

Fast and stable explicit operator splitting methods for phase-field models



Yuanzhen Cheng^a, Alexander Kurganov^{a,*}, Zhuolin Qu^a, Tao Tang^b

^a Mathematics Department, Tulane University, New Orleans, LA 70118, USA

^b Department of Mathematics, Hong Kong Baptist University, Kowloon Tong, Kowloon, Hong Kong

ARTICLE INFO

Article history:

Received 11 April 2015

Received in revised form 27 August 2015

Accepted 3 September 2015

Available online 25 September 2015

Keywords:

Phase-field models

Molecular beam epitaxy equation

Cahn–Hilliard equation

Operator splitting methods

Semi-discrete finite-difference schemes

Large stability domain explicit Runge–Kutta methods

Pseudo-spectral methods

Adaptive time-stepping

ABSTRACT

Numerical simulations of phase-field models require long time computations and therefore it is necessary to develop efficient and highly accurate numerical methods. In this paper, we propose fast and stable explicit operator splitting methods for both one- and two-dimensional nonlinear diffusion equations for thin film epitaxy with slope selection and the Cahn–Hilliard equation. The equations are split into nonlinear and linear parts. The nonlinear part is solved using a method of lines together with an efficient large stability domain explicit ODE solver. The linear part is solved by a pseudo-spectral method, which is based on the exact solution and thus has no stability restriction on the time-step size. We demonstrate the performance of the proposed methods on a number of one- and two-dimensional numerical examples, where different stages of coarsening such as the initial preparation, alternating rapid structural transition and slow motion can be clearly observed.

© 2015 Elsevier Inc. All rights reserved.

1. Introduction

Phase-field models have been recently introduced to describe interfacial phenomena. They were originally derived for the microstructure evolution and phase transition, but have been recently extended to many other physical phenomena, such as solid–solid transitions, growth of cancerous tumors, phase separation of block copolymers, dewetting and rupture of thin liquid films and infiltration of water into porous medium.

Two of these phase-field models have attracted much attention: the molecular beam epitaxy (MBE) equation with slope selection

$$u_t = -\delta \Delta^2 u + \nabla \cdot f(\nabla u), \quad (x, y) \in \Omega \subset \mathbb{R}^2, \quad t \in (0, T], \quad (1.1)$$

and the Cahn–Hilliard (CH) equation

$$u_t = -\delta \Delta^2 u + \Delta f(u), \quad (x, y) \in \Omega \subset \mathbb{R}^2, \quad t \in (0, T]. \quad (1.2)$$

In this paper, we consider

$$f(\varphi) = \varphi|\varphi|^2 - \varphi,$$

* Corresponding author.

E-mail addresses: ycheng5@tulane.edu (Y. Cheng), kurganov@math.tulane.edu (A. Kurganov), zqu1@tulane.edu (Z. Qu), ttang@math.hkbu.edu.hk (T. Tang).

for which the two phase-field models (1.1) and (1.2) become

$$u_t = -\delta\Delta^2 u + \nabla \cdot (|\nabla u|^2 \nabla u - \nabla u), \quad (x, y) \in \Omega \subset \mathbb{R}^2, \quad t \in (0, T], \tag{1.3}$$

and

$$u_t = -\delta\Delta^2 u + \Delta(u^3 - u), \quad (x, y) \in \Omega \subset \mathbb{R}^2, \quad t \in (0, T]. \tag{1.4}$$

In (1.3), u is a scaled height function of epitaxial growth of thin films in a co-moving frame and the parameter δ is a positive surface diffusion constant. In (1.4), u represents the concentration of one of the two metallic components of the alloy, and the positive parameter δ represents the interfacial width, which is small compared to the characteristic length of the laboratory scale. An important feature of these two equations is that they can be viewed as the gradient flow of the following energy functionals:

$$E(u) = \int_{\Omega} \left[\frac{\delta}{2} |\Delta u|^2 + \frac{1}{4} (|\nabla u|^2 - 1)^2 \right] dx dy \tag{1.5}$$

for the MBE equation and

$$E(u) = \int_{\Omega} \left[\frac{\delta}{2} |\nabla u|^2 + \frac{1}{4} (u^2 - 1)^2 \right] dx dy \tag{1.6}$$

for the CH one. As it has been shown in [4,15], both energy functionals decay in time:

$$E(u(t)) \leq E(u(s)), \quad \forall t \geq s.$$

Development of highly accurate and efficient numerical methods for (1.3) and (1.4) is a challenging task. Since explicit schemes usually suffer from severe stability restrictions caused by the presence of high-order derivative terms and do not obey the energy decay property, semi-implicit schemes are widely used. In [30], a combined spectral and large time-stepping method was studied for the MBE equation, in which an extra term was added to substantially improve the stability condition. The same method was applied to the CH equation in [13]. However, this artificial stabilization term depends on the unknown numerical solutions and if it is taken improperly, the resulting numerical scheme would be unstable. In [21], unconditionally energy stable finite-difference schemes were introduced and an adaptive time-stepping strategy was proposed to select time-steps adaptively based on the time variation of the energy. This technique was also successfully applied in the simulations of the CH equation in [31]. In [10], a high-order and energy stable scheme was developed to simulate some phase-field models by combining the semi-implicit spectral deferred correction method and the energy stable convex splitting technique. In [23], a set of unconditionally stable, unconditionally uniquely solvable and second-order schemes for general gradient flows of Ehrlich–Schwoebel energy type with a specific application to the MBE equation was presented. In addition, a variety of finite-element based unconditionally energy-stable schemes for the CH equation were proposed in [11,12], including first- and second-order in time linear schemes as well as an adaptive time-stepping algorithm. A detailed review of the recent updates on numerical methods for the CH equation and its applicability to related energy-based models, including phase-field models, can be found in [26].

In this paper, we develop accurate, efficient and robust explicit methods for both (1.3) and (1.4) subject to periodic boundary conditions. Our methods, which are described in detail in Section 2 and Section 3, are based on the large stability domain explicit Runge–Kutta methods [2,3,14,18] and the fast explicit operator splitting method proposed in [5,6,8,9] (see also [7]) in the context of convection–diffusion equations.

Following the approach in [5,6,8,9], we split equation (1.3) into the nonlinear,

$$u_t = \nabla \cdot (|\nabla u|^2 \nabla u), \tag{1.7}$$

and linear,

$$u_t = -\Delta u - \delta\Delta^2 u, \tag{1.8}$$

parts. We denote by $\mathcal{S}_{\mathcal{N}}$ the exact solution operator associated with (1.7) and by $\mathcal{S}_{\mathcal{L}}$ the exact solution operator associated with (1.8). Notice that the corresponding energy functionals,

$$E_{\mathcal{N}}(u) = \frac{1}{4} \int_{\Omega} |\nabla u|^4 dx dy \tag{1.9}$$

and

$$E_{\mathcal{L}}(u) = \int_{\Omega} \left(\frac{\delta}{2} |\Delta u|^2 - \frac{1}{2} |\nabla u|^2 + \frac{1}{4} \right) dx dy \tag{1.10}$$

decay. Then, introducing a (small) splitting step Δt , the solution of the original equation (1.3) (which is assumed to be available at time t) is evolved using the Strang splitting method [16,17,25], one step of which can be written as

$$u(x, y, t + \Delta t) = \mathcal{S}_{\mathcal{L}}(\Delta t/2)\mathcal{S}_{\mathcal{N}}(\Delta t)\mathcal{S}_{\mathcal{L}}(\Delta t/2)u(x, y, t).$$

A similar splitting approach is applied to equation (1.4), for which the linear part is still (1.8) and the nonlinear one is

$$u_t = \Delta(u^3). \tag{1.11}$$

As in the case of the MBE equation, the corresponding energy functionals,

$$E_{\mathcal{N}}(u) = \frac{1}{4} \int_{\Omega} u^4 dx dy \tag{1.12}$$

and

$$E_{\mathcal{L}}(u) = \int_{\Omega} \left(\frac{\delta}{2} |\nabla u|^2 - \frac{1}{2} u^2 + \frac{1}{4} \right) dx dy \tag{1.13}$$

decay. We stress that even though the linear parts of equations (1.3) and (1.4) are the same, the functionals (1.10) and (1.13) are different since they are associated with the corresponding parts of the energy functionals (1.5) and (1.6).

In order to implement the splitting method, the exact solution operators $\mathcal{S}_{\mathcal{N}}$ and $\mathcal{S}_{\mathcal{L}}$ have to be replaced by their numerical approximations. Note that one of the main advantages of the operator splitting technique is the fact that the nonlinear, (1.7) and (1.11), and linear, (1.8), subproblems, which are of different nature, can be solved numerically by different methods. First, using the method of lines, (1.7) and (1.11) can be reduced to systems of ODEs, which can be efficiently and accurately integrated by large stability domain explicit ODE solvers [2,3,14,18]. Second, since (1.8) is linear, one can solve it (practically) exactly using, for example, the pseudo-spectral method. This way, no stability restrictions on solving (1.8) are imposed. A detailed description of an efficient implementation of the proposed fast and stable explicit operator splitting methods is given in Section 2 and Section 3.

The paper is organized as follows. In Section 2, we build $2m$ th-order semi-discrete finite-difference schemes for (1.7) and (1.11). The resulting stiff system of ODE is then solved by an efficient large stability domain explicit ODE solver [1,19]. In Section 3, we develop a pseudo-spectral method for the linear equation (1.8). In Section 5, we demonstrate the performance of the proposed fast and stable explicit operator splitting methods on a number of one- (1-D) and two-dimensional (2-D) numerical examples, where different stages of coarsening such as the initial preparation, alternating rapid structural transition and slow motion can be clearly observed.

2. Finite-difference methods for (1.7) and (1.11)

In this section, we propose efficient explicit finite-difference methods for the degenerate parabolic equations (1.7) and (1.11). These methods are based on the semi-discretization of (1.7) and (1.11) followed by the use of an efficient and accurate ODE solver. The ODE solver will be utilized to evolve the solutions of (1.7) and (1.11) from time t to $t + \Delta t$. We note that in a general case the time-steps of the ODE solver denoted by Δt_{ODE} will be smaller than the splitting step Δt so that the approximation of $\mathcal{S}_{\mathcal{N}}(\Delta t)$ will typically require several Δt_{ODE} steps.

2.1. Finite-difference schemes for $u_t = (u_x^3)_x$

In this section, we design $2m$ th-order centered-difference schemes for the 1-D version of (1.7):

$$u_t = (u_x^3)_x, \quad x \in [0, L], \quad t \in (0, T]. \tag{2.1}$$

We consider a uniform grid with nodes x_j , such that $x_{j+1} - x_j = \Delta x, \forall j$, and introduce the following $2m$ th-order discrete approximation of the $\frac{\partial}{\partial x}$ operator:

$$(\psi_x)_j := \sum_{p=-m}^m \alpha_p \psi_{j+p} = \psi_x(x_j) + \mathcal{O}((\Delta x)^{2m}). \tag{2.2}$$

For example, when $m = 2$, we obtain a fourth-order centered-difference approximation by taking

$$\alpha_1 = -\alpha_{-1} = \frac{2}{3\Delta x}, \quad \alpha_2 = -\alpha_{-2} = -\frac{1}{12\Delta x}.$$

Equipped with the above approximation of spacial derivatives, we discretize equation (2.1) using the method of lines as follows:

$$\frac{du_j}{dt}(t) = \sum_{p=-m}^m \alpha_p H_{j+p}(t) =: F_j(t), \tag{2.3}$$

where $u_j(t)$ denotes the computed point value of the solution at (x_j, t) , and

$$H_j(t) := (u_x)_j^3(t) \quad \text{with} \quad (u_x)_j(t) := \sum_{p=-m}^m \alpha_p u_{j+p}(t). \tag{2.4}$$

Note that the above quantities depend on t , but for the sake of brevity we will suppress this dependence from now on.

Remark 2.1. One can show that the coefficients $\{\alpha_p\}$ satisfy the following conditions:

$$\alpha_0 = 0 \quad \text{and} \quad \alpha_p + \alpha_{-p} = 0, \quad p \neq 0. \tag{2.5}$$

Theorem 2.1. The semi-discrete schemes (2.3), (2.4) satisfy the following energy decay property:

$$\frac{d}{dt} E_{\mathcal{N}}^\Delta \leq 0,$$

where $E_{\mathcal{N}}^\Delta$ is a 1-D discrete version of the energy functional (1.9):

$$E_{\mathcal{N}}^\Delta := \frac{1}{4} \sum_j (u_x)_j^4 \Delta x.$$

Proof. Using (2.3)–(2.5) and the periodicity of computed solutions, one can obtain the following energy estimate:

$$\begin{aligned} \frac{d}{dt} \left(\frac{1}{4} \sum_j (u_x)_j^4 \right) &= \sum_j (u_x)_j^3 \frac{d}{dt} [(u_x)_j] \stackrel{(2.4)}{=} \sum_j H_j \frac{d}{dt} \left[\sum_{p=-m}^m \alpha_p u_{j+p} \right] \stackrel{(2.3)}{=} \sum_j H_j \sum_{p=-m}^m \alpha_p F_{j+p} \\ &= \sum_{p=-m}^m \alpha_p \sum_j H_j F_{j+p} = \sum_{p=-m}^m \alpha_p \sum_j H_{j-p} F_j = \sum_j F_j \sum_{p=-m}^m \alpha_p H_{j-p} \\ &= \sum_j F_j \sum_{p=-m}^m \alpha_{-p} H_{j+p} \stackrel{(2.5)}{=} \sum_j F_j \sum_{p=-m}^m (-\alpha_p) H_{j+p} \stackrel{(2.3)}{=} - \sum_j F_j^2 \leq 0. \quad \square \end{aligned}$$

2.2. Finite-difference schemes for $u_t = \nabla \cdot [|\nabla u|^2 \nabla u]$

We now turn to the 2-D equation (1.7). We consider a uniform grid with nodes (x_j, y_k) , such that $x_{j+1} - x_j = \Delta x, \forall j, y_{k+1} - y_k = \Delta y, \forall k$, and introduce the following $2m$ th-order discrete approximation of the $\frac{\partial}{\partial x}$ and $\frac{\partial}{\partial y}$ operators:

$$\begin{aligned} (\psi_x)_{j,k} &:= \sum_{p=-m}^m \alpha_p \psi_{j+p,k} = \psi_x(x_j, y_k) + \mathcal{O}((\Delta x)^{2m}), \\ (\psi_y)_{j,k} &:= \sum_{p=-m}^m \beta_p \psi_{j,k+p} = \psi_y(x_j, y_k) + \mathcal{O}((\Delta y)^{2m}). \end{aligned} \tag{2.6}$$

For example, when $m = 2$, we obtain a fourth-order centered-difference approximation by taking

$$\alpha_1 = -\alpha_{-1} = \frac{2}{3\Delta x}, \quad \alpha_2 = -\alpha_{-2} = -\frac{1}{12\Delta x}, \quad \beta_1 = -\beta_{-1} = \frac{2}{3\Delta y}, \quad \beta_2 = -\beta_{-2} = -\frac{1}{12\Delta y}.$$

Equipped with the above approximation of spacial derivatives, $2m$ th-order semi-discrete finite-difference schemes for (1.7) read:

$$\frac{du_{j,k}}{dt} = \sum_{p=-m}^m \alpha_p H_{j+p,k}^x + \sum_{p=-m}^m \beta_p H_{j,k+p}^y =: F_{j,k}, \tag{2.7}$$

where

$$H_{j,k}^x := (u_x)_{j,k}^3 + (u_y)_{j,k}^2 (u_x)_{j,k} \quad \text{and} \quad H_{j,k}^y := (u_y)_{j,k}^3 + (u_x)_{j,k}^2 (u_y)_{j,k} \tag{2.8}$$

with

$$(u_x)_{j,k} := \sum_{p=-m}^m \alpha_p u_{j+p,k} \quad \text{and} \quad (u_y)_{j,k} := \sum_{p=-m}^m \beta_p u_{j,k+p}. \tag{2.9}$$

Remark 2.2. One can show that the coefficients $\{\alpha_p\}$ and $\{\beta_p\}$ satisfy the following conditions:

$$\alpha_0 = 0, \beta_0 = 0 \quad \text{and} \quad \alpha_p + \alpha_{-p} = 0, \beta_p + \beta_{-p} = 0, \quad p \neq 0. \tag{2.10}$$

Theorem 2.2. The semi-discrete schemes (2.7)–(2.9) satisfy the following energy decay property:

$$\frac{d}{dt} E_{\mathcal{N}}^{\Delta} \leq 0,$$

where $E_{\mathcal{N}}^{\Delta}$ is a 2-D discrete version of the energy functional (1.9):

$$E_{\mathcal{N}}^{\Delta} := \frac{1}{4} \sum_j |\nabla_h u_{j,k}|^4 \Delta x \Delta y$$

with $\nabla_h u_{j,k} := ((u_x)_{j,k}, (u_y)_{j,k})^T$.

Proof. Using (2.7)–(2.10) and the periodicity of computed solutions, one can obtain the following energy estimate:

$$\begin{aligned} \frac{d}{dt} \left(\frac{1}{4} \sum_{j,k} |\nabla_h u_{j,k}|^4 \right) &\stackrel{(2.8)}{=} \sum_{j,k} H_{j,k}^x \frac{d}{dt} [(u_x)_{j,k}] + \sum_{j,k} H_{j,k}^y \frac{d}{dt} [(u_y)_{j,k}] \\ &\stackrel{(2.9)}{=} \sum_{j,k} H_{j,k}^x \frac{d}{dt} \left[\sum_{p=-m}^m \alpha_p u_{j+p,k} \right] + \sum_{j,k} H_{j,k}^y \frac{d}{dt} \left[\sum_{p=-m}^m \beta_p u_{j,k+p} \right] \\ &\stackrel{(2.7)}{=} \sum_{j,k} H_{j,k}^x \sum_{p=-m}^m \alpha_p F_{j+p,k} + \sum_{j,k} H_{j,k}^y \sum_{p=-m}^m \beta_p F_{j,k+p} \\ &\stackrel{(2.10)}{=} - \sum_{j,k} F_{j,k} \sum_{p=-m}^m \alpha_p H_{j+p,k}^x - \sum_{j,k} F_{j,k} \sum_{p=-m}^m \beta_p H_{j,k+p}^y \stackrel{(2.7)}{=} - \sum_{j,k} F_{j,k}^2 \leq 0. \quad \square \end{aligned}$$

2.3. Finite-difference schemes for $u_t = \Delta(u^3)$

We now design semi-discrete finite-difference schemes for the 2-D CH equation (1.11). We use the same grids and the same 2mth-order discrete approximation of the $\frac{\partial}{\partial x}$ and $\frac{\partial}{\partial y}$ operators as in Section 2.2. Then, 2mth-order semi-discrete finite-difference schemes for (1.11) read:

$$\frac{du_{j,k}}{dt} = \sum_{p=-m}^m \alpha_p H_{j+p,k}^x + \sum_{p=-m}^m \beta_p H_{j,k+p}^y =: F_{j,k}, \tag{2.11}$$

where

$$H_{j,k}^x := \sum_{p=-m}^m \alpha_p u_{j+p,k}^3 \quad \text{and} \quad H_{j,k}^y := \sum_{p=-m}^m \beta_p u_{j,k+p}^3. \tag{2.12}$$

Theorem 2.3. The semi-discrete schemes (2.11), (2.12) satisfy the following energy decay property:

$$\frac{d}{dt} E_{\mathcal{N}}^{\Delta} \leq 0,$$

where $E_{\mathcal{N}}^{\Delta}$ is a 2-D discrete version of the energy functional (1.12):

$$E_{\mathcal{N}}^{\Delta} := \frac{1}{4} \sum_j u_{j,k}^4 \Delta x \Delta y.$$

Proof. Using (2.10)–(2.12) and the periodicity of computed solutions, one can obtain the following energy estimate:

$$\begin{aligned} \frac{d}{dt} \left(\frac{1}{4} \sum_{j,k} u_{j,k}^4 \right) &= \sum_{j,k} u_{j,k}^3 \frac{du_{j,k}}{dt} \stackrel{(2.11)}{=} \sum_{j,k} \sum_{p=-m}^m \alpha_p H_{j+p,k}^x u_{j,k}^3 + \sum_{j,k} \sum_{p=-m}^m \beta_p H_{j,k+p}^y u_{j,k}^3 \\ &\stackrel{(2.10)}{=} - \sum_{j,k} H_{j,k}^x \sum_{p=-m}^m \alpha_p u_{j+p,k}^3 - \sum_{j,k} H_{j,k+p}^y \sum_{p=-m}^m \beta_p u_{j,k+p}^3 \stackrel{(2.12)}{=} - \sum_{j,k} \left[(H_{j,k}^x)^2 + (H_{j,k}^y)^2 \right] \leq 0. \quad \square \end{aligned}$$

2.4. Large stability domain explicit ODE solver

The ODE systems (2.3), (2.7) and (2.11) have to be solved numerically. Recall that explicit ODE solvers typically require time-steps to be $\Delta t_{\text{ODE}} \sim (\Delta x)^2$, while implicit ODE solvers can be made unconditionally stable. However, the accuracy requirements would limit time-step size and since a large nonlinear algebraic system of equations has to be solved at each time-step, implicit methods may not be efficient. Here, we apply the explicit third-order large stability domain Runge–Kutta method, developed in [18,19]. This method belongs to a class of Runge–Kutta–Chebyshev methods (see, e.g., [14,24,27–29]), which allow one to use much larger time-steps compared with the standard explicit Runge–Kutta methods. In practice, when the problem is not too stiff as in the case of ODEs arising in finite-difference approximation of parabolic PDEs, these methods preserve all the advantages of explicit methods and are typically more efficient than implicit methods (see [2,3,18,24,29] for details). We have implemented the code DUMKA3 [19], which incorporates the embedded formulas that permit an efficient stepsize control. The efficiency of DUMKA3 is further improved when the user provides an upper bound on the time-step stability restriction for the forward Euler method. We therefore establish such bounds in the following three theorems.

Theorem 2.4. Assume that the system of ODEs (2.3), (2.4) is numerically integrated by the forward Euler method from time t to $t + \Delta t_{\text{FE}}$ and that the following CFL condition holds:

$$\Delta t_{\text{FE}} \leq \frac{1}{am} \cdot \frac{1}{\max_j (u_x)_j^2}, \quad a := \sum_{p=-m}^m \alpha_p^2, \quad (2.13)$$

where α_p are the coefficients in (2.2) and $(u_x)_j$ are given by (2.4). Then

$$\|u(t + \Delta t_{\text{FE}})\|_{L^2} \leq \|u(t)\|_{L^2}, \quad (2.14)$$

where $\|u(t)\|_{L^2} := \sqrt{\sum_j u_j^2(t) \Delta x}$.

Theorem 2.5. Assume that the system of ODEs (2.7)–(2.9) is numerically integrated by the forward Euler method from time t to $t + \Delta t_{\text{FE}}$ and that the following CFL condition holds:

$$\Delta t_{\text{FE}} \leq \frac{1}{4m \cdot \max(a, b)} \cdot \frac{1}{\max\{(u_x)_{j,k}^2, (u_y)_{j,k}^2\}}, \quad a := \sum_{p=-m}^m \alpha_p^2, \quad b := \sum_{p=-m}^m \beta_p^2, \quad (2.15)$$

where α_p and β_p are the coefficients in (2.6) and $(u_x)_{j,k}$ and $(u_y)_{j,k}$ are given by (2.9). Then

$$\|u(t + \Delta t_{\text{FE}})\|_{L^2} \leq \|u(t)\|_{L^2}, \quad (2.16)$$

where $\|u(t)\|_{L^2} := \sqrt{\sum_{j,k} u_{j,k}^2(t) \Delta x \Delta y}$.

Theorem 2.6. Assume that the system of ODEs (2.11), (2.12) is numerically integrated by the forward Euler method from time t to $t + \Delta t_{\text{FE}}$ and that the following CFL condition holds:

$$\Delta t_{\text{FE}} \leq \frac{1}{6m \cdot \max(a, b)} \cdot \frac{1}{\max_{j,k} u_{j,k}^2}. \quad (2.17)$$

Then,

$$\|u(t + \Delta t_{\text{FE}})\|_{L^2} \leq \|u(t)\|_{L^2}, \quad (2.18)$$

with the same a and b as in Theorem 2.5.

Proofs of Theorem 2.4, Theorem 2.5 and Theorem 2.6 are provided in Appendix A.

Remark 2.3. We would like to emphasize that the code DUMKA3 automatically selects time-steps so that in average the selected time-steps Δt_{ODE} are much larger than Δt_{FE} .

3. Pseudo-spectral methods for (1.8)

In this section, we describe the (exact) pseudo-spectral solver for equation (1.8) and its 1-D version.

3.1. One-dimensional pseudo-spectral method

We consider the 1-D equation,

$$u_t = -u_{xx} - \delta u_{xxxx}, \quad x \in [0, L], \quad t \in (0, T], \tag{3.1}$$

subject to the L -periodic boundary conditions.

We first use the FFT algorithm to compute the discrete Fourier coefficients $\{\widehat{u}_m(t)\}$ from the available point values $\{u_j(t)\}$. This gives us the following spectral approximation of u on $[0, L]$:

$$u(x, t) \approx \sum_m \widehat{u}_m(t) e^{i \frac{2\pi m x}{L}}. \tag{3.2}$$

We then substitute (3.2) into (3.1) and obtain very simple linear ODEs for the discrete Fourier coefficients of u ,

$$\frac{d}{dt} \widehat{u}_m(t) = (s - \delta s^2) \widehat{u}_m(t), \quad s = \left(\frac{2\pi m}{L}\right)^2,$$

which can be solved exactly:

$$\widehat{u}_m(t + \Delta t) = e^{(s - \delta s^2)\Delta t} \widehat{u}_m(t).$$

Finally, we use the inverse FFT algorithm to obtain the point values of the solution at the new time level, $\{u_j(t + \Delta t)\}$, out of the set of the discrete Fourier coefficients $\{\widehat{u}_m(t + \Delta t)\}$.

3.2. Two-dimensional pseudo-spectral method

We now solve the 2-D equation (1.8),

$$u_t = -(u_{xx} + u_{yy}) - \delta(u_{xxxx} + 2u_{xxyy} + u_{yyyy}),$$

on a rectangular domain $\Omega = [0, L_x] \times [0, L_y]$ with the L_x - and L_y -periodic boundary conditions in the x - and y -directions, respectively.

Similar to the 1-D case, we apply the FFT algorithm and obtain very simple linear ODEs for the discrete Fourier coefficients of u ,

$$\frac{d}{dt} \widehat{u}_{m,\ell}(t) = (s - \delta s^2) \widehat{u}_{m,\ell}(t), \quad s = \left(\frac{2\pi m}{L_x}\right)^2 + \left(\frac{2\pi \ell}{L_y}\right)^2. \tag{3.3}$$

The exact solution of (3.3) is

$$\widehat{u}_{m,\ell}(t + \Delta t) = e^{(s - \delta s^2)\Delta t} \widehat{u}_{m,\ell}(t).$$

Finally, we apply the inverse FFT algorithm to obtain the point values of the solution at the new time level, $\{u_{j,k}(t + \Delta t)\}$, out of the set of the discrete Fourier coefficients $\{\widehat{u}_{m,\ell}(t + \Delta t)\}$.

Remark 3.1. Using Parseval’s theorem and the fact that $e^{(s - \delta s^2)\Delta t} \leq e^{\frac{\Delta t}{4\delta}}$, we obtain the following result on stability of the pseudo-spectral methods:

$$\|u(t + \Delta t)\|_{L^2} \leq e^{\frac{\Delta t}{4\delta}} \|u(t)\|_{L^2},$$

which is true in both the 1-D and 2-D cases.

4. Adaptive splitting time-stepping strategy

For practical applications, the efficiency of splitting methods hinges on its ability to use (relatively) large time-steps (see, e.g., [5–9]). Our numerical experiments indicate that taking $\Delta t = \delta/100$ for the MBE equations and $\Delta t = \delta/10$ for the CH equation leads to accurate results. However, one expects such a small Δt is only required when the phase transition occurs and the solution changes quite rapidly. At other times and especially the solution is close to its steady state, it might be safe to use much larger Δt . We therefore explore an adaptive splitting time-stepping strategy: We would like to use small Δt only whenever necessary.

To design an adaptive approach, we need to measure the solution variation. This can be done using either the energy or solution roughness at time t , which is defined by

$$w(t) = \sqrt{\frac{1}{|\Omega|} \int_{\Omega} [u(x, y, t) - \bar{u}(t)]^2 dx dy}, \tag{4.1}$$

Table 4.1
CPU times for Examples 1–4 in Section 5.

Example	Number of grid points	Final time	Splitting step	CPU time
1	256	240	constant adaptive	3.2805 0.9659
2	256 × 256	30	constant adaptive	4601.9 838.9
3	512 × 512	80,000	constant adaptive	223,370 38,775
4	128 × 128	20	constant adaptive	504.09 125.86

where

$$\bar{u}(t) = \frac{1}{|\Omega|} \int_{\Omega} u(x, y, t) dx dy \quad (4.2)$$

is the mean height at time t .

We adjust the size of splitting steps using the following roughness-dependent monitor function introduced in [21]

$$\Delta t = \max \left(\Delta t_{\min}, \frac{\Delta t_{\max}}{\sqrt{1 + \alpha |w'(t)|^2}} \right), \quad \alpha = \text{Const}. \quad (4.3)$$

Here, Δt_{\min} is the smallest splitting step, which is taken to be either $\Delta t_{\min} = \delta/100$ (for the MBE equations) or $\Delta t_{\min} = \delta/10$ (for the CH equation), Δt_{\max} is the largest allowed splitting step, and α is a positive adaption constant.

Notice that large $|w'(t)|$ will lead to small splitting step, which corresponds to the case of rapid change of roughness or quick motion of the structural transition from one stage to the next one. Similarly, small $|w'(t)|$ yields large splitting step, which corresponds to the slow MBE growth or slow phase interface motion.

Remark 4.1. A similar adaptive strategy can be designed by replacing $w(t)$ with $E(t)$. However, our numerical experiments indicate the roughness-based strategy is more robust than the energy-based one.

Our numerical experiments reported in Section 5 suggest that the adaptive splitting time-stepping strategy can lead to a substantial reduction of the CPU time without significantly affecting the accuracy of the computed solution. The data on the CPU time reduction achieved in different numerical examples are presented in Table 4.1: In average, the adaptive method is about 3–6 times more efficient.

5. Numerical examples

In this section, we illustrate the performance of our fast and stable explicit operator splitting methods on several 1-D and 2-D examples. When solving equation (1.7) and (1.11), we use the fourth-order finite-difference schemes developed in Section 2 (in Example 5, we also use the sixth-order scheme). Both constant and adaptive splitting steps are employed to obtain numerical solutions. The adaptive splitting step is determined by (4.3) with the values Δt_{\min} , Δt_{\max} and α being specified in each example.

To verify the rates of convergence of the proposed methods, we measure the difference between the solutions computed at the same time level on two consecutive grids using the L^1 - and L^∞ -errors, which are defined as follows:

$$\|u^{N, \Delta t_1} - u^{N/2, \Delta t_2}\|_1 := \frac{L_x L_y}{N^2} \sum_{j=1}^N \sum_{k=1}^N |u_{j,k}^{N, \Delta t_1} - u_{j,k}^{N/2, \Delta t_2}|,$$

and

$$\|u^{N, \Delta t_1} - u^{N/2, \Delta t_2}\|_\infty := \max_{1 \leq j, k \leq N} |u_{j,k}^{N, \Delta t_1} - u_{j,k}^{N/2, \Delta t_2}|,$$

where $u^{N, \Delta t} := \{u_{j,k}^N\}$ is a numerical solution obtained with a uniform $N \times N$ grid and a constant splitting step Δt at some time level. Then, to measure the experimental convergence rates, we use the ratio of errors:

$$r = \log_2 \left(\frac{\|u^{N/2, \Delta t_2} - u^{N/4, \Delta t_3}\|}{\|u^{N, \Delta t_1} - u^{N/2, \Delta t_2}\|} \right),$$

where we either take $\Delta t_1 = \Delta t$, $\Delta t_2 = 2\Delta t$ and $\Delta t_3 = 4\Delta t$ or fix the splitting step and set $\Delta t_1 = \Delta t_2 = \Delta t_3 = \Delta t$.

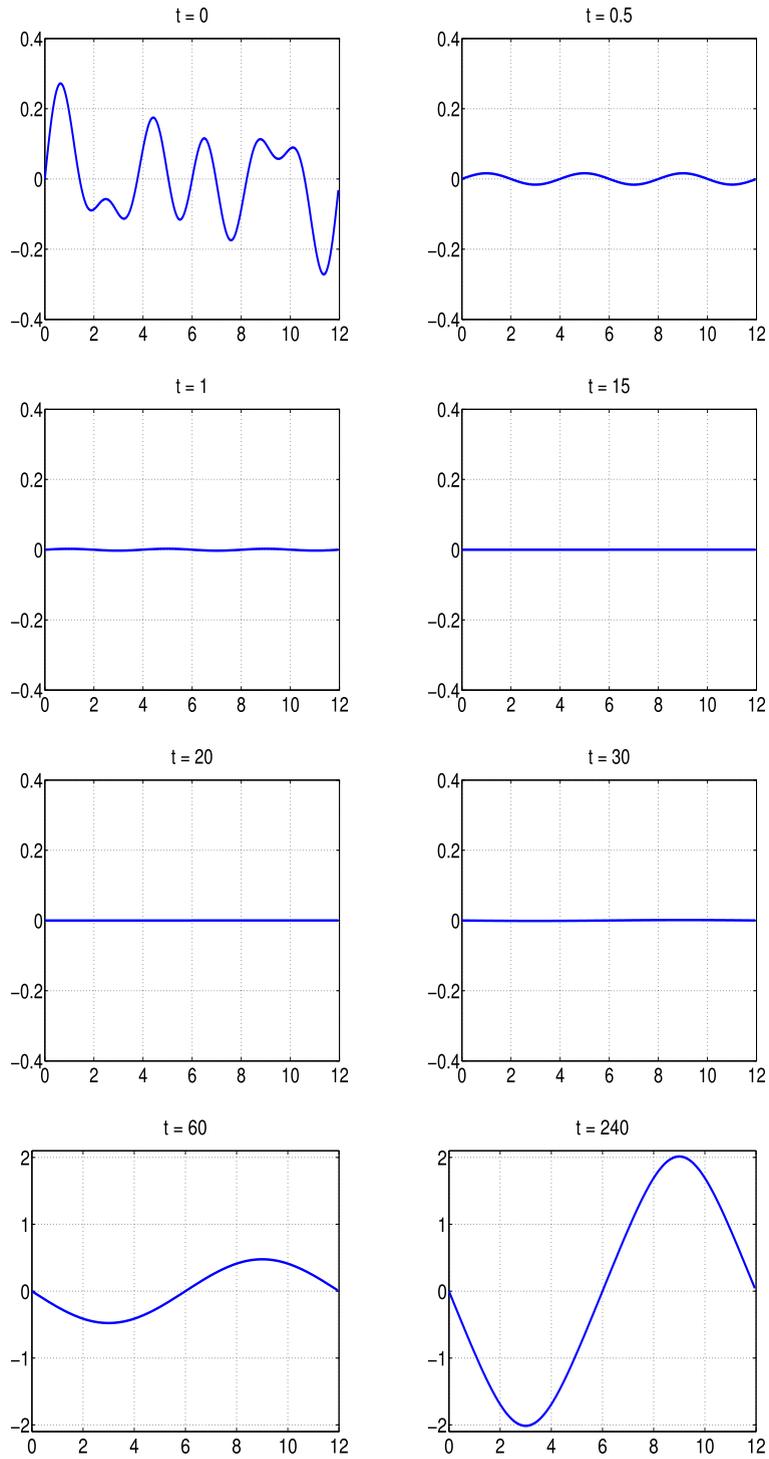


Fig. 5.1. Example 1: u computed with $\Delta t = 10^{-1}$.

In the 1-D case, the rates are computed similarly.

Example 1 (One-dimensional morphological instability). We first consider the 1-D MBE equation

$$u_t = (u_x^3)_x - u_{xx} - u_{xxxx},$$

subject to the initial condition

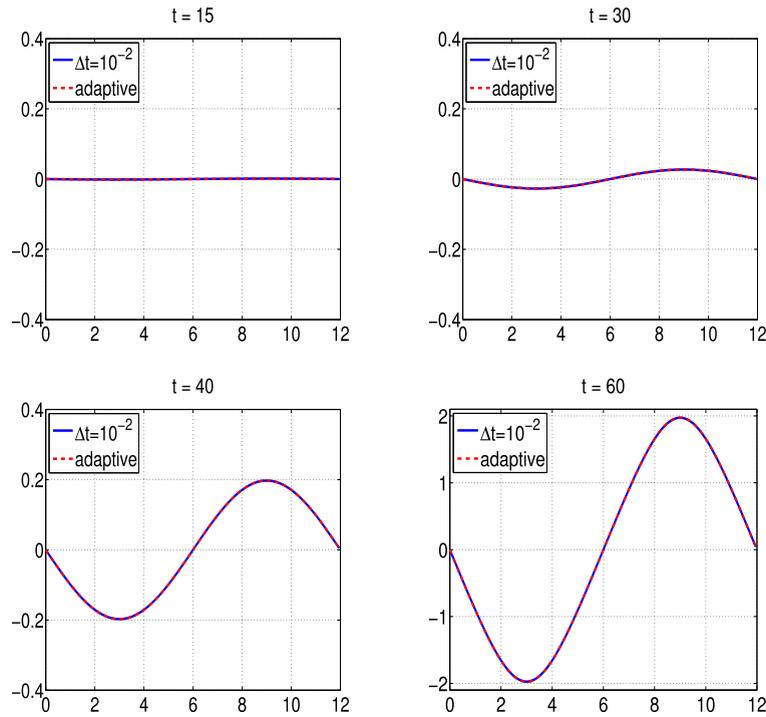


Fig. 5.2. Example 1: u computed with $\Delta t = 10^{-2}$ (solid line) and adaptive splitting time-stepping with $\Delta t_{\min} = 10^{-2}$, $\Delta t_{\max} = 10^{-1}$ and $\alpha = 10^3$ (dashed line).

$$u(x, 0) = 0.1 \left(\sin \frac{\pi x}{2} + \sin \frac{2\pi x}{3} + \sin \pi x \right), \quad x \in [0, 12].$$

This example was studied in [15] to observe the morphological instability due to the nonlinear interaction.

We compute the solution until the final time $t = 240$ with a constant splitting step $\Delta t = 10^{-1}$ on the uniform grid with $N = 256$. Fig. 5.1 shows a sequence of snapshots of the surface height at different times. As one can observe, the initial oscillation is damped by $t = 1$. After a relatively long period of “buffering” time, a new larger scale structure emerges, then it increases and finally the steady state is reached by $t = 240$.

Compared to the results reported in [15], our steady state is in a good agreement with the one obtained there, while the “buffering” time evolution is very different. We therefore reduce the splitting step by a factor of 10 and repeat the computation with $\Delta t = 10^{-2}$. The obtained solution, plotted in Fig. 5.2 (solid line), now matches the results in [15]: The structure emerges earlier and the steady state is reached by $t = 60$.

The time evolution process can be monitored by plotting the energy (1.5) and roughness (4.1), see Fig. 5.3. One can observe that initially both energy and roughness decay rapidly. However, after a relatively long period of time, roughness starts to grow, which is exactly the morphological instability in the rough–smooth–rough pattern. Notice that the flat tail in Fig. 5.3(b) and (d) indicates that the steady state is reached much later when $\Delta t = 10^{-1}$ is used.

To improve the efficiency of the proposed fast and stable explicit operator splitting methods, we implement the adaptive strategy described in Section 4. Here, we use $\Delta t_{\min} = 10^{-2}$, $\Delta t_{\max} = 10^{-1}$ and $\alpha = 10^3$. The obtained solution is shown in Fig. 5.2 (dashed line), and the corresponding energy and roughness are plotted in Fig. 5.3 (dashed line). As one can see, the adaptive solution practically coincides with the solution computed with $\Delta t = 10^{-2}$. It is instructive to check what splitting steps are used by the adaptive algorithm. To this end, we plot the splitting steps as a function of time in Fig. 5.4. As one can see, the splitting steps are smaller than 10^{-1} only initially and then at the intermediate times. We also compare the CPU times of the adaptive and constant (with $\Delta t = 10^{-2}$) splitting step computations. The results, shown in the first row of Table 4.1, indicate that the CPU time for the adaptive method is about four times smaller than the one for the constant splitting step.

Finally, we test the accuracy of the proposed fast and stable explicit splitting methods. To this end, we perform the mesh-refinement study and measure the L^1 - and L^∞ -errors. The results reported in Table 5.1 indicate that the experimental convergence rate is close to the expected second-order one. We next fix the splitting step to be very small ($\Delta t = 10^{-3}$) so that the splitting errors do not dominate and perform another mesh-refinement study. The results reported in Table 5.2 show that in this regime, the experimental convergence rate is four, which is the order of finite-difference scheme from Section 2.1.

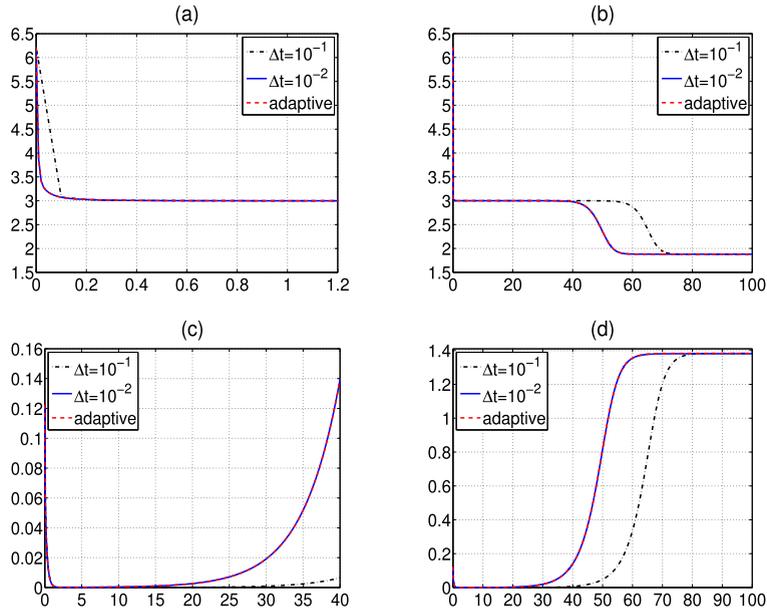


Fig. 5.3. Example 1: (a) Energy evolution in a short time period; (b) Energy evolution in a long time period; (c) Roughness development in a time period $t \in [0, 40]$; (d) Roughness development in a long time period. $\Delta t = 10^{-1}$ (dashed dotted line), $\Delta t = 10^{-2}$ (solid line) and adaptive splitting time-stepping with $\Delta t_{\min} = 10^{-2}$, $\Delta t_{\max} = 10^{-1}$ and $\alpha = 10^3$ (dashed line).

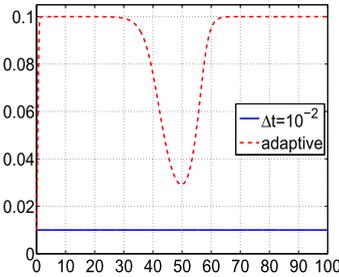


Fig. 5.4. Example 1: Splitting step evolution. $\Delta t = 10^{-2}$ (solid line) and adaptive splitting time-stepping with $\Delta t_{\min} = 10^{-2}$, $\Delta t_{\max} = 10^{-1}$ and $\alpha = 10^3$ (dashed line).

Table 5.1

Example 1: L^1 - and L^∞ -errors and experimental convergence rates at $t = 240$.

N	Δt	$\ u^{N,\Delta t} - u^{N/2,2\Delta t}\ _1$	Rate	$\ u^{N,\Delta t} - u^{N/2,2\Delta t}\ _\infty$	Rate
128	$2e-2$	$3.95e-03$	–	$7.58e-04$	–
256	$1e-2$	$1.07e-03$	1.89	$2.45e-04$	1.63
512	$5e-3$	$2.73e-04$	1.97	$7.17e-05$	1.78
1024	$2.5e-3$	$6.84e-05$	1.99	$1.93e-05$	1.89

Table 5.2

Example 1: L^1 - and L^∞ -errors and experimental convergence rates obtained with the fixed small splitting step $\Delta t = 10^{-3}$ at $t = 240$.

N	Δt	$\ u^{N,\Delta t} - u^{N/2,\Delta t}\ _1$	Rate	$\ u^{N,\Delta t} - u^{N/2,\Delta t}\ _\infty$	Rate
128	$1e-3$	$8.06e-05$	–	$2.25e-05$	–
256	$1e-3$	$5.18e-06$	3.96	$1.44e-06$	3.96
512	$1e-3$	$3.27e-07$	3.99	$9.10e-08$	3.99
1024	$1e-3$	$2.02e-08$	4.02	$5.62e-09$	4.02

Example 2 (Two-dimensional morphological instability). Next, we consider the 2-D MBE equation (1.3) with $\delta = 0.1$ subject to the following initial condition:

$$u(x, y, 0) = 0.1(\sin 3x \sin 2y + \sin 5x \sin 5y), \quad (x, y) \in [0, 2\pi] \times [0, 2\pi].$$

This example was studied in [15,30] to observe the morphological instability due to the nonlinear interaction.

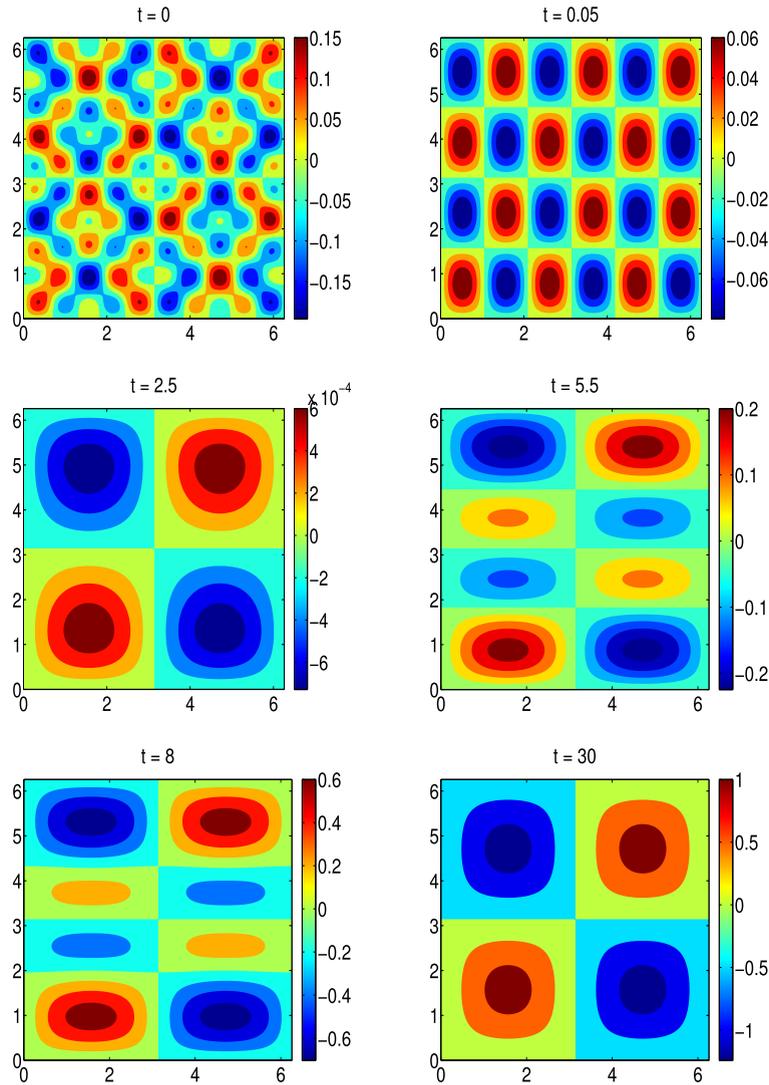


Fig. 5.5. Example 2: u computed with $\Delta t = 10^{-3}$.

We first compute the solution on a 256×256 uniform grid with the constant splitting step $\Delta t = 10^{-3}$. Fig. 5.5 shows the height profiles at times $t = 0, 0.5, 2.5, 5.5, 8$ and 30 . The corresponding gradients $|\nabla u|$ are plotted in Fig. 5.6. In Fig. 5.7, we demonstrate the experimental energy decay and roughness development, which indicate that the solution reaches a steady state at around $t = 12$. The obtained results are in good agreement with those reported in [15].

We then carry out the adaptive strategy to increase the efficiency of the proposed methods. Here, we choose $\Delta t_{\min} = 10^{-3}$, $\Delta t_{\max} = 10^{-2}$ and $\alpha = 10^3$. The corresponding energy and roughness curves in Fig. 5.7 are practically indistinguishable from those obtained using the small constant splitting step $\Delta t = 10^{-3}$. Splitting steps evolution, plotted in Fig. 5.8, shows that $\Delta t \approx \Delta t_{\max}$ when the solution approaches its steady state. This leads to a substantial decrease in CPU time, see the second row in Table 4.1.

Finally, we perform the mesh-refinement study and verify that the experimental convergence rates for the proposed fast and stable explicit operator splitting methods are close to the expected second-order one, see Table 5.3.

Example 3 (Coarsening dynamics). In this example, we study the 2-D MBE equation (1.3) with $\delta = 1$ subject to initial data, obtained by assigning a uniformly distributed random number in the range $[-0.001, 0.001]$ to each grid point value of $u(x, y, 0)$. We use a 512×512 uniform grid on the computational domain $\Omega = [0, 1000] \times [0, 1000]$.

Fig. 5.9 shows the contour lines of the free energy function

$$F_{\text{free}} := \frac{1}{4} (|\nabla u| - 1)^2 + \frac{\delta}{2} |\Delta u|^2$$

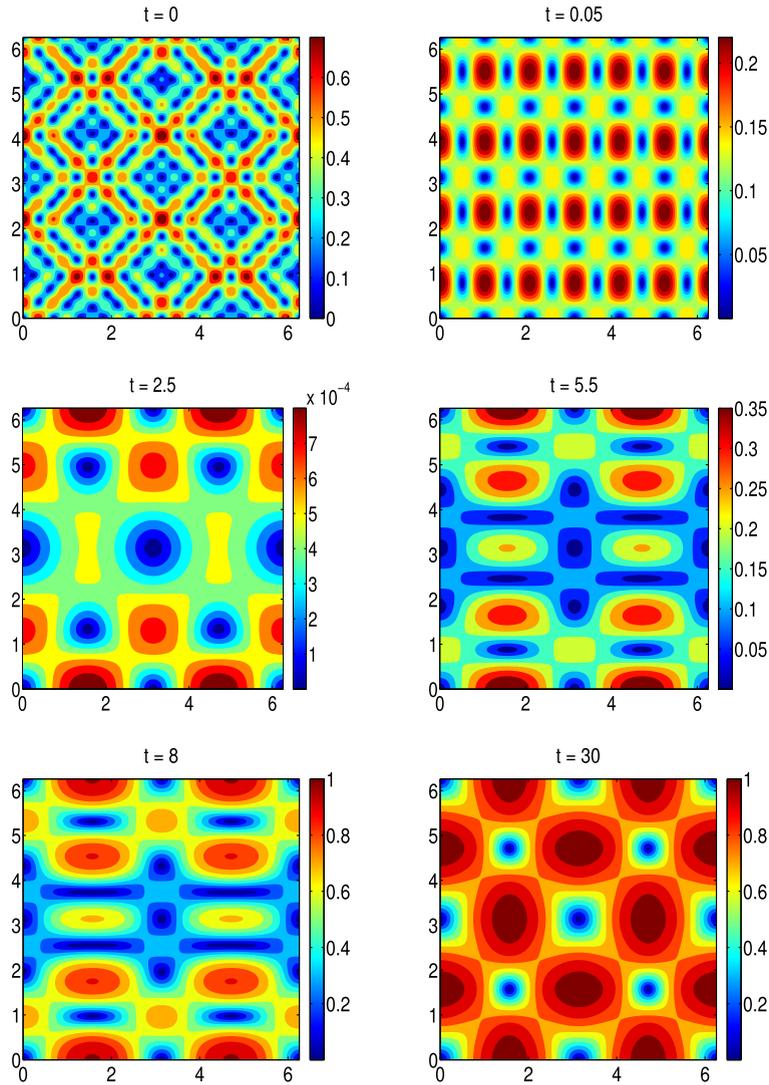


Fig. 5.6. Example 2: $|\nabla u|$ computed with $\Delta t = 10^{-3}$.

at $t = 40,000$ and $80,000$ computed using the constant splitting step $\Delta t = 10^{-1}$. As one can see, the free energy is concentrated on and thus could be used to identify the edges of the pyramidal structures; the pyramid edges form a random network over the surface, which results from the isotropic nature of the surface symmetry; the cells of the network grow in time via a coarsening process.

The energy (1.5), normalized by the domain size, and the roughness (4.1) are plotted in Fig. 5.10. To further demonstrate the robustness of the proposed methods, we experimentally verify several important properties of the computed solution. In Fig. 5.11(a), the energy (1.5), normalized by the domain size, is plotted in the log-log scale and it is nearly parallel to the dashed line representing the $t^{-1/3}$ curve. This suggests that the energy decays in time as a power law Ct^n with the exponent $n = -1/3$. In Fig. 5.11(b), the interface height, defined by

$$\tilde{u}(t) = \left(\frac{1}{|\Omega|} \int_{\Omega} u^2(x, y, t) dx dy \right)^{\frac{1}{2}},$$

and the roughness (4.1) are demonstrated to evolve in time according to a power law Ct^n with the exponent $n = 1/3$. Finally, Fig. 5.11(c) suggests that the difference $\bar{u}(t) - \bar{u}(0)$, where the mean height $\bar{u}(t)$ is defined in (4.2), remains practically zero at all times, which implies the mass conservation. The obtained results match the experimental and numerical results reported in [15,20,30].

When the adaptive technique with $\Delta t_{\min} = 10^{-1}$, $\Delta t_{\max} = 5$ and $\alpha = 1$ is implemented, the obtained results are very similar. As one can see in Fig. 5.10, the splitting step increases to Δt_{\max} very soon and then is always selected close to

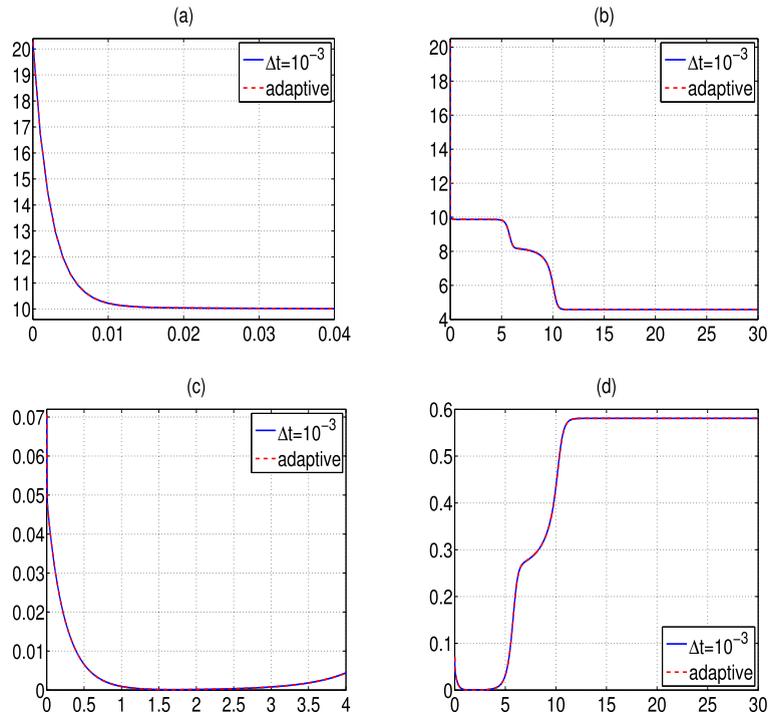


Fig. 5.7. Example 2: (a) Energy evolution in a short time period; (b) Energy evolution in a long time period; (c) Roughness development in a short time period; (d) Roughness development in a long time period. $\Delta t = 10^{-3}$ (solid line) and adaptive splitting time-stepping with $\Delta t_{\min} = 10^{-3}$, $\Delta t_{\max} = 10^{-2}$ and $\alpha = 10^3$ (dashed line).

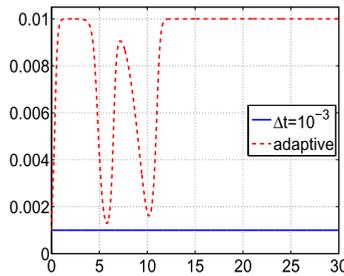


Fig. 5.8. Example 2: Splitting step evolution. $\Delta t = 10^{-3}$ (solid line) and adaptive splitting time-stepping with $\Delta t_{\min} = 10^{-3}$, $\Delta t_{\max} = 10^{-2}$ and $\alpha = 10^3$ (dashed line).

Table 5.3

Example 2: L^1 - and L^∞ -errors and experimental convergence rates at $t = 30$.

N	Δt	$\ u^{N,\Delta t} - u^{N/2,2\Delta t}\ _1$	Rate	$\ u^{N,\Delta t} - u^{N/2,2\Delta t}\ _\infty$	Rate
64	$4e-3$	$3.36e-03$	–	$6.01e-04$	–
128	$2e-3$	$9.09e-04$	1.88	$1.55e-04$	1.96
256	$1e-3$	$2.48e-04$	1.87	$4.96e-05$	1.64
512	$5e-4$	$6.52e-05$	1.93	$1.55e-05$	1.68

Δt_{\max} due to the slow variation of the roughness. This leads to substantial CPU time usage saving, see the third row in Table 4.1.

Remark 5.1. We would like to point out that in this example, the curves for the interface height growth and the roughness development in Fig. 5.11(b) almost overlap since $\bar{u}(t) = \bar{u}(0) \approx 0$.

Remark 5.2. We would like to stress that in this example, the energy transition does not occur, and the solution has a smooth variation. It is therefore safe to take a relatively small value $\alpha = 1$ and $\Delta t_{\min} = 10^{-1}$, which is much larger than $\delta/100 = 10^{-2}$.

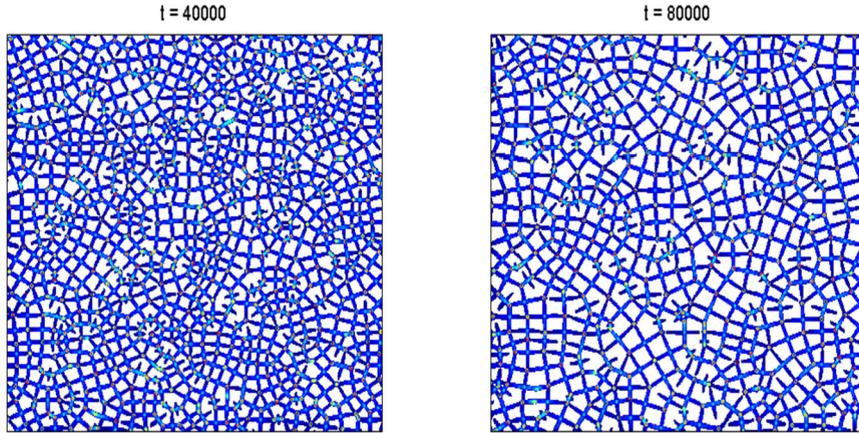


Fig. 5.9. Example 3: Contour plots of F_{free} computed with $\Delta t = 10^{-1}$.

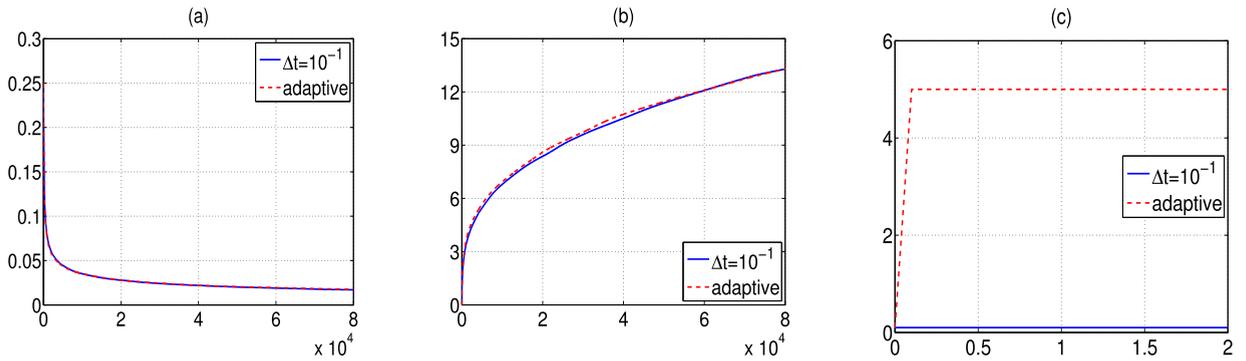


Fig. 5.10. Example 3: (a) Energy evolution; (b) Roughness development; (c) Splitting step evolution. $\Delta t = 10^{-1}$ (solid line) and adaptive splitting time-stepping with $\Delta t_{min} = 10^{-1}$, $\Delta t_{max} = 5$ and $\alpha = 1$ (dashed line).

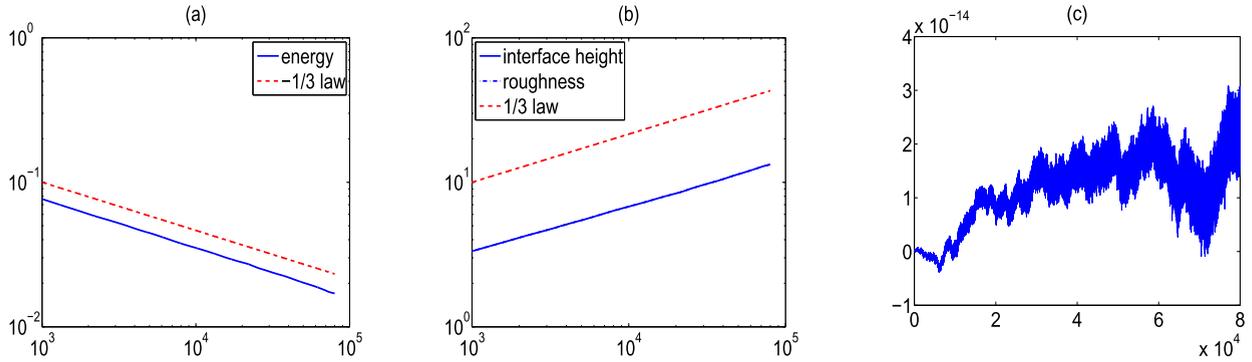


Fig. 5.11. Example 3: (a) The log-log scale plot of the energy evolution; (b) The log-log scale plot of the interface height and roughness development; (c) Mean height evolution. In (a)–(c), $\Delta t = 10^{-1}$.

Example 4 (Non-mean-zero initial data). In this example, taken from [10], we consider the 2-D CH equation (1.4) with $\delta = 0.01$ subject to the following non-mean-zero initial condition:

$$u(x, y, 0) = 0.05 \sin x \sin y + 0.001, \quad (x, y) \in [0, 2\pi] \times [0, 2\pi].$$

We first compute the solution on a 128×128 uniform grid with the constant splitting step $\Delta t = 10^{-3}$. The solution computed at times $t = 1, 2, 5$ and 20 is shown in Fig. 5.12 (left). The experimental energy decay and roughness development curves, shown in Fig. 5.13, indicate that the solution reaches a steady state at about $t = 9$. These results are in good agreement with those reported in [10].

We then compute the solution using the adaptive strategy with $\Delta t_{min} = 10^{-3}$, $\Delta t_{max} = 10^{-2}$ and $\alpha = 10^2$. The results are plotted in Fig. 5.12 (right). As one can see, the solution dynamics can be captured correctly when the adaptive strategy

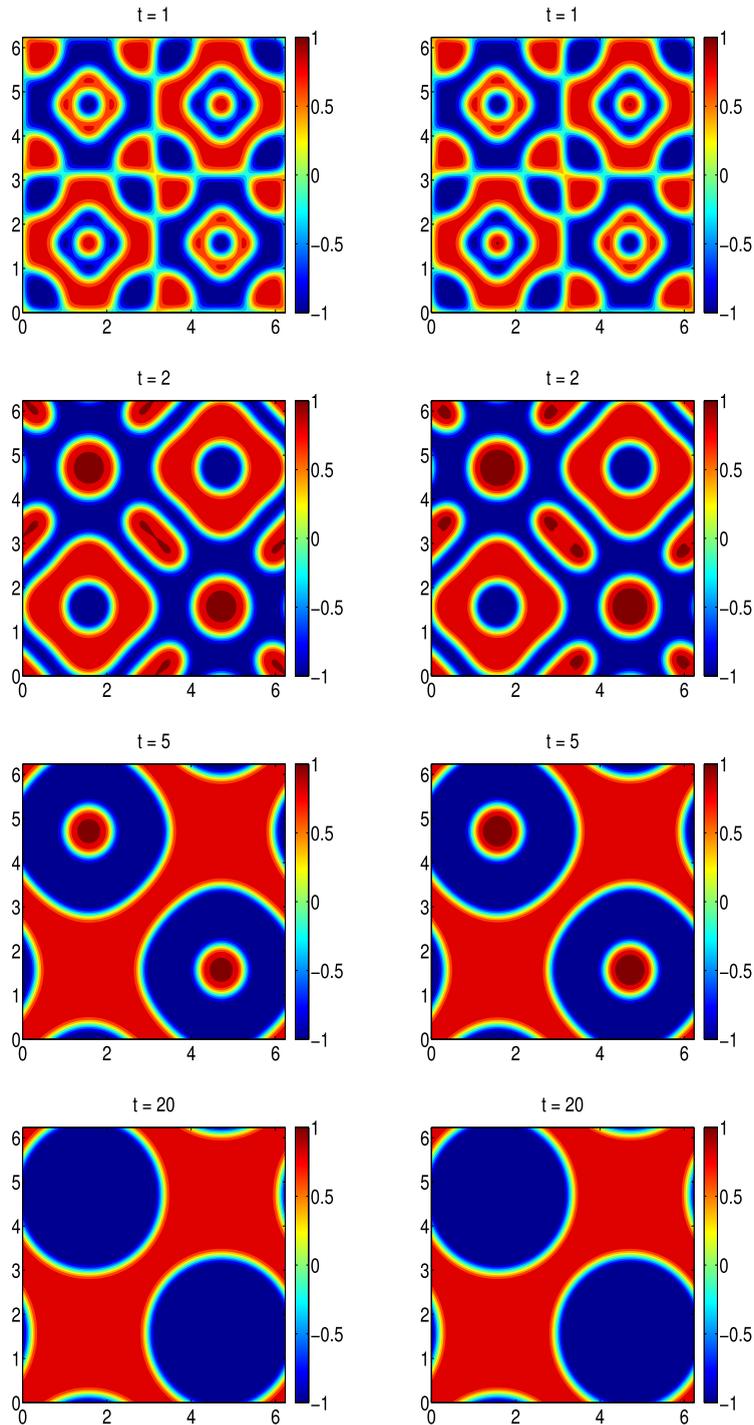


Fig. 5.12. Example 4: u computed with $\Delta t = 10^{-3}$ (left column) and adaptive splitting time-stepping with $\Delta t_{\min} = 10^{-3}$, $\Delta t_{\max} = 10^{-2}$ and $\alpha = 10^2$ (right column).

is employed. The corresponding energy and roughness curves shown in Fig. 5.13 have some discrepancy with those obtained using the small constant splitting step Δt^{-3} , though the adaptive and non-adaptive solutions are quite close and the resulting steady states seem to be the same. Splitting steps evolution, also plotted in Fig. 5.13, shows that $\Delta t \approx \Delta t_{\max}$ when the solution approaches its steady state, which leads to a substantial saving in CPU time, see the fourth row in Table 4.1.

Finally, we perform the mesh-refinement study and verify that the experimental convergence rates for the proposed fast and stable explicit operator splitting methods are close to the expected second-order one, see Table 5.4.

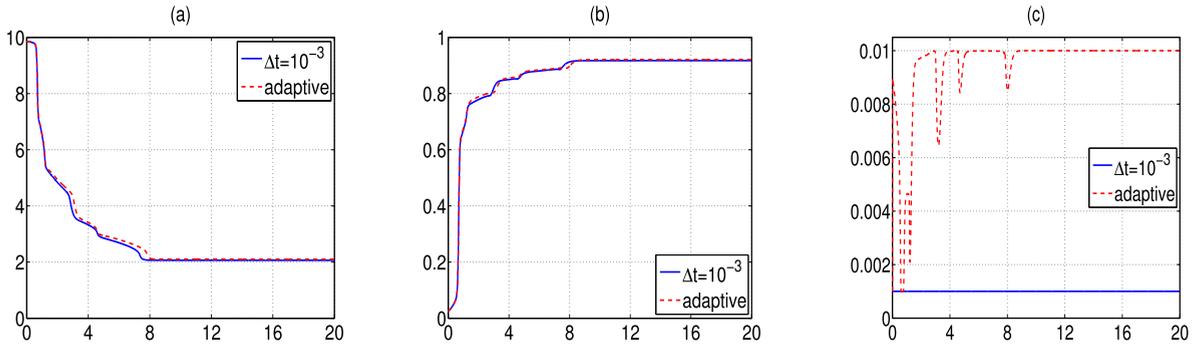


Fig. 5.13. Example 4: (a) Energy evolution; (b) Roughness development; (c) Splitting step evolution. $\Delta t = 10^{-3}$ (solid line) and adaptive splitting time-stepping with $\Delta t_{\min} = 10^{-3}$, $\Delta t_{\max} = 10^{-2}$ and $\alpha = 10^2$ (dashed line).

Table 5.4

Example 4: L^1 - and L^∞ -errors and experimental convergence rates at $t = 20$.

N	Δt	$\ u^{N,\Delta t} - u^{N/2,2\Delta t}\ _1$	Rate	$\ u^{N,\Delta t} - u^{N/2,2\Delta t}\ _\infty$	Rate
64	$2e-3$	$1.61e-00$	–	$3.01e-01$	–
128	$1e-3$	$1.37e-01$	3.55	$2.64e-02$	3.51
256	$5e-4$	$3.44e-02$	2.00	$5.79e-03$	2.19
512	$2.5e-4$	$1.12e-02$	1.62	$1.72e-03$	1.75

Example 5 (Mean-zero initial data). In this example, also taken from [10], we consider the 2-D CH equation with $\delta = 0.01$ subject to the following mean-zero initial condition:

$$u(x, y, 0) = 0.05 \sin x \sin y, \quad (x, y) \in [0, 2\pi] \times [0, 2\pi].$$

We first use a uniform 128×128 grid with a constant splitting step $\Delta t = 10^{-3}$ and compute the solution until a large final time $t = 100$. The obtained results are shown in Fig. 5.14 (left column). Even though the solution at a small time $t = 2$ is similar to the corresponding solution reported in [31], later on our solution bifurcates and seems to converge to a different steady state. We therefore perform a thorough comparative study by taking a smaller $\Delta t = 10^{-4}$ and finer 256×256 grid. The results, plotted in Fig. 5.14, clearly indicate that different numerical solutions may converge to different steady states. We then use the sixth-order semi-discrete finite-difference scheme for equation (1.11) instead of the fourth-order one and discover even more different steady-state patterns, see Fig. 5.15.

Our results suggest that the mean-zero solutions of the 2-D CH equation (1.4) may be unstable. Our conjecture is supported by recent analytical results on unstable equilibria in the 1-D CH equation, see [22].

Acknowledgements

The work of Y. Cheng, A. Kurganov and Z. Qu was supported in part by the NSF Grant # DMS-1115718. The research of T. Tang was supported in part by Hong Kong Research Grants Council CERG grants, the National Natural Science Foundation of China, and Hong Kong Baptist University FRG grants.

Appendix A. Proofs of Theorems 2.4–2.6

Here, we provide proofs of Theorems 2.4–2.6. We denote $t^{\text{new}} := t + \Delta t_{\text{FE}}$. We also use the following notations: $u_j^{\text{new}} := u(x_j, t + \Delta t_{\text{FE}})$ (in Theorem 2.4) and $u_{j,k}^{\text{new}} := u(x_j, y_k, t + \Delta t_{\text{FE}})$ (in Theorems 2.5 and 2.6).

A.1. Proof of Theorem 2.4 (1-D MBE equation)

Applying the forward Euler method to discretize (2.3) results in

$$\frac{u_j^{\text{new}} - u_j}{\Delta t} = F_j. \tag{A.1}$$

We first multiply both sides of equation (A.1) by $(u_j^{\text{new}} + u_j)/2$, replace u_j^{new} on the right using (A.1) and sum over the entire domain to obtain

$$\sum_j \frac{(u_j^{\text{new}})^2 - u_j^2}{2\Delta t} = \sum_j F_j u_j + \frac{\Delta t}{2} \sum_j F_j^2. \tag{A.2}$$

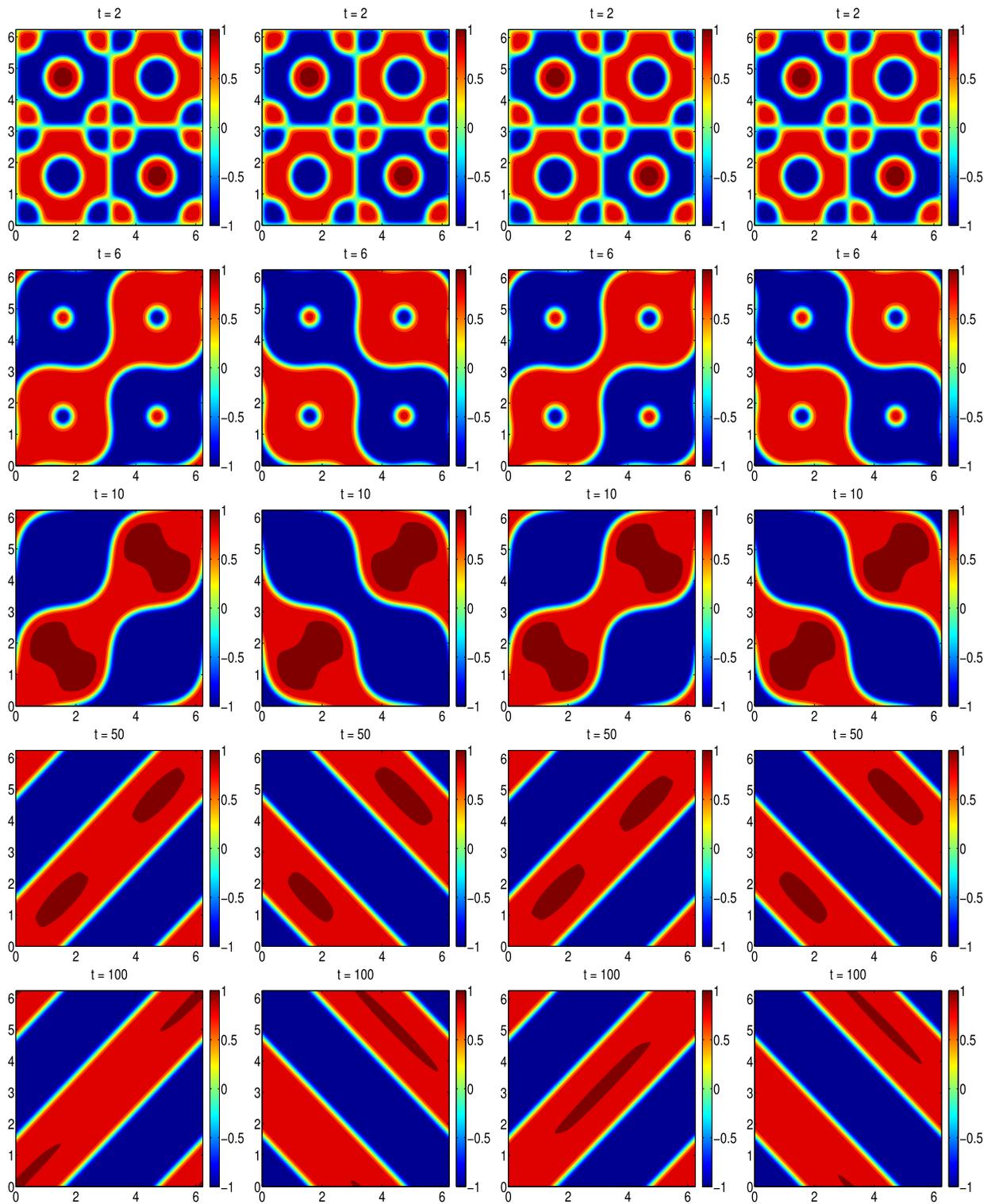


Fig. 5.14. Example 5: u computed with a 128×128 grid, $\Delta t = 10^{-3}$ (first column); 128×128 grid, $\Delta t = 10^{-4}$ (second column); 256×256 grid, $\Delta t = 10^{-3}$ (third column); 256×256 grid, $\Delta t = 10^{-4}$ (fourth column).

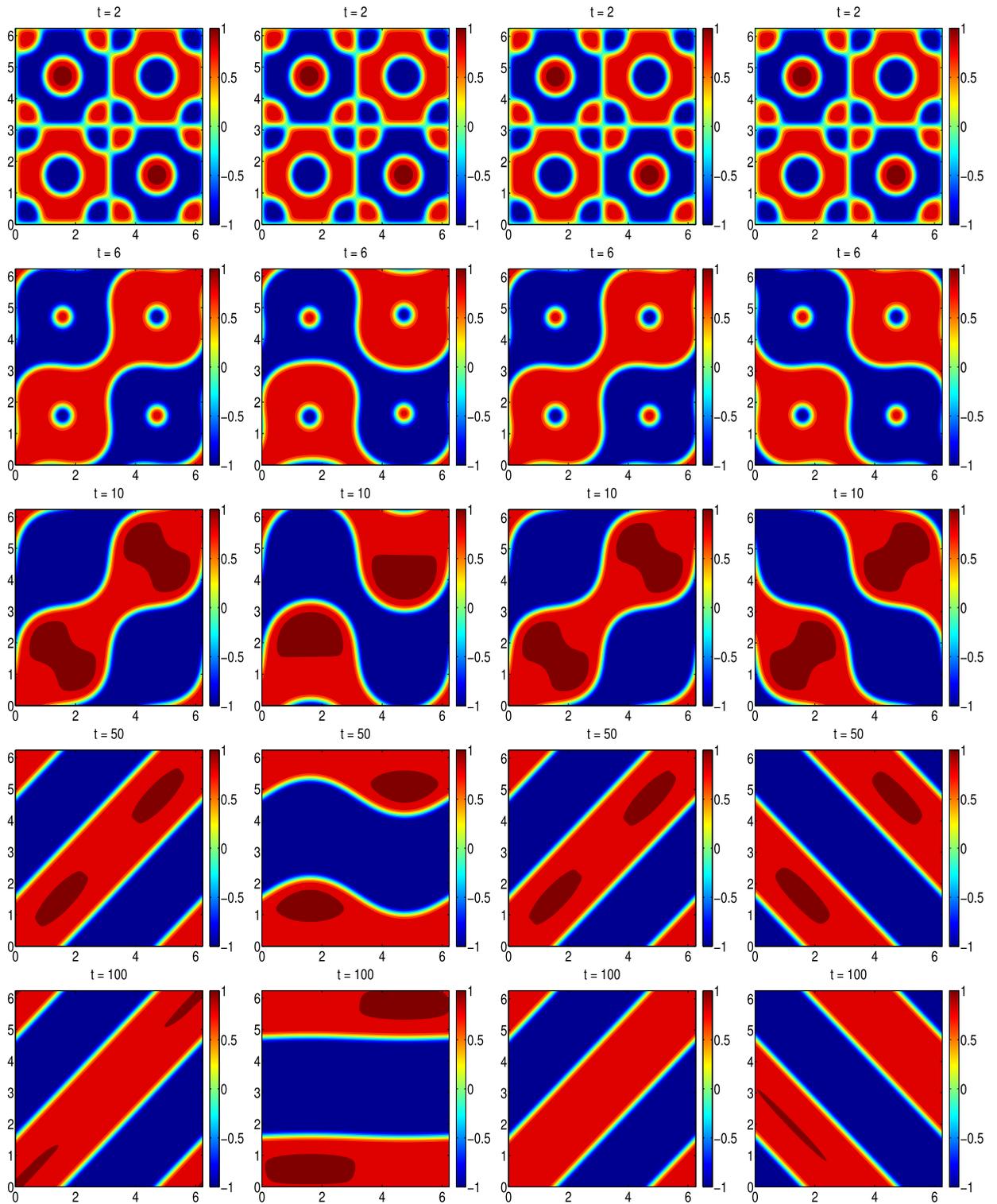


Fig. 5.15. Example 5: Same as in Fig. 5.14, but using the sixth-order scheme for (1.11).

We then rewrite the first term on the right-hand side (RHS) of (A.2) as follows:

$$\begin{aligned} \sum_j F_j u_j &\stackrel{(2.3)}{=} \sum_j \sum_{p=-m}^m \alpha_p H_{j+p} u_j = \sum_{p=-m}^m \alpha_p \sum_j H_{j+p} u_j = \sum_{p=-m}^m \alpha_p \sum_j H_j u_{j-p} \\ &= \sum_j H_j \sum_{p=-m}^m \alpha_p u_{j-p} = \sum_j H_j \sum_{p=-m}^m \alpha_{-p} u_{j+p} \stackrel{(2.5)}{=} \sum_j H_j \sum_{p=-m}^m (-\alpha_p) u_{j+p} \stackrel{(2.4)}{=} - \sum_j (u_x)_j^4, \end{aligned}$$

and estimate the second term on the RHS of (A.2) using the Cauchy–Schwarz inequality:

$$\begin{aligned} \sum_j F_j^2 &\stackrel{(2.3)}{=} \sum_j \left(\sum_{p=-m}^m \alpha_p H_{j+p} \right)^2 \leq 2m \sum_j \sum_{p=-m}^m \alpha_p^2 H_{j+p}^2 \\ &= 2m \sum_{p=-m}^m \alpha_p^2 \sum_j H_{j+p}^2 = 2am \sum_j H_j^2 \stackrel{(2.4)}{=} 2am \sum_j (u_x)_j^6. \end{aligned}$$

Therefore, the left-hand side (LHS) of equation (A.2) can be bounded by

$$\sum_j \frac{(u_j^{\text{new}})^2 - u_j^2}{2\Delta t} \leq - \sum_j (u_x)_j^4 + am\Delta t \sum_j (u_x)_j^6 \leq \left[am\Delta t \max_j (u_x)_j^2 - 1 \right] \sum_j (u_x)_j^4,$$

which is nonpositive provided the time-step is bounded by (2.13). □

A.2. Proof of Theorem 2.5 (2-D MBE equation)

Applying the forward Euler method to discretize (2.7) results in

$$\frac{u_{j,k}^{\text{new}} - u_{j,k}}{\Delta t} = F_{j,k}. \tag{A.3}$$

We first multiply both sides of equation (A.3) by $(u_{j,k}^{\text{new}} + u_{j,k})/2$, replace $u_{j,k}^{\text{new}}$ on the right using (A.3) and sum over the entire domain to obtain

$$\sum_{j,k} \frac{(u_{j,k}^{\text{new}})^2 - u_{j,k}^2}{2\Delta t} = \sum_{j,k} F_{j,k} u_{j,k} + \frac{\Delta t}{2} \sum_{j,k} F_{j,k}^2. \tag{A.4}$$

We then use (2.7)–(2.9) to rewrite the first term on the RHS of (A.4) as follows:

$$\sum_{j,k} F_{j,k} u_{j,k} = - \sum_{j,k} \left[(u_x)_{j,k}^4 + (u_x)_{j,k}^2 (u_y)_{j,k}^2 \right] - \sum_{j,k} \left[(u_y)_{j,k}^4 + (u_y)_{j,k}^2 (u_x)_{j,k}^2 \right],$$

and use (2.7), (2.8) and the Cauchy–Schwarz inequality to estimate the second term on the RHS of (A.4):

$$\sum_{j,k} F_{j,k}^2 \leq 8am \sum_{j,k} \left[(u_x)_{j,k}^6 + (u_x)_{j,k}^2 (u_y)_{j,k}^4 \right] + 8bm \sum_{j,k} \left[(u_y)_{j,k}^6 + (u_y)_{j,k}^2 (u_x)_{j,k}^4 \right].$$

Therefore, the LHS of equation (A.4) can be bounded by

$$\begin{aligned} \sum_{j,k} \frac{(u_{j,k}^{\text{new}})^2 - u_{j,k}^2}{2\Delta t} &\leq \left[4am\Delta t \max_{j,k} (u_x)_{j,k}^2 - 1 \right] \sum_{j,k} (u_x)_{j,k}^4 + \left[4bm\Delta t \max_{j,k} (u_y)_{j,k}^2 - 1 \right] \sum_{j,k} (u_y)_{j,k}^4 \\ &\quad + \left[4am\Delta t \max_{j,k} (u_y)_{j,k}^2 - 1 \right] \sum_{j,k} (u_x)_{j,k}^2 (u_y)_{j,k}^2 + \left[4bm\Delta t \max_{j,k} (u_x)_{j,k}^2 - 1 \right] \sum_{j,k} (u_y)_{j,k}^2 (u_x)_{j,k}^2, \end{aligned}$$

which is nonpositive provided the time-step is bounded by (2.15). □

A.3. Proof of Theorem 2.6 (2-D CH equation)

Applying the forward Euler method to discretize (2.11) results in

$$\frac{u_{j,k}^{\text{new}} - u_{j,k}}{\Delta t} = F_{j,k}. \tag{A.5}$$

Once again, we multiply both sides of equation (A.5) by $(u_{j,k}^{\text{new}} + u_{j,k})/2$, replace $u_{j,k}^{\text{new}}$ on the right using (A.5) and sum over the entire domain to obtain

$$\sum_{j,k} \frac{(u_{j,k}^{\text{new}})^2 - u_{j,k}^2}{2\Delta t} = \sum_{j,k} F_{j,k} u_{j,k} + \frac{\Delta t}{2} \sum_{j,k} F_{j,k}^2. \quad (\text{A.6})$$

We now notice that within the accuracy of the scheme

$$H_{j,k}^x = 3u_{j,k}^2(u_x)_{j,k} \quad \text{and} \quad H_{j,k}^y = 3u_{j,k}^2(u_y)_{j,k}. \quad (\text{A.7})$$

We then use (A.7), (2.10) and (2.11) to rewrite the first term on the RHS of (A.6) as follows:

$$\sum_{j,k} F_{j,k} u_{j,k} = -3 \sum_{j,k} u_{j,k}^2 (u_x)_{j,k}^2 - 3 \sum_{j,k} u_{j,k}^2 (u_y)_{j,k}^2,$$

and use (A.7), (2.11) and the Cauchy–Schwarz inequality to estimate the second term on the RHS of (A.6):

$$\sum_{j,k} F_{j,k}^2 \leq 36am \sum_{j,k} u_{j,k}^4 (u_x)_{j,k}^2 + 36bm \sum_{j,k} u_{j,k}^4 (u_y)_{j,k}^2.$$

Therefore, the LHS of equation (A.6) can be bounded by

$$\sum_{j,k} \frac{(u_{j,k}^{\text{new}})^2 - u_{j,k}^2}{2\Delta t} \leq 3 \left[6am\Delta t \max_{j,k} (u_{j,k}^2) - 1 \right] \sum_{j,k} u_{j,k}^2 (u_x)_{j,k}^2 + 3 \left[6bm\Delta t \max_{j,k} (u_{j,k}^2) - 1 \right] \sum_{j,k} u_{j,k}^2 (u_y)_{j,k}^2,$$

which is nonnegative provided the time-step is bounded by (2.17). \square

References

- [1] A. Abdulle, ROCK4 code, available at <http://www.unige.ch/~hairer/software.html>.
- [2] A. Abdulle, Fourth order Chebyshev methods with recurrence relation, *SIAM J. Sci. Comput.* 23 (2002) 2041–2054 (electronic).
- [3] A. Abdulle, A.A. Medovikov, Second order Chebyshev methods based on orthogonal polynomials, *Numer. Math.* 90 (2001) 1–18.
- [4] J.W. Cahn, J.E. Hilliard, Free energy of a nonuniform system. I. Interfacial free energy, *J. Chem. Phys.* 28 (1958) 258–267.
- [5] A. Chertock, C. Doering, E. Kashdan, A. Kurganov, A fast explicit operator splitting method for passive scalar advection, *J. Sci. Comput.* 45 (2010) 200–214.
- [6] A. Chertock, E. Kashdan, A. Kurganov, Propagation of diffusing pollutant by a hybrid Eulerian–Lagrangian method, in: S. Benzeni-Gavage, D. Serre (Eds.), *Hyperbolic Problems: Theory, Numerics, Applications*, Lyon, 2006, Springer, 2008, pp. 371–380.
- [7] A. Chertock, A. Kurganov, On splitting-based numerical methods for convection–diffusion equations, in: G. Puppo, G. Russo (Eds.), *Numerical Methods for Balance Laws*, in: *Quad. Mat.*, vol. 24, Dept. Math., Seconda Univ. Napoli, Caserta, 2009, pp. 303–343.
- [8] A. Chertock, A. Kurganov, G. Petrova, Fast explicit operator splitting method. Application to the polymer system, in: F. Benkhalidoun, D. Ouazar, S. Raghy (Eds.), *Finite Volumes for Complex Applications IV*, Hermes Science Publishing, 2005, pp. 63–72.
- [9] A. Chertock, A. Kurganov, G. Petrova, Fast explicit operator splitting method for convection–diffusion equations, *Int. J. Numer. Methods Fluids* 59 (2009) 309–332.
- [10] X. Feng, T. Tang, J. Yang, Long time numerical simulations for phase-field problems using p -adaptive spectral deferred correction methods, *SIAM J. Sci. Comput.* 37 (2015) A271–A294.
- [11] F. Guillén-González, G. Tierra, On linear schemes for a Cahn–Hilliard diffuse interface model, *J. Comput. Phys.* 234 (2013) 140–171.
- [12] F. Guillén-González, G. Tierra, Second order schemes and time-step adaptivity for Allen–Cahn and Cahn–Hilliard models, *Comput. Math. Appl.* 68 (2014) 821–846.
- [13] Y. He, Y. Liu, T. Tang, On large time-stepping methods for the Cahn–Hilliard equation, *Appl. Numer. Math.* 57 (2007) 616–628.
- [14] W. Hundsdorfer, J. Verwer, *Numerical Solution of Time-Dependent Advection–Diffusion–Reaction Equations*, Springer Ser. Comput. Math., vol. 33, Springer-Verlag, Berlin, 2003.
- [15] B. Li, J.-G. Liu, Thin film epitaxy with or without slope selection, *Eur. J. Appl. Math.* 14 (2003) 713–743.
- [16] G.I. Marchuk, *Metody Rasshchepleniya (Splitting Methods)*, Nauka, Moscow, 1988 (in Russian).
- [17] G.I. Marchuk, Splitting and alternating direction methods, in: *Handbook of Numerical Analysis*, vol. I, North-Holland, Amsterdam, 1990, pp. 197–462.
- [18] A. Medovikov, High order explicit methods for parabolic equations, *BIT Numer. Math.* 38 (1998) 372–390.
- [19] A.A. Medovikov, DUMKA3 code, available at <http://dumkaland.org/>.
- [20] D. Moldovan, L. Golubovic, Interfacial coarsening dynamics in epitaxial growth with slope selection, *Phys. Rev. E* 61 (2000) 6190–6214.
- [21] Z. Qiao, Z. Zhang, T. Tang, An adaptive time-stepping strategy for the molecular beam epitaxy models, *SIAM J. Sci. Comput.* 33 (2011) 1395–1414.
- [22] A. Scheel, Spinodal decomposition and coarsening fronts in the Cahn–Hilliard equation, preprint, 2014.
- [23] J. Shen, C. Wang, X. Wang, S.M. Wise, Second-order convex splitting schemes for gradient flows with Ehrlich–Schwoebel type energy: application to thin film epitaxy, *SIAM J. Numer. Anal.* 50 (2012) 105–125.
- [24] B.P. Sommeijer, L.F. Shampine, J.G. Verwer, RKC: an explicit solver for parabolic PDEs, *J. Comput. Appl. Math.* 88 (1998) 315–326.
- [25] G. Strang, On the construction and comparison of difference schemes, *SIAM J. Numer. Anal.* 5 (1968) 506–517.
- [26] G. Tierra, F. Guillén-González, Numerical methods for solving the Cahn–Hilliard equation and its applicability to related energy-based models, *Arch. Comput. Methods Eng.* 22 (2015) 269–289.
- [27] P.J. van der Houwen, B.P. Sommeijer, On the internal stability of explicit, m -stage Runge–Kutta methods for large m -values, *Z. Angew. Math. Mech.* 60 (1980) 479–485.
- [28] J.G. Verwer, W.H. Hundsdorfer, B.P. Sommeijer, Convergence properties of the Runge–Kutta–Chebyshev method, *Numer. Math.* 57 (1990) 157–178.
- [29] J.G. Verwer, B.P. Sommeijer, W. Hundsdorfer, RKC time-stepping for advection–diffusion–reaction problems, *J. Comput. Phys.* 201 (2004) 61–79.
- [30] C. Xu, T. Tang, Stability analysis of large time-stepping methods for epitaxial growth models, *SIAM J. Numer. Anal.* 44 (2006) 1759–1779.
- [31] Z. Zhang, Z. Qiao, An adaptive time-stepping strategy for the Cahn–Hilliard equation, *Commun. Comput. Phys.* 1 (2012) 1261–1278.