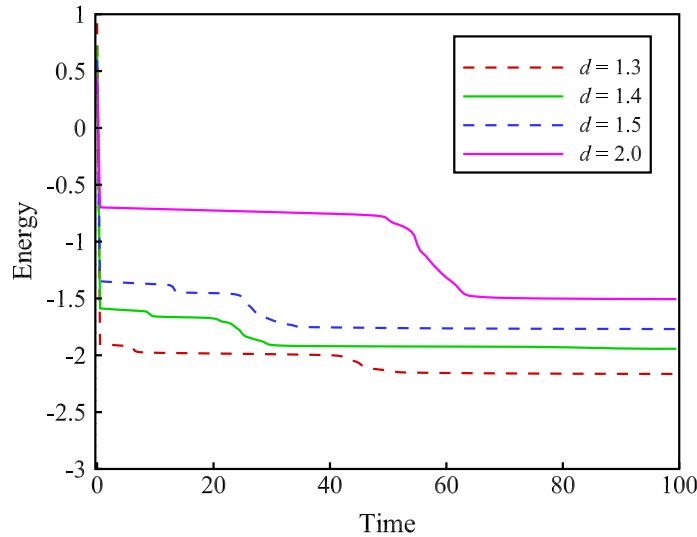


Fig. 13. The energy evolution curves of the modified energy for different values of d until $t = 100$ in equation (5.27) by applying the BDF2-SAV scheme.

find the same results as before: the BDF2-SAV scheme can effectively reduce the simulation time by a factor of two to three at the price of losing one to two digits of precision.

6. Conclusions

We designed three unconditionally modified-energy stable linear numerical schemes, BDF1-SAV, BDF2-SAV, and CN-SAV, by using the SAV method for approximating solutions of a symmetric FCH equation. To avoid spurious oscillations of the physical energy at modest time step sizes, we have added linear stabilization terms. We proved that each scheme admits a corresponding discrete modified-energy dissipation law. To study pearling bifurcations, we applied our schemes to solve the more general FCH equation (5.10)–(5.11) in which the double-well potential function is asymmetric.

Using a toy test problem, we verified the expected convergence rates of each SAV scheme. In addition, with this example, we numerically explained the reason for adding the extra stabilization term, and we determined the specific value of the stabilization coefficient S for a given set of parameters. We found that the introduction of the stabilization term compensates for the fact that we can use large time steps in these schemes, especially for the two second-order schemes, BDF2-SAV and CN-SAV. We carefully choose the stabilization parameters to balance stability and accuracy.

To ascertain the relative strengths and weaknesses of the stabilized SAV schemes, we have compared output with the classical BDF1 and BDF2 schemes. To make these fully implicit, nonlinear BDF schemes competitive, we used an efficient PSD solver. We gauged the accuracy and computational efficiency (CPU usage time) of the methods based on some benchmark problems, using both symmetric and asymmetric double-well potential functions. To make the discussion and comparisons as simple as possible, we used a fixed time step size. However, since the FCH equation, and many others like it, have solutions that exhibit a wide variety of time scales, time adaptive schemes should be considered in the future.

For the symmetric double-well form of the FCH equation, we used an infinitely smooth function as the initial value to compare the schemes, BDF1-SAV, BDF2-SAV, CN-SAV, BDF1, BDF2. We found that the SAV schemes can generally execute faster than the fully implicit schemes for modest time step sizes. But, as expected, the classical BDF2 scheme is more accurate than all the others, for $\delta t = 1 \times 10^{-3}$. For $\delta t = 1 \times 10^{-4}$, the SAV schemes still run slightly faster than the fully implicit schemes at early times, while the standard BDF schemes are the winners at the higher temporal resolution in terms of overall efficiency. At the smaller time step size, the CN-SAV was the winner with smallest error, slightly edging out the BDF2 method. The simulation results of this benchmark problem are qualitatively consistent with the results in [4].

For some physics problems, we must use large time steps to access later times. Such is the case for the meandering instability, where a bilayer can lengthen over a very large time scale. For this problem, we compared the schemes, BDF1-

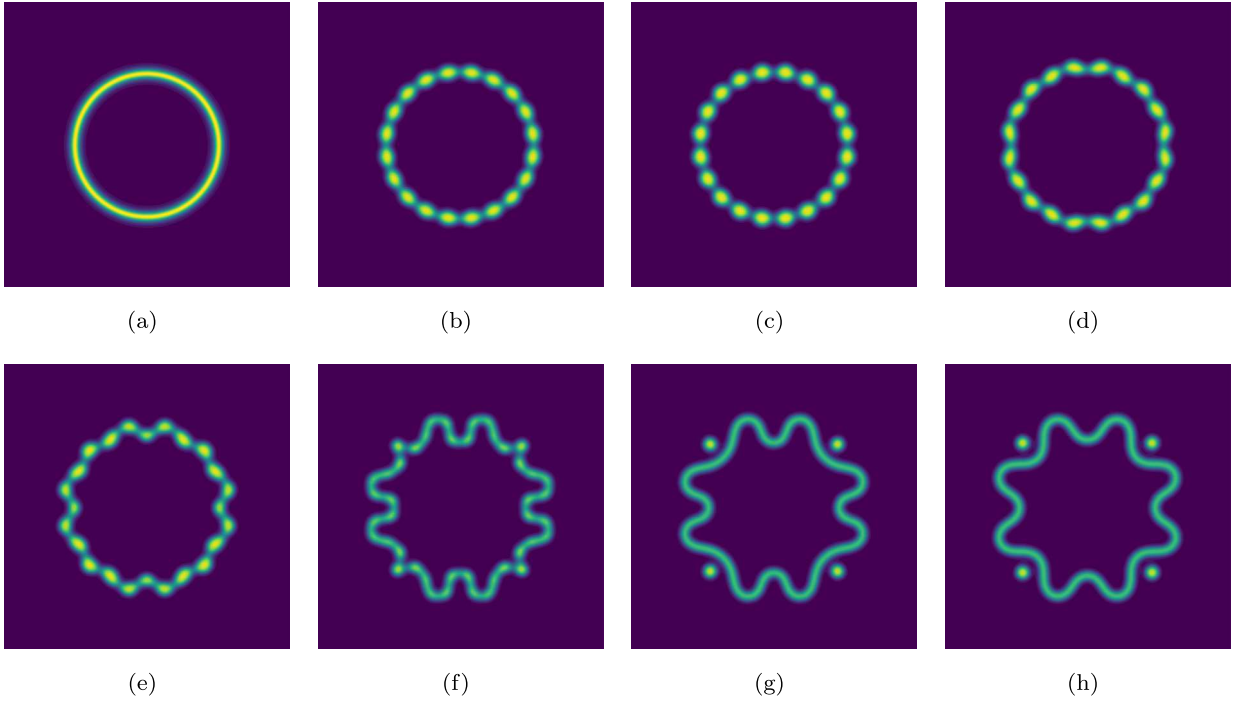


Fig. 14. The 2D dynamical evolution of phase variable ϕ simulated by the BDF2-SAV scheme along with $\delta t = 1 \times 10^{-3}$, which starts from initial condition (5.27) with $d = 1.3$. The snapshots are the simulation results of the pearling bifurcation at a sequence of time instants, $t = 0, 6.5, 7.5, 40, 43, 50, 70$ and 100 , respectively, along with $M = 1$, $\epsilon = 0.1$, $\eta_1 = 2\epsilon$, $\eta_2 = 2\epsilon$ and $\tau = 0.125$.

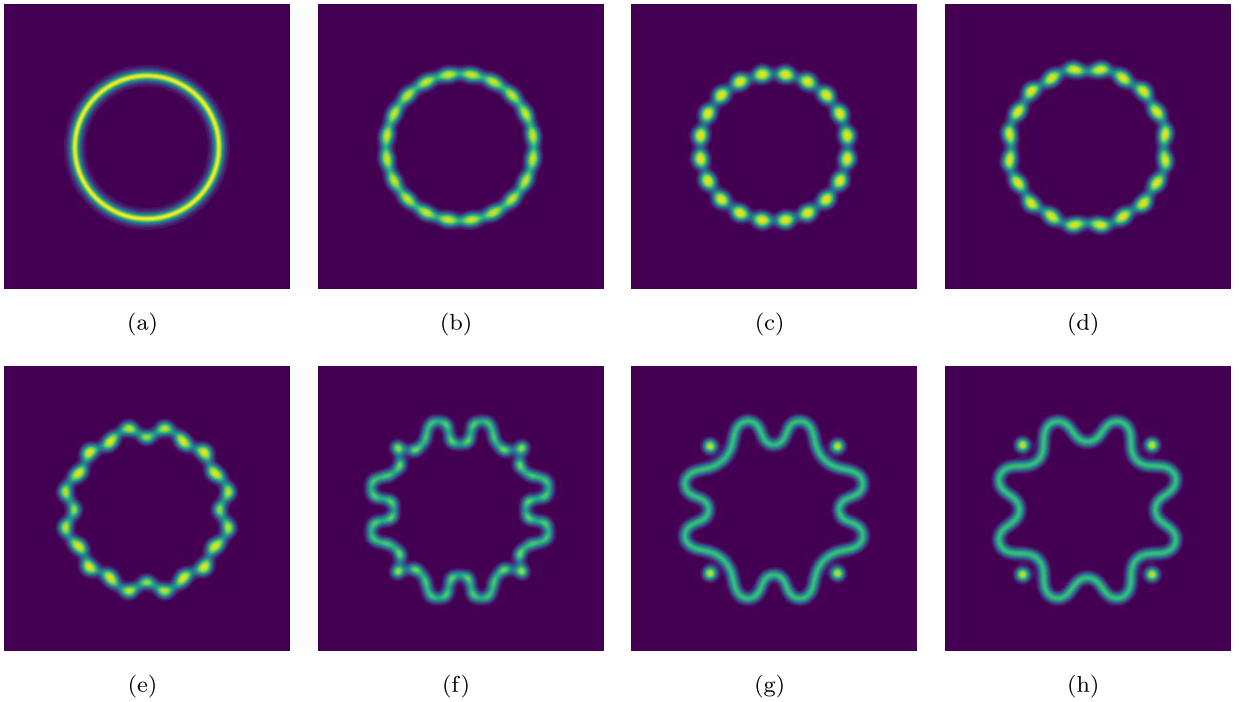


Fig. 15. The 2D dynamical evolution of phase variable ϕ simulated by the standard BDF2 scheme along with $\delta t = 1 \times 10^{-3}$, which starts from initial condition (5.27) with $d = 1.3$. The snapshots are the simulation results of pearling bifurcation a sequence of time instants, $t = 0, 6.5, 7.5, 40, 43, 50, 70$ and 100 , respectively, along with $M = 1$, $\epsilon = 0.1$, $\eta_1 = 2\epsilon$, $\eta_2 = 2\epsilon$ and $\tau = 0.125$.

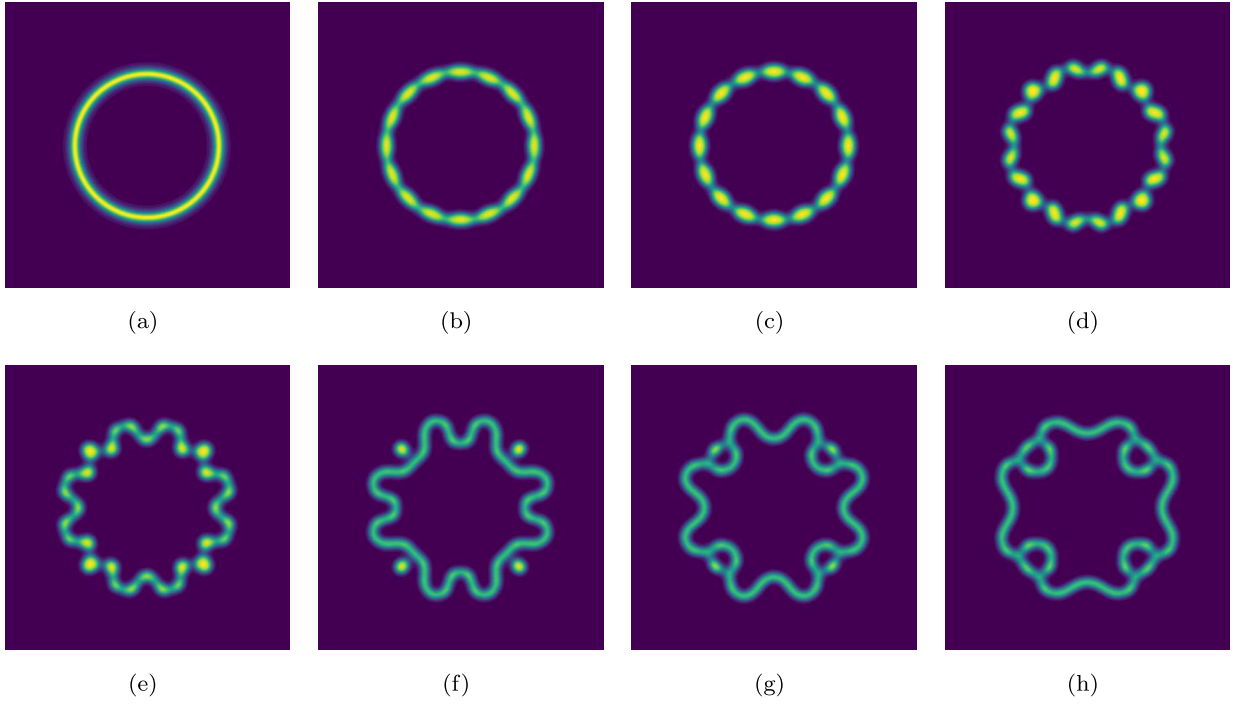


Fig. 16. The 2D dynamical evolution of phase variable ϕ simulated by the BDF2-SAV scheme along with $\delta t = 1 \times 10^{-3}$, which starts from initial condition (5.27) with $d = 1.4$. The snapshots are the simulation results of pearling bifurcation at a sequence of time instants, $t = 0, 9, 10, 22, 25, 30, 82$ and 100 , respectively, along with $M = 1$, $\epsilon = 0.1$, $\eta_1 = 2\epsilon$, $\eta_2 = 2\epsilon$ and $\tau = 0.125$.

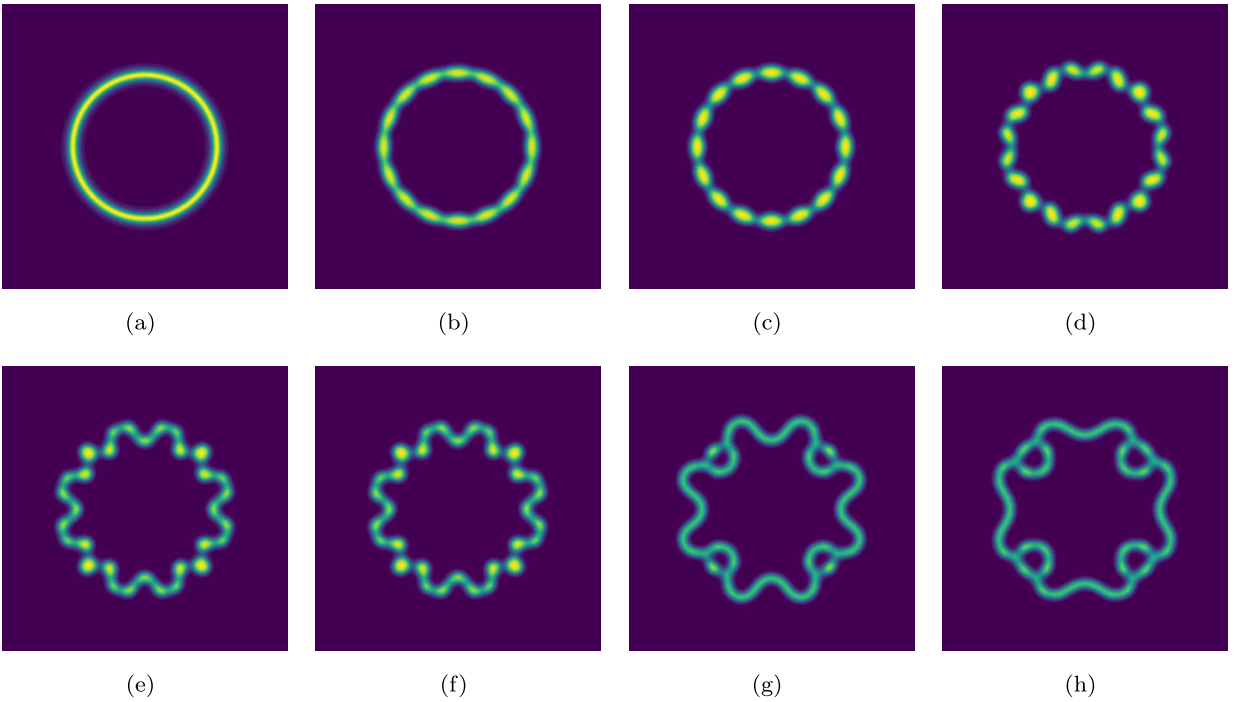


Fig. 17. The 2D dynamical evolution of phase variable ϕ simulated by the standard BDF2 scheme along with $\delta t = 1 \times 10^{-3}$, which starts from initial condition (5.27) with $d = 1.4$. The snapshots are the simulation results of pearling bifurcation at a sequence of time instants, $t = 0, 9, 10, 22, 25, 30, 82$ and 100 , respectively, along with $M = 1$, $\epsilon = 0.1$, $\eta_1 = 2\epsilon$, $\eta_2 = 2\epsilon$ and $\tau = 0.125$.

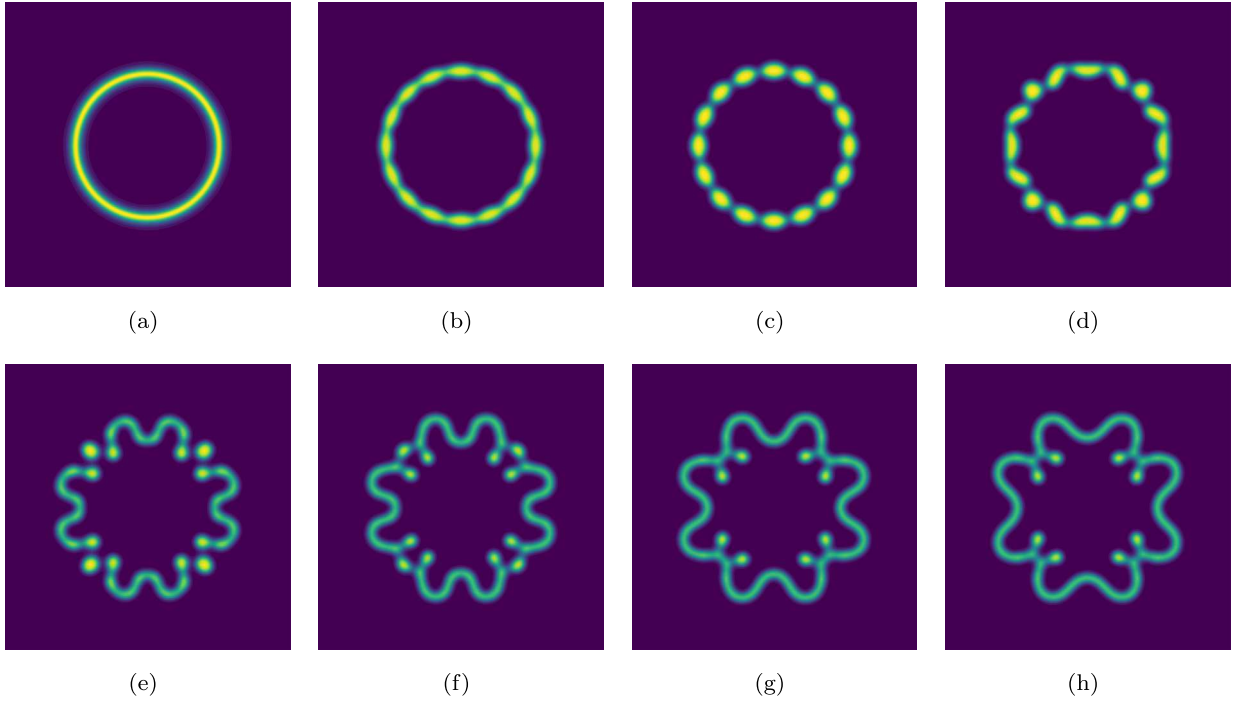


Fig. 18. The 2D dynamical evolution of phase variable ϕ simulated by the BDF2-SAV scheme along with $\delta t = 1 \times 10^{-3}$, which starts from initial condition (5.27) with $d = 1.5$. The snapshots are the simulation results of pearling bifurcation at a sequence of time instants, $t = 0, 13, 15, 24, 30, 34, 70$ and 100 , respectively, along with $M = 1$, $\epsilon = 0.1$, $\eta_1 = 2\epsilon$, $\eta_2 = 2\epsilon$ and $\tau = 0.125$.

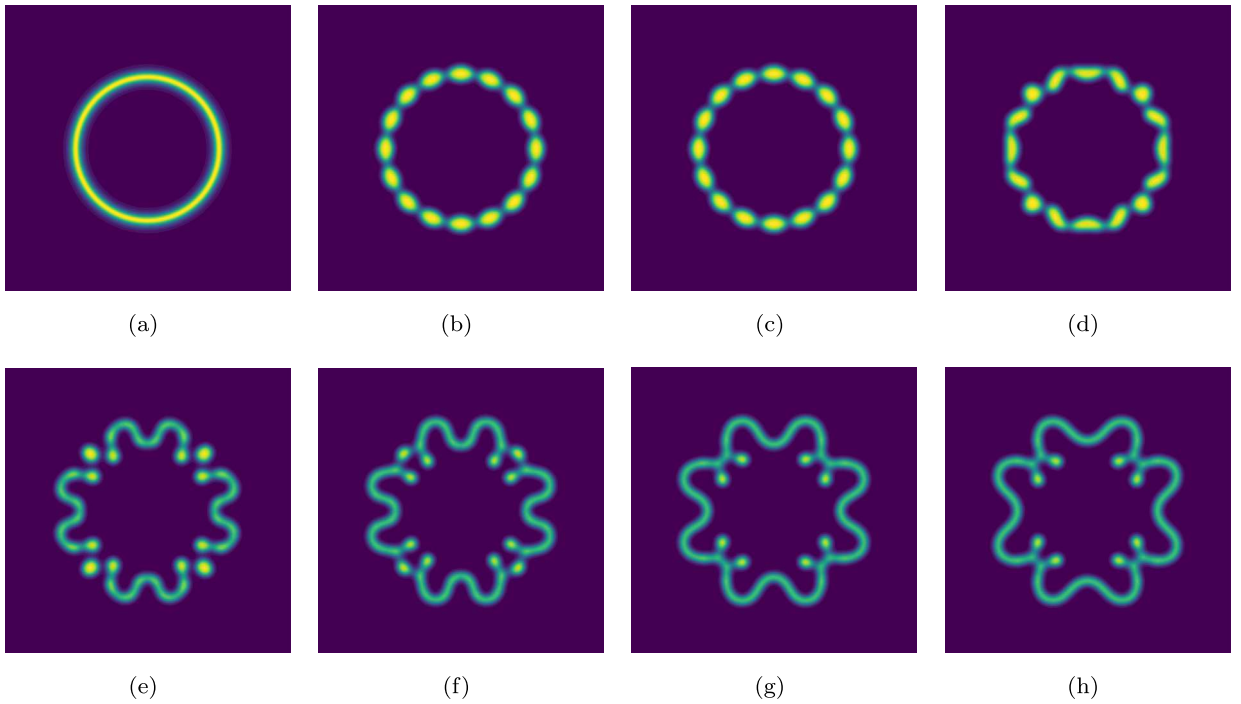


Fig. 19. The 2D dynamical evolution of phase variable ϕ simulated by the standard BDF2 scheme along with $\delta t = 1 \times 10^{-3}$, which starts from initial condition (5.27) with $d = 1.5$. The snapshots are the simulation results of pearling bifurcation at a sequence of time instants, $t = 0, 13, 15, 24, 30, 34, 70$ and 100 , respectively, along with $M = 1$, $\epsilon = 0.1$, $\eta_1 = 2\epsilon$, $\eta_2 = 2\epsilon$ and $\tau = 0.125$.

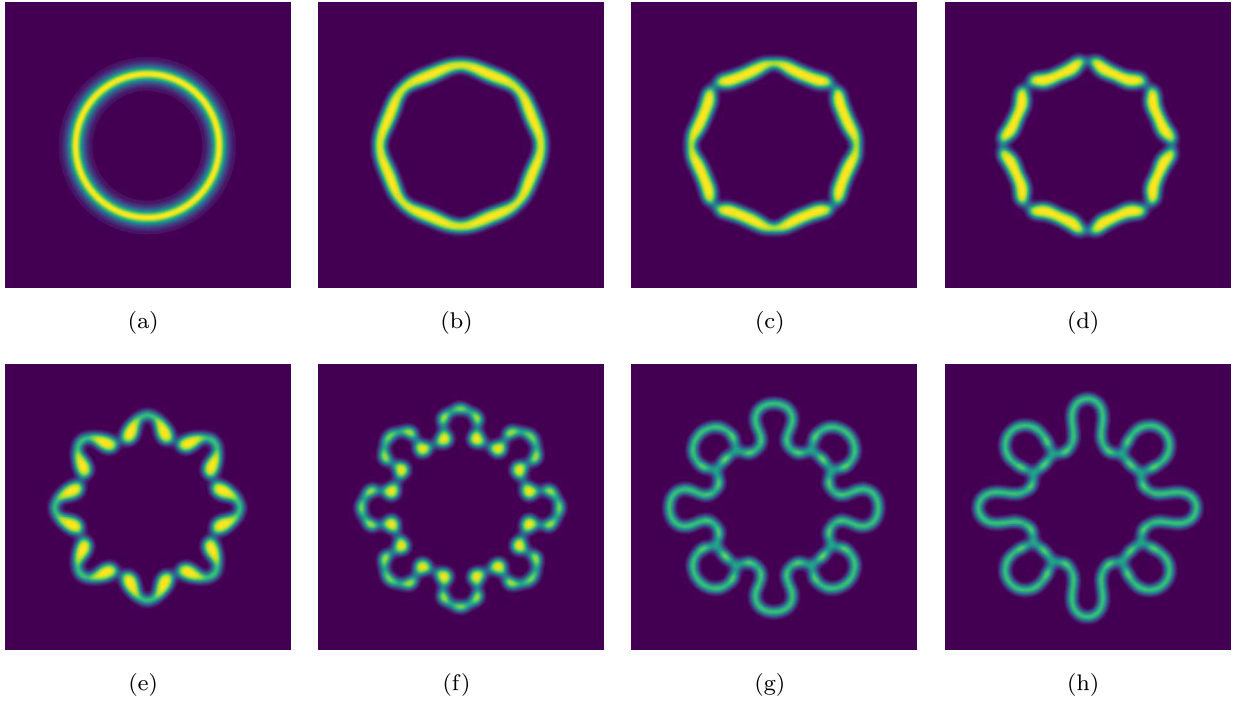


Fig. 20. The 2D dynamical evolution of phase variable ϕ simulated by the BDF2-SAV scheme along with $\delta t = 1 \times 10^{-3}$, which starts from initial condition (5.27) with $d = 2.0$. The snapshots are the simulation results of pearling bifurcation at a sequence of time instants, $t = 0, 48, 50, 51, 55, 60, 70$ and 100 , respectively, along with $M = 1$, $\epsilon = 0.1$, $\eta_1 = 2\epsilon$, $\eta_2 = 2\epsilon$ and $\tau = 0.125$.

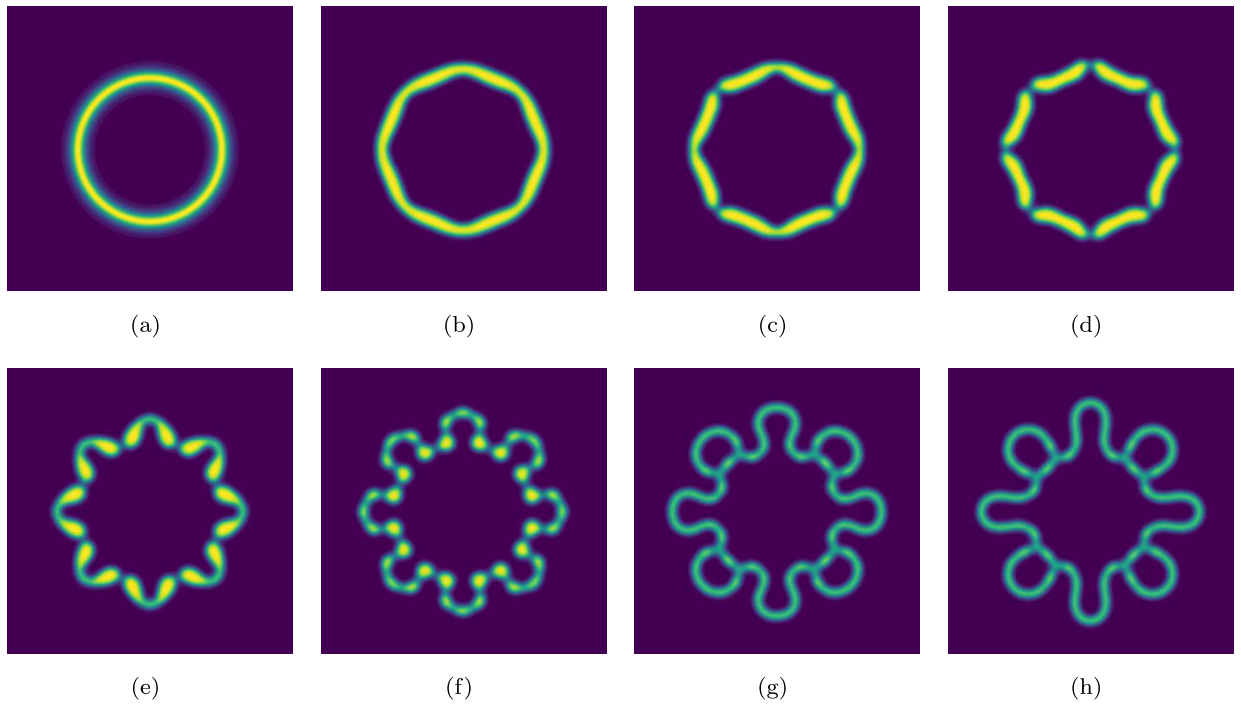


Fig. 21. The 2D dynamical evolution of phase variable ϕ simulated by the standard BDF2 scheme along with $\delta t = 1 \times 10^{-3}$, which starts from initial condition (5.27) with $d = 2.0$. The snapshots are the simulation results of pearling bifurcation at a sequence of time instants, $t = 0, 48, 50, 51, 55, 60, 70$ and 100 , respectively, along with $M = 1$, $\epsilon = 0.1$, $\eta_1 = 2\epsilon$, $\eta_2 = 2\epsilon$ and $\tau = 0.125$.

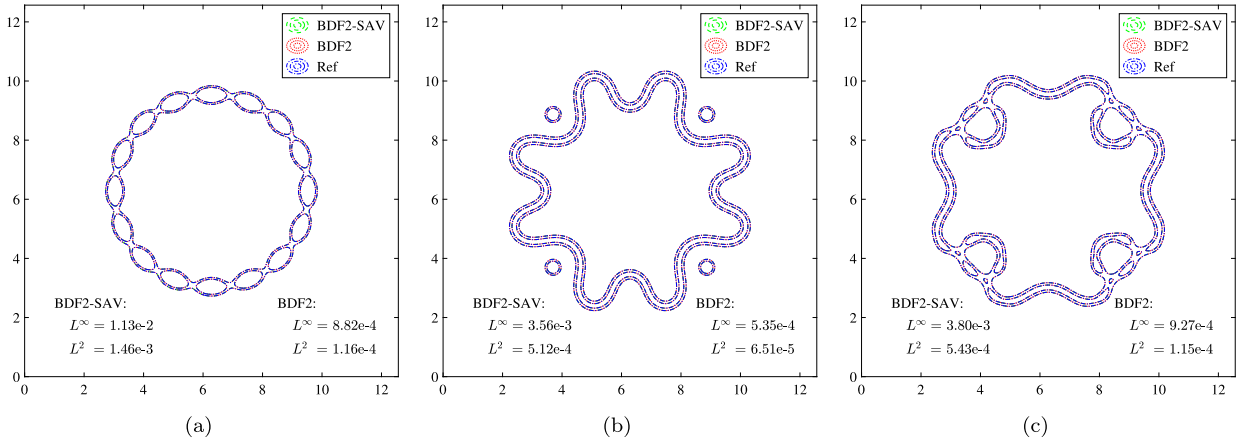


Fig. 22. The accuracy comparisons at $t = 10, 50$, and 100 between scheme BDF2-SAV and scheme BDF2 by using the same large time step $\delta t = 1.0 \times 10^{-3}$ and the same initial condition (5.27) where $d = 1.4$. The level set $\phi = \pm 0.2$ are shown. The “Ref” represents the reference solution computed by the standard BDF2 scheme with time step $\delta t = 1.0 \times 10^{-5}$. The L^∞ and L^2 errors are presented in these pictures.

SAV, BDF2-SAV, CN-SAV, BDF2. In this case, we found that the BDF2 scheme is about 50-100 times more accurate than the others for $\delta t = 1 \times 10^{-3}$, at a cost of a little less than twice the amount of CPU time. Specifically, in the early stage of evolution, where the interface changes drastically, the CPU time used by the BDF2 scheme is about four times that of the SAV schemes. The convergence speed of the PSD solver is accelerated in the later stage, which makes up for a part of the time difference. The solution speed of SAV schemes is always about the same; they each require the solution of two sixth-order linear PDEs at each time step, which is efficiently done with FFTs here. This finding is consistent with the results of the previous benchmark problem. If accuracy becomes an issue, it is likely that the pure BDF2 method would be the method of choice. Finally, we studied the phase separation dynamics in both 2D and 3D spaces. We only provided output for the BDF2-SAV scheme, and the morphologies encountered in the solutions compared well with previous findings.

For the last battery of tests, using the asymmetric version of the FCH equation, we apply the BDF2-SAV and the standard BDF2 schemes to study the phenomenon of the pearling bifurcation. If the initial shape is an elliptical ring, we observed the pearling bifurcations using both schemes. We compared the accuracies and CPU time consumptions of these two schemes, for $\delta t = 1 \times 10^{-3}$ and $\delta t = 1 \times 10^{-4}$, and find that the results of the standard BDF2 scheme are still more accurate than the BDF2-SAV scheme, but slower than the latter for $\delta t = 1 \times 10^{-3}$. However, for $\delta t = 1 \times 10^{-4}$, the BDF2 scheme is faster (and more accurate) than the BDF2-SAV scheme for the same final time $t = 100$. But, the final numerical error of the BDF2 scheme is only slightly smaller than that of the BDF2-SAV scheme.

Finally, if a smooth circular ring is taken as the initial value, we find that both the BDF2-SAV and BDF2 schemes can qualitatively capture the pearling bifurcation phenomenon. Both schemes give similar results for the same time step sizes. This test suggests that both schemes are accurate. In addition, the energy evolution curves also demonstrate the energy stability of the BDF2-SAV scheme, for $\delta t = 1 \times 10^{-3}$. The BDF2 method was only slightly more accurate, though it cost about twice the CPU time.

To summarize, we observed that the SAV schemes often have an advantage in terms of computational efficiency, being up to three times faster in CPU time when a moderately large to large time step size is used. However, when accuracy is counted in the measures of computational efficiency, the classical BDF methods often performed better than the linear SAV methods, with an advantage of up to three digits of precision, mostly due to a smaller local truncation error. If the final target of a computation is relatively high global accuracy, then the method with the least computational time to achieve that desired accuracy is very often classical BDF2. But, these conclusions are not universal; some test results were subtle and ambiguous. This may be due, in part, to our imperfect method for approximating the global numerical errors. Certainly, more advanced testing can be done. We have shown that SAV methods can be constructed in such a way that they are both physically energy stable and accurate. However, they are not a clear choice in practical, real-world computations, because their large LTEs limit their true efficiency.

Finally, our preliminary computations are limited in a couple of ways. First, we have not done exhaustive comparisons of errors for our schemes. This is for the sake of brevity and also because the SAV schemes are new, and we have spent some time establishing their properties and highlighting their implementation issues. Second, we have ignored adaptive time stepping, which should be invaluable at dealing with the multiple time scales in the FCH and similar models. Third, we have only compared the SAV methods against fully implicit BDF methods. Other methods, in particular linear IMEX methods, could and should be considered against the SAV schemes, as they may balance efficiency and accuracy better than the BDF methods used for benchmarking herein. A recently submitted arXiv manuscript [9], addresses most of these points directly.

CRediT authorship contribution statement

Chenhui Zhang is a graduate student. He performed almost all of the computational test results and wrote the SAV codes. He helped to write the first version of the manuscript.

Jie Ouyang advised student ZHANG in China and helped prepare the first version of the manuscript. Dr. OUYANG also contributed to the revised manuscript.

Cheng Wang aided in the manuscript writing and early computational tests.

Steven Wise developed and provided Matlab codes to ZHANG based on the PSD solver for the FCH problem. He helped in the development of the first draft and the current revision. Dr. WISE advised student ZHANG during his six month visit to UTK, where the two developed the ideas for the manuscript.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

CZ and JO would like to thank Professor Xiaofeng Yang, University of South Carolina, for in depth discussions about the SAV method applied to the FCH equation. We are also grateful for the support of the major research plan of National Natural Science Foundation of China (grant numbers 11671321, 11971387, 11901051, 91434201). SMW gratefully acknowledges support from The National Science Foundation grants NSF-DMS 1719854, NSF DMS-2012634, and CW gratefully acknowledges support from The National Science Foundation grant NSF DMS-2012669. Much of this work was performed while CZ was a visiting member of the computational sciences group of SMW at the University of Tennessee (UT) in the 2018–2019 academic year. We gratefully acknowledge the Mathematics Department at UT for support during CZ's visit. SMW is indebted to Professors Keith Promislow (Michigan State University) and Brian Wetton (University of British Columbia) for several enlightening discussions about the FCH equation and early computational results from our study. Additionally, they made several concrete suggestions that, ultimately, improved the manuscript. Thanks, Keith and Brian. Finally, the authors thank the anonymous reviewers whose comments helped to significantly strengthen the paper.

Appendix A. Proof of Theorem 4.2

Proof. By taking the L^2 inner product of (4.1a) with μ^{n+1} , we get

$$(\phi^{n+1} - \phi^n, \mu^{n+1}) = -M\delta t \|\nabla \mu^{n+1}\|^2. \quad (\text{A.1})$$

By taking the L^2 inner product of (4.1b) with $\phi^{n+1} - \phi^n$, we get

$$\begin{aligned} (\mu^{n+1}, \phi^{n+1} - \phi^n) &= \epsilon^4 (\Delta \phi^{n+1}, \Delta \phi^{n+1} - \Delta \phi^n) + S(\nabla(\phi^{n+1} - \phi^n), \nabla(\phi^{n+1} - \phi^n)) \\ &\quad - (2 + \eta)\epsilon^2 (\nabla \phi^n, \nabla(\phi^{n+1} - \phi^n)) + U^{n+1}(X^n, \phi^{n+1} - \phi^n). \end{aligned} \quad (\text{A.2})$$

Multiplying (4.1c) by $2U^{n+1}$ gives

$$2(U^{n+1} - U^n)U^{n+1} = U^{n+1}(X^n, \phi^{n+1} - \phi^n). \quad (\text{A.3})$$

Then, using the identity

$$2a(a - b) = |a|^2 - |b|^2 + |a - b|^2, \quad (\text{A.4})$$

we obtain

$$U^{n+1}(X^n, \phi^{n+1} - \phi^n) = |U^{n+1}|^2 - |U^n|^2 + |U^{n+1} - U^n|^2. \quad (\text{A.5})$$

By using (A.4) again, we can also derive

$$2(\Delta \phi^{n+1}, \Delta \phi^{n+1} - \Delta \phi^n) = \|\Delta \phi^{n+1}\|^2 - \|\Delta \phi^n\|^2 + \|\Delta \phi^{n+1} - \Delta \phi^n\|^2. \quad (\text{A.6})$$

In addition, thanks to

$$2b(a - b) = |a|^2 - |b|^2 - |a - b|^2, \quad (\text{A.7})$$

we easily know that

$$2(\nabla \phi^n, \nabla \phi^{n+1} - \nabla \phi^n) = \|\nabla \phi^{n+1}\|^2 - \|\nabla \phi^n\|^2 - \|\nabla \phi^{n+1} - \nabla \phi^n\|^2. \quad (\text{A.8})$$

So, substituting (A.5), (A.6) and (A.8) into (A.2), we are able to derive the identity

$$\begin{aligned} (\mu^{n+1}, \phi^{n+1} - \phi^n) &= \frac{1}{2}\epsilon^4 \left(\|\Delta\phi^{n+1}\|^2 - \|\Delta\phi^n\|^2 + \|\Delta\phi^{n+1} - \Delta\phi^n\|^2 \right) \\ &\quad - \frac{2+\eta}{2}\epsilon^2 \left(\|\nabla\phi^{n+1}\|^2 - \|\nabla\phi^n\|^2 - \|\nabla\phi^{n+1} - \nabla\phi^n\|^2 \right) \\ &\quad + S\|\nabla(\phi^{n+1} - \phi^n)\|^2 + |U^{n+1}|^2 - |U^n|^2 + |U^{n+1} - U^n|^2. \end{aligned} \quad (\text{A.9})$$

Combining the above equations, i.e., (A.1) and (A.9), we obtain

$$\begin{aligned} -M\delta t\|\nabla\mu^{n+1}\|^2 &= \frac{1}{2}\epsilon^4(\|\Delta\phi^{n+1}\|^2 - \|\Delta\phi^n\|^2) - \frac{2+\eta}{2}\epsilon^2(\|\nabla\phi^{n+1}\|^2 - \|\nabla\phi^n\|^2) \\ &\quad + |U^{n+1}|^2 - |U^n|^2 + S\|\nabla(\phi^{n+1} - \phi^n)\|^2 + \frac{1}{2}\epsilon^4\|\Delta(\phi^{n+1} - \phi^n)\|^2 \\ &\quad + \frac{2+\eta}{2}\epsilon^2\|\nabla(\phi^{n+1} - \phi^n)\|^2 + |U^{n+1} - U^n|^2. \end{aligned}$$

Finally, if we drop some non-negative terms, we obtain the desired result (4.10). $\square$

Appendix B. Proof of Theorem 4.3

Proof. The proof of the solvability is similar to that of the first-order scheme (4.1) and is skipped for the sake of brevity.

Regarding modified-energy stability, taking the L^2 inner product of (4.11a) with μ^{n+1} , we get

$$(3\phi^{n+1} - 4\phi^n + \phi^{n-1}, \mu^{n+1}) = -2M\delta t\|\nabla\mu^{n+1}\|^2. \quad (\text{B.1})$$

By taking the L^2 inner product of (4.11b) with $3\phi^{n+1} - 4\phi^n + \phi^{n-1}$, we have

$$\begin{aligned} (\mu^{n+1}, 3\phi^{n+1} - 4\phi^n + \phi^{n-1}) &= \epsilon^4 (\Delta\phi^{n+1}, \Delta(3\phi^{n+1} - 4\phi^n + \phi^{n-1})) \\ &\quad + S(\nabla(\phi^{n+1} - 2\phi^n + \phi^{n-1}), \nabla(3\phi^{n+1} - 4\phi^n + \phi^{n-1})) \\ &\quad - (2+\eta)\epsilon^2 (\nabla\phi^{*,n+1}, \nabla(3\phi^{n+1} - 4\phi^n + \phi^{n-1})) \\ &\quad + U^{n+1}(X^{*,n+1}, 3\phi^{n+1} - 4\phi^n + \phi^{n-1}). \end{aligned} \quad (\text{B.2})$$

Using the identity

$$2a(3a - 4b + c) = |a|^2 + |2a - b|^2 - |b|^2 - |2b - c|^2 + |a - 2b + c|^2, \quad (\text{B.3})$$

we find

$$\begin{aligned} I_1 &:= \epsilon^4 (\Delta\phi^{n+1}, \Delta(3\phi^{n+1} - 4\phi^n + \phi^{n-1})) \\ &= \frac{1}{2}\epsilon^4 \left((\|\Delta\phi^{n+1}\|^2 + \|\Delta(2\phi^{n+1} - \phi^n)\|^2) - (\|\Delta\phi^n\|^2 + \|\Delta(2\phi^n - \phi^{n-1})\|^2) \right. \\ &\quad \left. + \|\Delta(\phi^{n+1} - 2\phi^n + \phi^{n-1})\|^2 \right). \end{aligned} \quad (\text{B.4})$$

Using another identity

$$(a - 2b + c)(3a - 4b + c) = |a - b|^2 - |b - c|^2 + 2|a - 2b + c|^2, \quad (\text{B.5})$$

we obtain

$$\begin{aligned} I_2 &:= S(\nabla(\phi^{n+1} - 2\phi^n + \phi^{n-1}), \nabla(3\phi^{n+1} - 4\phi^n + \phi^{n-1})) \\ &= S\left(\|\nabla\phi^{n+1} - \nabla\phi^n\|^2 - \|\nabla\phi^n - \nabla\phi^{n-1}\|^2 + 2\|\nabla\phi^{n+1} - 2\nabla\phi^n + \nabla\phi^{n-1}\|^2\right). \end{aligned} \quad (\text{B.6})$$

Finally, using the identity

$$\begin{aligned} 2(2b - c)(3a - 4b + c) &= |a|^2 + |2a - b|^2 - 2|a - b|^2 - (|b|^2 + |2b - c|^2 - 2|b - c|^2) \\ &\quad - 3|a - 2b + c|^2, \end{aligned} \quad (\text{B.7})$$

we get

$$\begin{aligned}
I_3 &:= -(2 + \eta)\epsilon^2 (\nabla \phi^{*,n+1}, \nabla (3\phi^{n+1} - 4\phi^n + \phi^{n-1})) \\
&= -(2 + \eta)\epsilon^2 (\nabla (2\phi^n - \phi^{n-1}), \nabla (3\phi^{n+1} - 4\phi^n + \phi^{n-1})) \\
&= -\frac{2 + \eta}{2}\epsilon^2 \left(\|\nabla \phi^{n+1}\|^2 + \|2\nabla \phi^{n+1} - \nabla \phi^n\|^2 \right. \\
&\quad \left. - 2\|\nabla \phi^{n+1} - \nabla \phi^n\|^2 \right) - 3\|\nabla \phi^{n+1} - 2\nabla \phi^n + \nabla \phi^{n-1}\|^2 \\
&\quad - \left(\|\nabla \phi^n\|^2 + \|2\nabla \phi^n - \nabla \phi^{n-1}\|^2 - 2\|\nabla \phi^n - \nabla \phi^{n-1}\|^2 \right). \tag{B.8}
\end{aligned}$$

Multiplying (4.11c) by $2U^{n+1}$ leads to

$$2U^{n+1}(3U^{n+1} - 4U^n + U^{n-1}) = U^{n+1}(X^{*,n+1}, 3\phi^{n+1} - 4\phi^n + \phi^{n-1}), \tag{B.9}$$

and, according to (B.3), we can derive

$$\begin{aligned}
2U^{n+1}(3U^{n+1} - 4U^n + U^{n-1}) &= |U^{n+1}|^2 + |2U^{n+1} - U^n|^2 - |U^n|^2 - |2U^n - U^{n-1}|^2 \\
&\quad + |U^{n+1} - 2U^n + U^{n-1}|^2. \tag{B.10}
\end{aligned}$$

Combining all the above equations, we obtain

$$\begin{aligned}
-M\delta t \|\nabla \mu^{n+1}\|^2 &= \frac{1}{2}\epsilon^4 \left(\frac{\|\Delta \phi^{n+1}\|^2 + \|\Delta(2\phi^{n+1} - \phi^n)\|^2}{2} - \frac{\|\Delta \phi^n\|^2 + \|\Delta(2\phi^n - \phi^{n-1})\|^2}{2} \right) \\
&\quad + S \frac{\|\nabla \phi^{n+1} - \nabla \phi^n\|^2 - \|\nabla \phi^n - \nabla \phi^{n-1}\|^2}{2} \\
&\quad - \frac{2 + \eta}{2}\epsilon^2 \left(\frac{\|\nabla \phi^{n+1}\|^2 + \|2\nabla \phi^{n+1} - \nabla \phi^n\|^2 - 2\|\nabla \phi^{n+1} - \nabla \phi^n\|^2}{2} \right. \\
&\quad \left. - \frac{\|\nabla \phi^n\|^2 + \|2\nabla \phi^n - \nabla \phi^{n-1}\|^2 - 2\|\nabla \phi^n - \nabla \phi^{n-1}\|^2}{2} \right) \\
&\quad + \frac{|U^{n+1}|^2 + |2U^{n+1} - U^n|^2}{2} - \frac{|U^n|^2 + |2U^n - U^{n-1}|^2}{2} \\
&\quad + \frac{\|\Delta(\phi^{n+1} - 2\phi^n + \phi^{n-1})\|^2}{2} + S\|\nabla \phi^{n+1} - 2\nabla \phi^n + \nabla \phi^{n-1}\|^2 \\
&\quad + \frac{3}{4}(2 + \eta)\epsilon^2 \|\nabla \phi^{n+1} - 2\nabla \phi^n + \nabla \phi^{n-1}\|^2 \\
&\quad + \frac{1}{2}|U^{n+1} - 2U^n + U^{n-1}|^2.
\end{aligned}$$

Finally, if we drop some non-negative terms, we obtain the desired result (4.12). $\square$

Appendix C. Proof of Theorem 4.4

Proof. We omit the proof of the existence and uniqueness of the solution to scheme CN-SAV, which is similar to that of the first-order scheme (4.1). The following is the proof of the unconditional energy stability.

Taking the L^2 inner product of (4.13a) with $\mu^{n+\frac{1}{2}}$, we get

$$(\phi^{n+1} - \phi^n, \mu^{n+\frac{1}{2}}) = -M\delta t \|\nabla \mu^{n+\frac{1}{2}}\|^2. \tag{C.1}$$

By taking the L^2 inner product of (4.13b) with $\phi^{n+1} - \phi^n$, we find

$$\begin{aligned}
(\mu^{n+\frac{1}{2}}, \phi^{n+1} - \phi^n) &= \frac{1}{2}\epsilon^4 (\Delta(\phi^{n+1} + \phi^n), \Delta(\phi^{n+1} - \phi^n)) \\
&\quad + S(\nabla(\phi^{n+1} - 2\phi^n + \phi^{n-1}), \nabla(\phi^{n+1} - \phi^n)) \\
&\quad - (2 + \eta)\epsilon^2 \left(\nabla \phi^{\circ, n+\frac{1}{2}}, \nabla(\phi^{n+1} - \phi^n) \right) \\
&\quad + \frac{1}{2}(U^{n+1} - U^n)(X^{\circ, n+\frac{1}{2}}, \phi^{n+1} - \phi^n). \tag{C.2}
\end{aligned}$$

By applying the point-wise identities

$$2(a-b)(a-2b+c) = |a-b|^2 - |b-c|^2 + |a-2b+c|^2 \quad (\text{C.3})$$

and

$$\frac{1}{2}(3b-c)(a-b) = \frac{1}{2}(|a|^2 - |b|^2) - \frac{1}{4}(|a-b|^2 - |b-c|^2 + |a-2b+c|^2), \quad (\text{C.4})$$

we get these following identities,

$$I_1 := \frac{1}{2}\epsilon^4 (\Delta(\phi^{n+1} + \phi^n), \Delta(\phi^{n+1} - \phi^n)) = \frac{1}{2}\epsilon^4 (\|\Delta\phi^{n+1}\|^2 - \|\Delta\phi^n\|^2), \quad (\text{C.5})$$

$$\begin{aligned} I_2 &:= S(\nabla(\phi^{n+1} - 2\phi^n + \phi^{n-1}), \nabla(\phi^{n+1} - \phi^n)) \\ &= S(\|\nabla\phi^{n+1} - \nabla\phi^n\|^2 - \|\nabla\phi^n - \nabla\phi^{n-1}\|^2 + \|\nabla\phi^{n+1} - 2\nabla\phi^n + \nabla\phi^{n-1}\|^2), \end{aligned} \quad (\text{C.6})$$

and, finally,

$$\begin{aligned} I_3 &:= -(2+\eta)\epsilon^2 (\nabla\phi^{\circ, n+1}, \nabla(\phi^{n+1} - \phi^n)) \\ &= -(2+\eta)\epsilon^2 \left(\nabla\left(\frac{3}{2}\phi^n - \frac{1}{2}\phi^{n-1}\right), \nabla(\phi^{n+1} - \phi^n) \right) \\ &= -\frac{2+\eta}{2}\epsilon^2 \left((\|\nabla\phi^{n+1}\|^2 - \|\nabla\phi^n\|^2) - \frac{1}{2}(\|\nabla\phi^{n+1} - \nabla\phi^n\|^2 \right. \\ &\quad \left. - \|\nabla\phi^n - \nabla\phi^{n-1}\|^2 + \|\nabla\phi^{n+1} - 2\nabla\phi^n + \nabla\phi^{n-1}\|^2) \right). \end{aligned} \quad (\text{C.7})$$

Multiplying (4.13c) with $U^{n+1} + U^n$ leads to

$$\frac{1}{2}(U^{n+1} + U^n) \left(X^{\circ, n+\frac{1}{2}}, \phi^{n+1} - \phi^n \right) = (U^{n+1} - U^n)(U^{n+1} + U^n) = |U^{n+1}|^2 - |U^n|^2. \quad (\text{C.8})$$

Combining all the above equations, we obtain

$$\begin{aligned} 0 \geq & -M\delta t \|\nabla\mu^{n+\frac{1}{2}}\|^2 = \frac{1}{2}\epsilon^4 (\|\Delta\phi^{n+1}\|^2 - \|\Delta\phi^n\|^2) + S \frac{\|\nabla\phi^{n+1} - \nabla\phi^n\|^2 - \|\nabla\phi^n - \nabla\phi^{n-1}\|^2}{2} \\ & - \frac{2+\eta}{2}\epsilon^2 \left((\|\nabla\phi^{n+1}\|^2 - \frac{1}{2}\|\nabla\phi^{n+1} - \nabla\phi^n\|^2) \right. \\ & \left. - (\|\nabla\phi^n\|^2 - \frac{1}{2}\|\nabla\phi^n - \nabla\phi^{n-1}\|^2) \right) + |U^{n+1}|^2 - |U^n|^2 \\ & + S\|\nabla\phi^{n+1} - 2\nabla\phi^n + \nabla\phi^{n-1}\|^2 \\ & + \frac{2+\eta}{4}\epsilon^2 \|\nabla\phi^{n+1} - 2\nabla\phi^n + \nabla\phi^{n-1}\|^2. \end{aligned} \quad (\text{C.9})$$

Finally, if we drop some positive terms, we will obtain the desired result (4.14). $\square$

References

- [1] Luigi Ambrosio, Nicola Gigli, Giuseppe Savaré, Gradient Flows in Metric Spaces and in the Space of Probability Measures, 1 edition, Lectures in Mathematics Eth Zurich, Birkhäuser Verlag, Basel, 2005.
- [2] Andreas C. Aristotelous, Ohannes Karakashian, Steven M. Wise, A mixed discontinuous Galerkin, convex splitting scheme for a modified Cahn-Hilliard equation and an efficient nonlinear multigrid solver, *Discrete Contin. Dyn. Syst., Ser. B* 18 (9) (2013) 2211–2238.
- [3] Claudio Baiocchi, Michel Crouzeix, On the equivalence of A-stability and G-stability, *Appl. Numer. Math.* 5 (1–2) (1989) 19–22.
- [4] Matthew Causley, Hana Cho, Andrew Christlieb, Method of lines transpose: energy gradient flows using direct operator inversion for phase-field models, *SIAM J. Sci. Comput.* 39 (5) (2017) B968–B992.
- [5] Feng Chen, Jie Shen, Efficient spectral-Galerkin methods for systems of coupled second-order equations and their applications, *J. Comput. Phys.* 231 (15) (2012) 5016–5028.
- [6] Long Chen, Xiaozhe Hu, Steven M. Wise, Convergence analysis of the fast subspace descent method for convex optimization problems, *Math. Comput.* 89 (325) (2020) 2249–2282.
- [7] Ying Chen, John S. Lowengrub, Jie Shen, Cheng Wang, Steven M. Wise, Efficient energy stable schemes for isotropic and strongly anisotropic Cahn-Hilliard systems with the Willmore regularization, *J. Comput. Phys.* 365 (2018) 57–73.
- [8] Andrew Christlieb, Jaylan Jones, Keith Promislow, Brian Wetton, Mark Willoughby, High accuracy solutions to energy gradient flows from material science models, *J. Comput. Phys.* 257 (A) (2014) 193–215.
- [9] Andrew Christlieb, Keith Promislow, Zengqiang Tan, Sulin Wang, Brian Wetton, Steven M. Wise, Benchmark computation of morphological complexity in the functionalized Cahn-Hilliard gradient flow, *J. Comput. Phys.* (2020), submitted for publication, in review: <https://arxiv.org/abs/2006.04784>.
- [10] Arjen Doelman, Gugen Hayrapetyan, Keith Promislow, Brian Wetton, Meander and pearling of single-curvature bilayer interfaces in the functionalized Cahn-Hilliard equation, *SIAM J. Math. Anal.* 46 (6) (2014) 3640–3677.
- [11] Peter J. Dowling, Rob Atkin, Brian Vincent, Philippe Bouillot, Oil core-polymer shell microcapsules prepared by internal phase separation from emulsion droplets. I. Characterization and release rates for microcapsules with polystyrene shells, *Langmuir* 20 (26) (2004) 11374–11379.

- [12] Qiang Du, Lili Ju, Xiao Li, Zhonghua Qiao, Stabilized linear semi-implicit schemes for the nonlocal Cahn–Hilliard equation, *J. Comput. Phys.* 363 (2018) 39–54.
- [13] Etienne Emmrich, Error of the two-step BDF for the incompressible Navier–Stokes problem, *Comput. Methods Appl. Math.* 38 (5) (2004) 757–764.
- [14] Etienne Emmrich, Stability and convergence of the two-step BDF for the incompressible Navier–Stokes problem, *Int. J. Nonlinear Sci. Numer. Simul.* 5 (3) (2004) 199–210.
- [15] Etienne Emmrich, Stability and error of the variable two-step BDF for semilinear parabolic problems, *J. Appl. Math. Comput.* 19 (1) (2005) 33–55.
- [16] Etienne Emmrich, Convergence of a time discretization for a class of non-Newtonian fluid flow, *Commun. Math. Sci.* 6 (4) (2008) 827–843.
- [17] Etienne Emmrich, Convergence of the variable two-step BDF time discretisation of nonlinear evolution problems governed by a monotone potential operator, *BIT Numer. Math.* 49 (2) (2009) 297–323.
- [18] Etienne Emmrich, Two-step BDF time discretisation of nonlinear evolution problems governed by monotone operators with strongly continuous perturbations, *Comput. Methods Appl. Math.* 9 (1) (2009) 37–62.
- [19] Etienne Emmrich, David Šiška, Full discretisation of second-order nonlinear evolution equations: strong convergence and applications, *Comput. Methods Appl. Math.* 11 (4) (2008) 441–459.
- [20] David J. Eyre, Unconditionally gradient stable time marching the Cahn–Hilliard equation, in: *MRS Online Proceedings Library Archive*, vol. 529, 1998, pp. 39–46.
- [21] Wenqiang Feng, Zhen Guan, John Lowengrub, Cheng Wang, Steven M. Wise, Ying Chen, A uniquely solvable, energy stable numerical scheme for the functionalized Cahn–Hilliard equation and its convergence analysis, *J. Sci. Comput.* 76 (3) (2018) 1938–1967.
- [22] Wenqiang Feng, Abner J. Salgado, Cheng Wang, Steven M. Wise, Preconditioned steepest descent methods for some nonlinear elliptic equations involving p-Laplacian terms, *J. Comput. Phys.* 334 (2017) 45–67.
- [23] Wenqiang Feng, Cheng Wang, Steven M. Wise, Zhengru Zhang, A second-order energy stable backward differentiation formula method for the epitaxial thin film equation with slope selection, *Numer. Methods Partial Differ. Equ.* 34 (6) (2018) 1975–2007.
- [24] Xiaobing Feng, Andreas Prohl, Analysis of a fully discrete finite element method for the phase field model and approximation of its sharp interface limits, *Math. Comput.* 73 (246) (2004) 541–567.
- [25] Nir Gavish, Gurgun Hayrapetyan, Keith Promislow, Li Yang, Curvature driven flow of bi-layer interfaces, *Physica D* 240 (7) (2011) 675–693.
- [26] Nir Gavish, Jaylan Jones, Zhengfu Xu, Andrew Christlieb, Keith Promislow, Variational models of network formation and ion transport: applications to perfluorosulfonate ionomer membranes, *Polymers* 4 (1) (2012) 630–655.
- [27] Gerhard Gompper, Michael Schick, Correlation between structural and interfacial properties of amphiphilic systems, *Phys. Rev. Lett.* 65 (9) (1990) 1116–1119.
- [28] Yuezheng Gong, Jia Zhao, Qi Wang, Arbitrarily high-order unconditionally energy stable schemes for gradient flow models using the scalar auxiliary variable approach, *J. Comput. Phys.* 419 (2020) 109610.
- [29] Zhen Guan, John S. Lowengrub, Cheng Wang, Steven M. Wise, Second order convex splitting schemes for periodic nonlocal Cahn–Hilliard and Allen–Cahn equations, *J. Comput. Phys.* 277 (2014) 48–71.
- [30] Zhen Guan, Cheng Wang, Steven M. Wise, A convergent convex splitting scheme for the periodic nonlocal Cahn–Hilliard equation, *Numer. Math.* 128 (2) (2014) 377–406.
- [31] Ruihan Guo, Yan Xu, Zhengfu Xu, Local discontinuous Galerkin methods for the functionalized Cahn–Hilliard equation, *J. Sci. Comput.* 63 (3) (2015) 913–937.
- [32] Sumeet Jain, Frank S. Bates, Consequences of nonergodicity in aqueous binary PEO–PB micellar dispersions, *Macromolecules* 37 (4) (2004) 1511–1523.
- [33] Noa Kraitzman, Keith Promislow, An overview of network bifurcations in the functionalized Cahn–Hilliard free energy, in: J.P. Bourguignon, R. Jeltsch, A.A. Pinto, M. Viana (Eds.), *Mathematics of Energy and Climate Change*, in: *CIM Series in Mathematical Sciences*, vol. 2, 2015, pp. 191–214.
- [34] Dong Li, Zhonghua Qiao, On the stabilization size of semi-implicit Fourier–spectral methods for 3D Cahn–Hilliard equations, *Commun. Math. Sci.* 15 (6) (2017) 1489–1506.
- [35] Weijia Li, Wenbin Chen, Cheng Wang, Yue Yan, Ruijian He, A second order energy stable linear scheme for a thin film model without slope selection, *J. Sci. Comput.* 76 (3) (2018) 1905–1937.
- [36] Fei Liu, Jie Shen, Stabilized semi-implicit spectral deferred correction methods for Allen–Cahn and Cahn–Hilliard equations, *Math. Methods Appl. Sci.* 38 (18) (2015) 4564–4575.
- [37] Keith Promislow, Brian Wetton, Pem fuel cells: a mathematical overview, *SIAM J. Appl. Math.* 70 (2) (2009) 369–409.
- [38] Jie Shen, Jie Xu, Convergence and error analysis for the scalar auxiliary variable (SAV) schemes to gradient flows, *SIAM J. Numer. Anal.* 56 (5) (2018) 2895–2912.
- [39] Jie Shen, Jie Xu, Jiang Yang, The scalar auxiliary variable (SAV) approach for gradient flows, *J. Comput. Phys.* 353 (2018) 407–416.
- [40] Jie Shen, Jie Xu, Jiang Yang, A new class of efficient and robust energy stable schemes for gradient flows, *SIAM Rev.* 61 (3) (2019) 474–506.
- [41] Jie Shen, Xiaofeng Yang, Numerical approximations of Allen–Cahn and Cahn–Hilliard equations, *Discrete Contin. Dyn. Syst.* 28 (4) (2010) 1669–1691.
- [42] Philippe Vignal, Lisandro Dalcin, Donald L. Brown, Nathan Collier, Victor M. Calo, An energy-stable convex splitting for the phase-field crystal equation, *Comput. Struct.* 158 (2015) 355–368.
- [43] Lin Wang, Haijun Yu, On efficient second order stabilized semi-implicit schemes for the Cahn–Hilliard phase-field equation, *J. Sci. Comput.* 77 (2) (2018) 1185–1209.
- [44] X. Wu, G.J. Van Zwieten, K.G. Van der Zee, Stabilized second-order convex splitting schemes for Cahn–Hilliard models with application to diffuse-interface tumor-growth models, *Int. J. Numer. Methods Biomed. Eng.* 30 (2) (2014) 180–203.
- [45] Yue Yan, Wenbin Chen, Cheng Wang, Steven M. Wise, A second-order energy stable BDF numerical scheme for the Cahn–Hilliard equation, *Commun. Comput. Phys.* 23 (2018) 572–602.
- [46] Wen Zhou, Jie Ouyang, Xiaodong Wang, Jin Su, Binxin Yang, Numerical simulation of viscoelastic fluid flows using a robust FVM framework on triangular grid, *J. Non-Newton. Fluid Mech.* 236 (2016) 18–34.