Available online at www.sciencedirect.com
ScienceDirect
Journal of Computational Physics 225 (2007) 1849–1862
JOURNAL OF COMPUTATIONAL PHYSICS
www.elsevier.com/locate/jcp

A nonstiff, adaptive mesh refinement-based method for the Cahn–Hilliard equation

Hector D. Ceniceros [a], Alexandre M. Roma [b,*]

[a] Department of Mathematics, University of California, Santa Barbara, CA 93106, United States
[b] Departamento de Matemática Aplicada, Universidade de São Paulo, Caixa Postal 66281, CEP 05311-970, São Paulo, SP, Brazil

Received 10 November 2006; received in revised form 8 February 2007; accepted 21 February 2007
Available online 2 March 2007

Abstract

We present a nonstiff, fully adaptive mesh refinement-based method for the Cahn–Hilliard equation. The method is based on a semi-implicit splitting, in which linear leading order terms are extracted and discretized implicitly, combined with a robust adaptive spatial discretization. The fully discretized equation is written as a system which is efficiently solved on composite adaptive grids using the linear multigrid method without any constraint on the time step size. We demonstrate the efficacy of the method with numerical examples. Both the transient stage and the steady state solutions of spinodal decompositions are captured accurately with the proposed adaptive strategy. Employing this approach, we also identify several stationary solutions of that decomposition on the 2D torus.

Keywords: Adaptive method; Conservative phase field models; Spinodal decomposition; Adaptive mesh refinements; Semi-implicit methods; Multilevel multigrid; Biharmonic equation

1. Introduction

Conservative phase field models have received renewed attention for the simulation of multi-phase, multi-component fluids. These diffuse interface models are appealing because the mean of the order parameter or phase field ϕ is preserved, ϕ has a physical meaning, no re-initialization is required to maintain well-defined interfaces, and different physical effects, such as surfactants [1–3], long-range forces [4], viscoelasticiy [5–8] can be modeled by a suitable modification of the free energy. These conservative diffuse interface models share a common feature: a Cahn–Hilliard type equation.

The Cahn–Hilliard equation models the macrophase separation that occurs in an isothermal binary fluid when a spatially uniform mixture is quenched below a critical temperature at which it becomes unstable.

* Corresponding author. Tel.: +55 11 3091 6136; fax: +55 11 3091 6131.
E-mail addresses: roma@ime.usp.br (H.D. Ceniceros), hdc@math.ucsb.edu (A.M. Roma).
URLs: www.ime.usp.br/~roma (H.D. Ceniceros), www.math.ucsb.edu/~hdc (A.M. Roma).

0021-9991/$ - see front matter
doi:10.1016/j.jcp.2007.02.019

Let ϕ be the relative concentration of the two components and let $\phi(\mathbf{x}) \equiv \phi_m = \text{constant}$ correspond to the spatially uniform mixture, then the equilibrium profiles can be found by minimizing the free energy [9]

$$H[\phi] = \int_\Omega \left\{ f(\phi(\mathbf{x})) + \frac{1}{2}\epsilon^2 |\nabla \phi(\mathbf{x})|^2 \right\} \mathrm{d}\mathbf{x}, \tag{1}$$

subject to the conservation of mass constraint

$$\int_\Omega \phi(\mathbf{x})\, \mathrm{d}\mathbf{x} = \phi_m |\Omega|, \tag{2}$$

where Ω is the region of space occupied by the system. The gradient term accounts for the surface energy. The thickness of the interfacial layers is $\mathrm{O}(\epsilon)$, and $f(\phi(\mathbf{x}))$ is the bulk energy density. To be concrete, we take f to be the following symmetric double-well potential

$$f(\phi) = \frac{1}{4}(1-\phi^2)^2. \tag{3}$$

The chemical potential μ is defined as the first variation of (1)

$$\mu(\phi) = \frac{\delta H[\phi]}{\delta \phi(\mathbf{x})} = f'(\phi(\mathbf{x})) - \epsilon^2 \nabla^2 \phi(\mathbf{x}), \tag{4}$$

and thus an equilibrium state corresponds to a solution of $\mu(\phi) = \text{constant}$. Cahn and Hilliard [10,11] generalized the problem to time-dependent situations by approximating interfacial diffusion fluxes as being proportional to chemical potential gradients and enforcing conservation of the field. That is,

$$\frac{\partial \phi(t,\mathbf{x})}{\partial t} = -\nabla \cdot \mathbf{J} \quad \text{with } \mathbf{J} = -\lambda(\phi)\nabla \mu, \tag{5}$$

where $\lambda(\phi) > 0$ is the mobility or Onsager coefficient. This gives the Cahn–Hilliard equation

$$\frac{\partial \phi(t,\mathbf{x})}{\partial t} = \nabla \cdot [\lambda(\phi)\nabla \mu(\phi)], \quad \mu(\phi) = -\epsilon^2 \nabla^2 \phi + f'(\phi), \quad \text{for } \mathbf{x} \in \Omega. \tag{6}$$

Eq. (6) models the creation, evolution, and dissolution of diffusively controlled phase-field interfaces [12] (for a review of the Cahn–Hilliard model see for example [13]). Periodic or no-flux boundary conditions ($\mathbf{n} \cdot \nabla \phi = 0$ and $\mathbf{n} \cdot \lambda \nabla \mu = 0$, where $\mathbf{n}$ is the unit vector normal to the domain boundary) are generally used for the Cahn–Hilliard equation.

Note that the Cahn–Hilliard equation (6) has spatial derivatives of fourth order and a Laplacian acting on the nonlinear term $f'(\phi)$. Thus, explicit time integration methods require a prohibitively small time step for stability. Several fully implicit and semi-implicit time discretizations have been proposed [13–20] to relieve the high order stability constraints. The computation of the initial-value problem for (6) also demands high resolution both in space and time. Spatially, the solution develops fine microstructures in short times, $\mathrm{O}(\epsilon^2)$, and a coarsening process separating the two phases settles in a much longer time scale, $\mathrm{O}(\epsilon^{-1})$. To accurately model real experimental systems, ϵ needs to be very small relative to the domain size which implies that the solution will have sharp gradients of $\mathrm{O}(\epsilon^{-1})$. Thus, the accurate computation of the solution calls for an adaptive approach which can only be made practical if the time discretizations are nonstiff, i.e. free of any high order stability constraint, and robust.

Adaptive finite element methods for the Cahn–Hilliard equation have been proposed in [21–23]. These methods rely on an implicit treatment of both the bihamornic and the nonlinear term to remove the stability constraints. As a result, nonlinear iterative solvers have to be invoked at each time step. Here we present an adaptive finite difference-based alternative that removes the stability constraints without cost of nonlinear iterative solvers. The proposed method combines for the first time an efficient semi-implicit scheme based on the extraction of linear leading order terms [19] with adaptive mesh refinements [24] applied to the Cahn–Hilliard equation recast as a system of second order equations [25,20]. At each time step, that system is solved at optimal cost on the composite adaptive grids using a *linear* multigrid method without any constraint on the time

step size. The remarkable stability properties of this linear semi-implicit method have been established recently by Xu and Tang [26] for the case of some continuum epitaxial growth models.

Despite much work on the Cahn–Hilliard equation, there is still a lack of a complete understanding of its stationary solutions in multi-dimensions (cf. [27]). It is known [28] that minimizers of H (scaled by $1/\epsilon$) converge to minimizers of a sharp interface, isoperimetric problem in which solutions have phase boundaries of constant mean curvature (i.e. circles and lines in 2D). It has been shown [27] that, for small ϵ, minimizers of H in the doubly periodic case (i.e. the 2D torus) exhibit a profile asymptotic to the solutions of the doubly periodic isoperimetric problem. A stationary solution with a circular phase boundary and one with two horizontal strips have been obtained numerically [14,20]. We present here four more stationary states in addition to those already reported. The new stationary states consists of two vertical, four horizontal, left and right slanted strips. Interestingly, these solutions are obtained from a random perturbation of the uniform state $\phi_m = 0$ with some variation on how the initial data are set up in the domain. The interested reader can find these solutions at the end of Section 5.

The rest of the paper is organized as follows. We devote Section 2 to introduce the nonstiff and adaptive time discretization while the adaptive mesh refinement strategy and the spatial discretization are discussed in detail in Section 3. The multilevel multigrid employed to solve the linear system resulting from the time and spatial discretizations is described in Section 4. We document the performance of the fully adaptive method and present examples of spinodal decomposition that lead to different stationary states in Section 5.

2. Time discretization

We consider in this work the case of constant mobility and without loss of generality we take $\lambda \equiv 1$ (one can always rescale time with λ). Following the approach described in [19], we extract the linear leading order terms of (6) and rewrite this equation as

$$\frac{\partial \phi(t,\mathbf{x})}{\partial t} = \nabla^2(\tau\phi - \epsilon^2\nabla^2\phi) + g(\phi), \quad \text{for } \mathbf{x} \in \Omega, \tag{7}$$

where $g(\phi) = \nabla^2(f'(\phi) - \tau\phi)$, τ is a constant and here we use $\tau = \max f''(\phi)$.

Introducing the auxiliary variables

$$\varphi_1 = \phi \quad \text{and} \quad \varphi_2 = \tau\phi - \epsilon^2\nabla^2\phi, \tag{8}$$

the modified equation (7) can be viewed as the system

$$\frac{\partial \varphi_1(t,\mathbf{x})}{\partial t} = \nabla^2\varphi_2 + g(\varphi_1), \tag{9}$$

$$\varphi_2(t,\mathbf{x}) = \tau\varphi_1 - \epsilon^2\nabla^2\varphi_1, \tag{10}$$

for $\mathbf{x} \in \Omega$, where $g(\varphi_1) = \nabla^2(f'(\varphi_1) - \tau\varphi_1)$. Here, we consider only periodic boundary conditions, i.e. φ_1 and φ_2 are periodic, and the domain Ω is a rectangle in the plane. We note that Zhu et al. [17] used a similar approach to treat a variable mobility case. They left however the nonlinear, potential term explicit in their time discretization which carries a consequent time step restriction. The full extraction of the leading linear terms for the Cahn–Hilliard equation as in (7) was first done in [19].

2.1. Adaptive extrapolated gear (SBDF) scheme

Among the large class of semi-implicit second order methods, we choose the semi backward difference formula (SBDF) or extrapolated Gear scheme because of its high modal frequency damping and robust stability properties [29]. It has been recognized that non-dissipative schemes such as the popular Crank–Nicolson generally perform poorly for the Cahn–Hilliard equation [21,19]. Indeed, by testing a Crank–Nicolson based method for the system (9), (10), we found that this was the case.

We modify the traditional SBDF to allow for variable time step and apply it to the system (9) and (10), obtaining

$$\frac{\alpha_2\varphi_1^{n+1}+\alpha_1\varphi_1^{n}+\alpha_0\varphi_1^{n-1}}{\Delta t}=\nabla^2\varphi_2^{n+1}+\beta_1 g^n+\beta_0 g^{n-1}, \tag{11}$$

$$\varphi_2^{n+1}=\tau\varphi_1^{n+1}-\epsilon^2\nabla^2\varphi_1^{n+1}, \tag{12}$$

where $\alpha_0=\Delta t^2/(\Delta t_0\Delta t_1)$, $\alpha_1=-\Delta t_1/\Delta t_0$, and $\alpha_2=(\Delta t_0+2\Delta t)/\Delta t_1$, $\beta_0=-\Delta t/\Delta t_0$ and $\beta_1=\Delta t_1/\Delta t_0$, with $\Delta t=t^{n+1}-t^n$, $\Delta t_0=t^n-t^{n-1}$, and $\Delta t_1=\Delta t_0+\Delta t$.

For fixed time step, the coefficients above assume their usual, constant values $\alpha_0=1/2$, $\alpha_1=-2$, and $\alpha_2=3/2$, $\beta_0=-1$ and $\beta_1=2$. Eqs. (11) and (12) can be re-written as

$$\frac{\alpha_2\varphi_1^{n+1}}{\Delta t}-\nabla^2\varphi_2^{n+1}=-\frac{\alpha_1\varphi_1^{n}}{\Delta t}-\frac{\alpha_0\varphi_1^{n-1}}{\Delta t}+\beta_1 g^n+\beta_0 g^{n-1}, \tag{13}$$

$$-\left[\tau\varphi_1^{n+1}-\epsilon^2\nabla^2\varphi_1^{n+1}\right]+\varphi_2^{n+1}=0. \tag{14}$$

Since the SBDF is a two-step method, an approximation at $t=t^1$ for (φ_1,φ_2) is required in addition to the initial condition. This approximation is obtained by using the semi-implicit Euler's method:

$$\frac{\varphi_1^{n+1}-\varphi_1^{n}}{\Delta t}=\nabla^2\varphi_2^{n+1}+g^n, \tag{15}$$

$$\varphi_2^{n+1}=\tau\varphi_1^{n+1}-\epsilon^2\nabla^2\varphi_1^{n+1}, \tag{16}$$

which is enough for achieving second order rates of convergence for the SBDF.

3. Discretization in space

3.1. Composite grid generation and dynamic adaptation

To capture accurately and efficiently the small scale structures in the solutions of the Cahn–Hilliard equation (7), we employ a *composite grid*, that is, a block-structured grid defined as a hierarchical sequence of nested, progressively finer grid levels [24]. Each level is formed by a set of non-overlapping rectangular grid patches aligned with the coordinate axes and the refinement ratio between two successive levels is two. *Ghost cells*, are employed around each grid, for all levels, and underneath fine grid patches. Values at these extended cells are obtained from interpolation to prevent the finite difference operators from being redefined at grid borders and at interior regions which are covered by finer levels.

While the generation of this type of grid is mature, the effective application of the adaptive mesh refinement (AMR) technique to the Cahn–Hilliard equation requires addressing three important problems:

(1) *Effective flagging*, that is, to determine the cells whose collection gives the region where refinement is to be applied.
(2) *Accurate interpolations* across grid interfaces to guarantee global high-order accuracy in the solution.
(3) *Fast multilevel multigrid solvers* which are cost-efficient on composite grids.

We tested two flagging strategies. One, based on the gradient of the order parameter for which we flag the cells with indices *rs* such that

$$\|\nabla\phi_{rs}\|/\|\nabla\phi\|_\infty\geqslant\delta, \tag{17}$$

and another one, for which we flag the cells with indices *rs* with relative values of the order parameter close to zero, that is, cells close to the phase transition layer satisfying

$$|\phi_{rs}|/|\phi|_\infty\leqslant\gamma, \tag{18}$$

where the positive "tunning" parameters δ and γ satisfy $\delta<1$ and $\gamma\ll 1$. We found that the latter performs better for the spinodal decomposition problem considered here and thus it is the strategy we have adopted. Typical runs employed $\gamma=0.35$. Once the collection of flagged cells is obtained, grids in each level are generated by applying the algorithm for point clustering introduced by Berger and Rigoutsos [30]. In each grid patch 75–85% of its cells are flagged (grid efficiency). The rest of the cells are included so the grid patch is rectangular.

Initially, when the phases are completely mixed, we cover the computational domain entirely with the finest level. This is to ensure that the spinodal patterns at late times correspond to those that would be obtained by using a uniform mesh with a mesh size equal to that of the finest level. Thus, the computational cost is the highest in the very few initial time steps and progressively drops down as the phase domains begin to form and coarsen. For example, with four levels of refinement the computational cost drops down more that an order of magnitude. The savings increase as the number of levels of refinement increases. Note that to accurately resolve the phase field, the number of refinement levels must be chosen such that the mesh size of the finest level is $O(\epsilon)$, which is the smallest scale associated with the width of the phase transition layers.

The composite grid must be replaced (dynamic adaptation) in two situations. First, when phase transition layers tend to escape or to form outside the regions covered by the finest level. Second, at every certain fixed number of time steps (e.g. at every 500 time steps) to "refresh" the composite grid. Without the latter, composite grids generated with finest levels covering a large portion of the domain would tend to stay permanently in use and the integration would become inefficient.

The presence of fourth order derivatives in the Cahn–Hilliard equation poses the question of whether or not highly accurate interpolation schemes are required for the computation of ghost values at the borders of each grid patch to achieve global second order accuracy. This question is well studied for second order problems. However, for fourth-order equations, it has not been investigated much, neither analytically nor numerically. It turns out, as our numerical experiments will demonstrate, that the very same interpolation schemes used for computing ghost values in second order problems (e.g. [31]) work rather well with the linear, multilevel multigrid method proposed here for the Cahn–Hilliard equation. More specifically, second order polynomial interpolations at grid interfaces with third order interpolations only near T-junctions of two grid patches of the same level are sufficient to obtain global second order accuracy for the order parameter.

Next, we describe the discrete spatial operators and devote Section 4 to the discussion of the proposed linear multilevel multigrid solver.

3.2. Gradient, divergence and Laplacian difference operators

The phase field, and the auxiliary variables φ_1 and φ_2 are placed at cell centers. Second order approximations for their first order derivatives are given by the centered finite difference operators

$$D_x\phi_{i-1/2,j} = \frac{\phi_{i,j} - \phi_{i-1,j}}{h} \quad \text{and} \quad D_y\phi_{i,j-1/2} = \frac{\phi_{i,j} - \phi_{i,j-1}}{h}, \tag{19}$$

where for simplicity, it is assumed that the mesh size is equal to h in both directions. We observe that (19) defines the derivatives at the cell edges. Fig. 1 shows the locations of the variables and their first-order derivatives near an interface between two successive refinement levels.

At coarse cell edges, underneath fine grid patches belonging to the next finer level, the first derivative is defined by the average of the corresponding finer ones. Referring to Fig. 1, the x derivative at the coarse cell edge covered by the border of the fine grid patch displayed, is given by

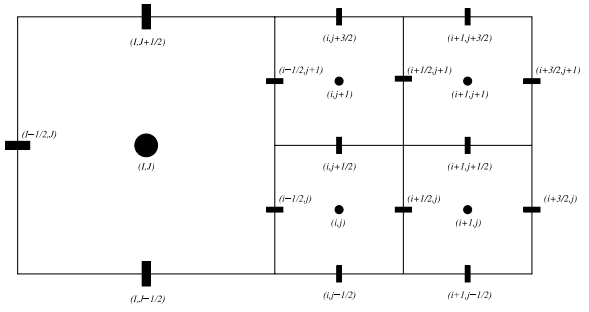

Fig. 1. Location of coarse and fine variables.

$$D_x\phi_{I+1/2,J} = \frac{1}{2}(D_x\phi_{i-1/2,j} + D_x\phi_{i-1/2,j+1}). \tag{20}$$

Based on (19) and (20), the gradient of a cell centered variable can be defined as

$$G\phi_{i,j} = (D_x\phi_{i-1/2,j}, D_y\phi_{i,j-1/2}). \tag{21}$$

The divergence operator defined for a vector $\psi_{i,j} = (\psi_{1_{i-1/2,j}}\psi_{2_{i,j-1/2}})$, whose components are located at the center of the cell edges, is given at the cell center by

$$\mathbf{D}\cdot\psi_{i,j} = \frac{\psi_{1_{i+1/2,j}} - \psi_{1_{i-1/2,j}}}{h} + \frac{\psi_{2_{i,j+1/2}} - \psi_{2_{i,j-1/2}}}{h}. \tag{22}$$

By simply composing the divergence and the gradient operators (21) and (22), an approximation for the Laplacian operator can be naturally obtained for cell-centered variables on the composite grid, such that the result of

$$\mathbf{D}\cdot G\phi_{i,j} = \frac{1}{h}(D_x\phi_{i+1/2,j} - D_x\phi_{i-1/2,j}) + \frac{1}{h}(D_y\phi_{i,j+1/2} - D_y\phi_{i,j-1/2}), \tag{23}$$

is defined at cell centers (note that, since the first derivatives are computed by averages close to coarse-fine grid interfaces, (23) is not the 5-point stencil there). This is the approximation of the Laplacian we employ, except during the *smoothing steps,* in the linear multilevel multigrid method, to be detailed next, where the relaxations are based only on the standard 5-point stencil (no averaging involved).

Introducing the discretization in space, (13) and (14) assume the form

$$\alpha_2\varphi^{n+1}_{1_{i,j}}/\Delta t - \mathbf{D}\cdot G\varphi^{n+1}_{2_{i,j}} = b_{1_{i,j}}, \tag{24}$$

$$-(\tau\varphi^{n+1}_{1_{i,j}} - \epsilon^2\mathbf{D}\cdot G\varphi^{n+1}_{1_{i,j}}) + \varphi^{n+1}_{2_{i,j}} = b_{2_{i,j}}, \tag{25}$$

where $b_{1_{i,j}} = -\alpha_1\varphi^{n}_{1_{i,j}}/\Delta t - \alpha_0\varphi^{n-1}_{1_{i,j}}/\Delta t + \beta_1 g^n_{i,j} + \beta_0 g^{n-1}_{i,j}$, and $b_{2_{i,j}} = 0$.

Note that, by working with the Canh–Hilliard equation as a system, we avoid the direct discretization of the biharmonic differential operator. The linear system in the unknowns $\varphi^{n+1}_{1_{i,j}}$ and $\varphi^{n+1}_{2_{i,j}}$, (24) and (25), can be efficiently solved by a linear multilevel multigrid method as we detail next.

4. Linear, multilevel multigrid method

We first describe the application of the linear multigrid method for a uniform grid and then we comment on how to modify the procedure appropriately for the multilevel context of a composite grid (here, the term *multilevel*, refers to the actual refinement levels of the adaptive grid and not to the "virtual" levels needed in the multigrid method).

We employ the V-cycle schedule within a *coarse grid correction scheme* [32]. Given an *initial guess*, for the solution of (24) and (25) on each computational cell (i,j), $\varphi^{n+1,0}_{i,j} = (\varphi^{n+1,0}_{1_{i,j}}, \varphi^{n+1,0}_{2_{i,j}})$, we define the *correction* $\mathbf{e}_{i,j}$ as the difference between exact and approximate solutions,

$$\mathbf{e}_{i,j} = \varphi^{n+1}_{i,j} - \varphi^{n+1,0}_{i,j}, \tag{26}$$

and, from (24) and (25), we define the *residual*, $\mathbf{r}_{i,j} = (r_{1_{i,j}} r_{2_{i,j}})$ by

$$r_{1_{i,j}} = b_{1_{i,j}}, -(\alpha_2\varphi^{n+1,0}_{1_{i,j}}/\Delta t - \mathbf{D}\cdot G\varphi^{n+1,0}_{2_{i,j}}), \tag{27}$$

$$r_{2_{i,j}} = b_{2_{i,j}}, -[-(\tau\varphi^{n+1,0}_{1_{i,j}} - \epsilon^2\mathbf{D}\cdot G\varphi^{n+1,0}_{1_{i,j}}) + \varphi^{n+1,0}_{2_{i,j}}], \tag{28}$$

for each computational cell (i,j).

One of the most important elements in a multigrid method is the relaxation operation or *smoothing step*. Relaxation methods should eliminate effectively the high-frequency components of the error, while leaving the low-frequency components relatively untouched. We employ a red-black relaxation (smoothing) scheme based on the linear system (24) and (25), which is given by

$$\begin{bmatrix} \alpha_2/\Delta t & 4/h^2 \\ -(\tau + 4\epsilon^2/h^2) & 1 \end{bmatrix} \begin{bmatrix} e^k_{1_{i,j}} \\ e^k_{2_{i,j}} \end{bmatrix} = \begin{bmatrix} rhs_{1_{i,j}} \\ rhs_{2_{i,j}} \end{bmatrix}, \tag{29}$$

with

$$rhs_{1_{i,j}} = r_{1_{i,j}}, + \frac{e^l_{2_{i-1,j}} + e^l_{2_{i+1,j}} + e^l_{2_{i,j-1}} + e^l_{2_{i,j+1}}}{h^2}, \tag{30}$$

and

$$rhs_{2_{i,j}} = r_{2_{i,j}}, -\epsilon^2, \frac{e^l_{1_{i-1,j}} + e^l_{1_{i+1,j}} + e^l_{1_{i,j-1}} + e^l_{1_{i,j+1}}}{h^2}. \tag{31}$$

In the linear system for the correction, (29), k is the relaxation index and, in (30) and (31), either $l = k - 1$ or $l = k$ depending on whether (i, j) determines a red or a black cell. Note that a slightly different two-by-two linear system must be solved during the smoothing step for the semi-implicit Euler method employed in the first time step.

One complete smoothing step is given by solving (29)–(31) successively, in turns, once for the red cells and once for the black cells. When going down in the V-cycle, smoothing is performed by relaxing the correction v_1 times before restricting down the residual, and by relaxing it v_2 times when going up, after prolongating coarser corrections up. As shown in Section 5, good rates of convergence can be obtained by simply taking $v_1 = v_2 = 1$, and by performing restrictions down by *simple average* of fine residual values and prolongations up by *bilinear interpolation* of coarse correction values.

Typically, each (1,1) V-cycle reduces the residual by a factor of approximately 10. For the cases run, only from 6 to 11 of these cycles are needed to decrease the residual to $O(h^2)$ (regardless uniform or composite grids, h being the mesh size of the finest level in the latter case). Periodic boundary conditions are employed for each of the systems to be solved.

In the composite grid context, each level of refinement is also viewed as one of the virtual multigrid levels. In this case, applying the method requires some additional steps since refinement levels do not completely cover the computational domain. During the smoothing sweeps, ghost cells appended to grid borders are updated immediately after each red or black relaxation sweep for all grids in the same level. Also, even though the difference operator defined by (23) cease to be the usual 5-point discretization for the Laplacian at coarse-fine level interfaces on composite grids, relaxations are still performed by employing (29)–(31) for the residual-correction equation. Only when computing residuals at these locations, the first order derivatives appearing in (23) are obtained as the simple average of the finer ones, before computing the Laplacian employing the difference operator $\mathbf{D} \cdot G$.

5. Numerical results

5.1. Numerical validation of the approach

We now validate the proposed approach by performing a convergence-under-refinement analysis for the *forced*, Cahn–Hilliard equation

$$\frac{\partial \phi}{\partial t}(t, \mathbf{x}) = \nabla^2 \mu(\phi(t, \mathbf{x})) + F(t, \mathbf{x}), \tag{32}$$

$$\mu(\phi) = f'(\phi) - \epsilon^2 \nabla^2 \phi(t, \mathbf{x}), \tag{33}$$

where $F(t, \mathbf{x})$ is a forcing term, and $f'(\phi) = \phi^3 - \phi$ is the first derivative of the double-well potential (3). To setup a smooth model problem, we first choose the function

$$\phi_e(t, \mathbf{x}) = -1 + \kappa, \left[-\exp(-1), +, \exp\left(\cos(2\pi x + 2\pi y + w, t)\right)\right], \tag{34}$$

as the exact solution of (32) and (33), for $0 \leqslant t \leqslant 10$, $(x, y) = \mathbf{x} \in [0, 1] \times [0, 1]$, where $\kappa = 2/[\exp(1) - \exp(-1)]$, and $w = 20\pi$. Thus, the forcing term is given by

$$F(t,\mathbf{x}) = \frac{\partial \phi_e}{\partial t}(t,\mathbf{x}) - \nabla^2 \mu(\phi_e(t,\mathbf{x})), \tag{35}$$

and the initial condition by $\phi(0,\mathbf{x}) = \phi_e(0,\mathbf{x})$. Note that $-1 \leqslant \phi_e(t,\mathbf{x}) \leqslant +1$ for any point $(t,\mathbf{x})$. We take $\epsilon^2 = 5 \times 10^{-2}$ in the chemical potential (33).

For the results that follow, we adopt doubly periodic boundary conditions. The parameter τ appearing in (7) is chosen to be

$$\tau = \max_{-1\leqslant\phi\leqslant+1} \{f''(\phi)\} = 2,$$

and the time step is selected as $\Delta t = \Delta x$, which is kept fixed throughout the integration of this test problem. Note that this choice of Δt is for illustrating the rate of convergence of the method and *not* from a stability requirement. The method behaved as unconditionally stable in all the numerical tests we conducted.

With the above choice of parameters we obtain the convergence results summarized in Table 1 for a sequence of $n \times n$ uniform grids. These results show a clear second order convergence behavior of the method on uniform grids.

Next, on the composite grid shown in Fig. 2, we perform a standard *static-grid* test. In this case, the grid was selected arbitrarily and kept fixed at all times. The purpose of this test is to verify that the truncation errors introduced by interpolation and the discretization schemes at coarse-fine level interfaces are correctly controlled to prevent global accuracy degradation. Table 2 shows the results for this case.

In Table 2, the notation "$n+1$" in the first line, for $n = 128$, 256, and 512, stands for "a two-level composite grid formed by a $n \times n$ uniform grid (level 1) plus one additional refinement level (level 2)". The results indicate again a clear second order convergence behavior for the numerical scheme in the presence of coarse-fine grid interfaces (even in the maximum norm).

Table 1
L_2- and L_∞-norms of the errors, and convergence ratios on uniform grids ($t = 10$)

n	128	Ratio	256	Ratio	512
$\|\phi_n - \phi_e\|_2$	2.41×10^{-1}	3.81	6.33×10^{-2}	3.98	1.59×10^{-2}
$\|\phi_n - \phi_e\|_\infty$	5.28×10^{-1}	3.67	1.44×10^{-1}	3.91	3.68×10^{-2}

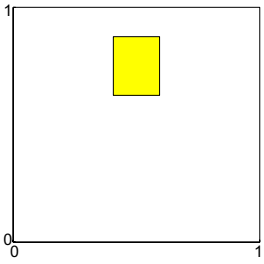

Fig. 2. Composite grid employed in the static-grid test.

Table 2
L_2- and L_∞-norms of the errors, and convergence ratios on composite grids ($t = 10$)

$n+1$	128+1	Ratio	256+1	Ratio	512+1
$\|\phi_n - \phi_e\|_2$	6.36×10^{-2}	3.98	1.60×10^{-2}	4.00	4.00×10^{-3}
$\|\phi_n - \phi_e\|_\infty$	1.45×10^{-1}	3.91	3.71×10^{-2}	3.98	9.32×10^{-3}

5.2. Capturing spinodal decomposition with the fully adaptive strategy

Next, we consider the process of spinodal decomposition to illustrate the performance of the fully adaptive method. We first describe our space and time adaptive strategy which is then followed by the numerical results.

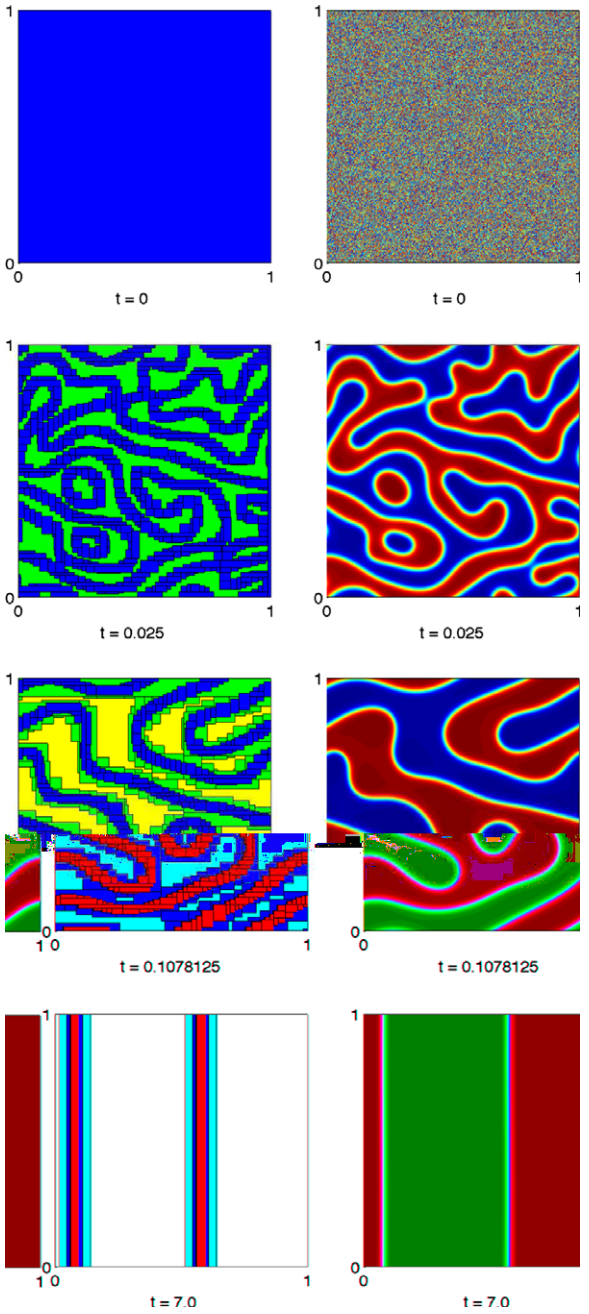

Fig. 3. Spinodal decomposition. The composite adaptive mesh (left column) and the phase field (right column). The refinement levels appear as grid patches of different colors. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

5.2.1. The adaptive strategy

The time step size Δt is carefully selected to accurately resolve both the initial fast dynamics and the late slower, coarsening motion while retaining at all times a monotone decrease of the energy and second order accuracy. Initially, we take $\Delta t = 2.5\epsilon^2$ and integrate up to $t = 10\epsilon$ with this time step. Then, we increase the time step to $\Delta t = h$ to speed up the computations while retaining second order accuracy. This time step size selection is solely based on accuracy and on the need to capture the fast initial dynamics and not imposed by a stability constraint. Indeed, we have tested the semi-implicit scheme with $\Delta t = O(1)$ on both uniform and composite grids with resolutions up to $1/512$ and found it to be always stable.

We cover initially the computational domain entirely with the finest level. Once the phase domains begin to form, the adaptive composite mesh is automatically triggered, keeping the phase transition layers covered by the finest level patches at all times, employing the flagging criterion explained in 3.1.

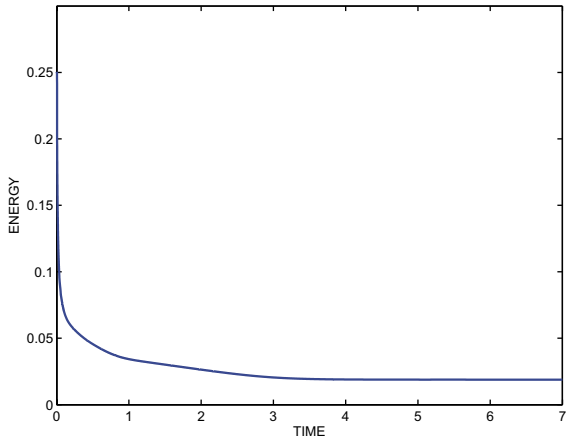

Fig. 4. Energy versus time for an initially random perturbation of a uniform equal mixture, $\epsilon = 0.01$ and $\tau = 2$.

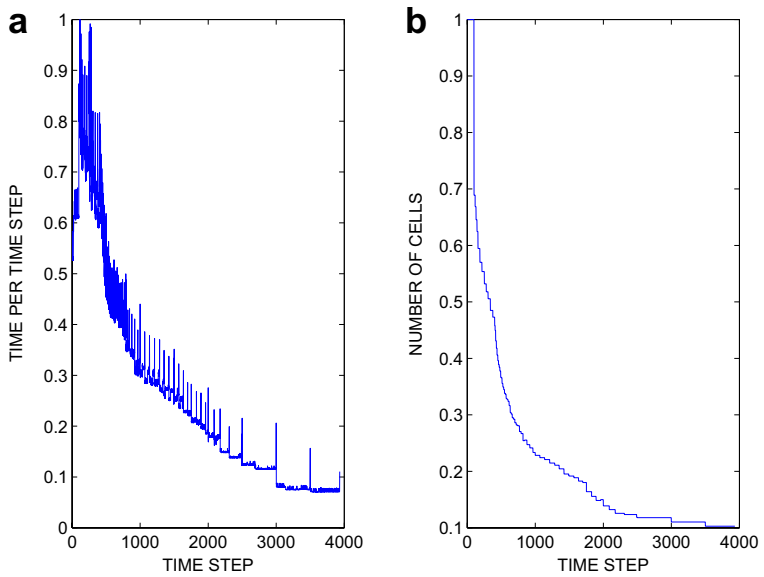

Fig. 5. Performance of the fully adaptive strategy: (a) relative CPU time per time step, and (b) relative number of computational cells per time step (scaled by the total number of the initial uniform fine grid).

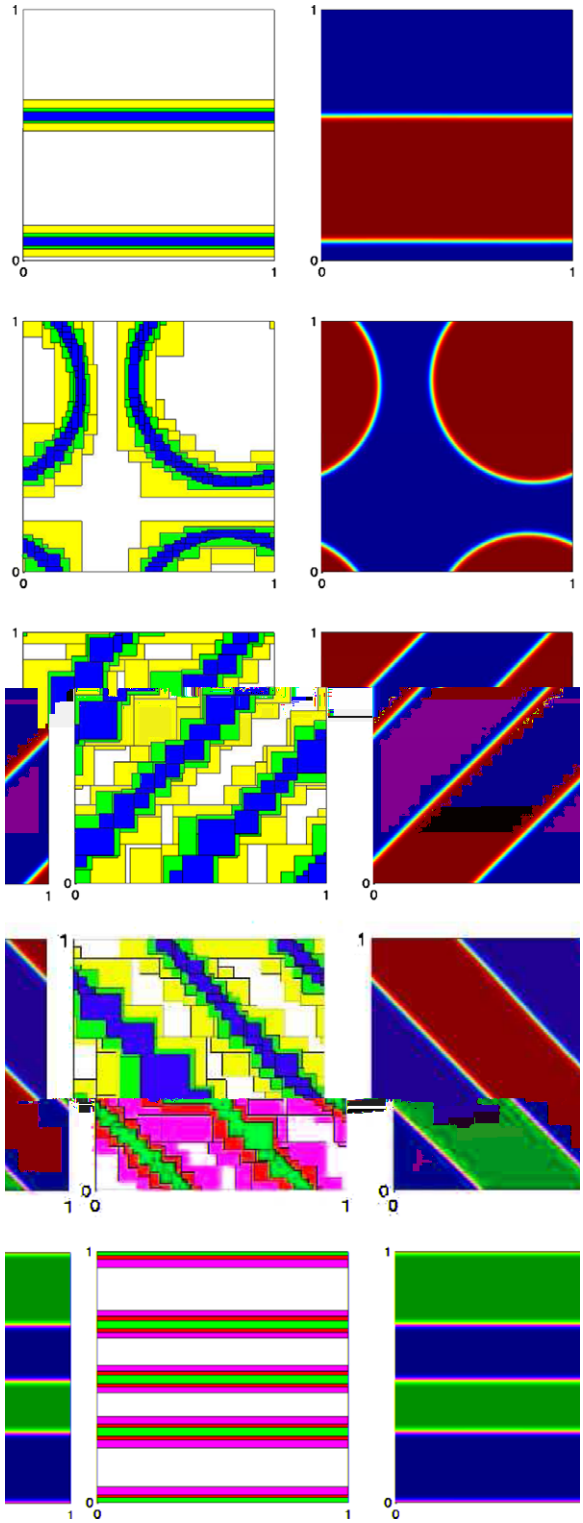

Fig. 6. Different stationary states ($t = 7$) starting from a random perturbation of an equal composition mixture.

5.2.2. Dynamics of a perturbed equal composition

We take the initial condition to be a random perturbation of a uniform equal mixture

$$\phi(0,\mathbf{x}) = \epsilon r(\mathbf{x}), \quad \mathbf{x} \in [0,1]\times[0,1], \tag{36}$$

where the random function $r(\mathbf{x}) \in [-1,1]$, has zero mean. In (36), $\epsilon = 0.01$, and $\tau = 2$ in (7).

Fig. 3 presents a series of snapshots of both the composite adaptive mesh (left column) and the phase field ϕ (right column). The refinement levels are shown as grid patches of different colors. After a very fast initial dynamics, the phase domains are already well defined at $t = O(\epsilon)$ (second row) and a composite grid with two levels suffices to efficiently cover the thin domains and their boundaries. As the phase domains coarsen, around $t = 0.36$ (third row), the adaptive mesh is composed of three levels of refinement but shortly after that (around $t = 1.1$) and up to until the final time $t = 7$ four levels are employed. Note that at all times the finest level covers entirely the phase boundaries. As Fig. 4 demonstrates, with the adaptive time step selection detailed in Section 5.2.1, we obtain a monotonic decrease of the free energy per time step. The stationary-state value of free energy scaled by the interfacial width, H/ϵ, is 1.8811. The mean of the order parameter is also preserved up to $O(h^2)$ at all times.

The performance of the fully adaptive strategy is documented in Fig. 5 in terms of (a) the relative CPU time and (b) the relative number of computational cells as functions of the time step number. Because of the use of the uniform fine mesh in the initial transient stage, most of the computational work is spent during that short time interval; 100 time steps out of a total of 3393. As Fig. 5 shows, once the phase domains are formed and begin to coarsen the CPU time and the total number of computational cells decrease an order of magnitude with the adaptive method. The spikes in the plot shown in Fig. 5a correspond to the events when remeshing was performed (short spikes), and when output data were written out for visualization purposes (long spikes).

5.2.3. Multiple doubly-periodic stationary states

While developing the adaptive strategy that captures accurately both the transient and the stationary states of the solution, we found several stationary solutions. These solutions began with the same random perturbation of the uniform state (36), but these initial data were distributed differently on the initial mesh. This serendipitous finding lead us to consider a fixed set of randomly generated numbers $\{\epsilon r_j\}_{j=1}^{k}$, $-1 \leqslant r_j \leqslant +1$, $j = 1,2,\ldots,k$, with zero mean, and to explore different distributions of these points on the initial (fine) uniform mesh. For example, we distributed these numbers on the mesh column-wise or row-wise. In another case, trying to favor different modes, we divided the computational domain into four equal sub-domains and visited each of these in a particular order (e.g. diagonally, lower left, upper right lower right, and upper left). Fig. 6 displays the different stationary states and the corresponding adaptive meshes we obtained with this procedure. Not surprisingly, the solution in the first row of Fig. 6 has the same relative stationary energy as that given by two vertical strips, $H/\epsilon = 1.8811$ (Fig. 3), and half of the energy of the stationary solution given by the four strips, $H/\epsilon = 3.7623$ (Fig. 6, last row). For the other stationary solutions on the second to the fourth row of Fig. 6, H/ϵ is equal to 2.3547, 2.6634, and 2.6634, respectively. In all the cases, both the final patterns and the energy remain the same even for very long times (we continued the computations up $t = 50$) and thus these stationary states appear to be fairly stable. These stationary solutions are consistent with the recent result [27] that, for small ϵ, minimizers of H in the 2D torus exhibit a profile asymptotic to the solutions of the corresponding isoperimetric problem which in 2D are circles and lines.

In the case when $\tau = 0$ in the splitting (7), only the biharmonic term is treated implicitly and thus the computations require a somewhat restrictive time step. Interestingly, we have found that for the same random initial condition that lead to the circular stationary state for $\tau = 2$, the scheme with $\tau = 0$ (and $\Delta t \approx Ch^2$) selects a stationary state consisting of the two vertical strips. This is not entirely surprising as the schemes for $\tau = 0$ and $\tau = 2$ have different numerical dissipation and the initial condition is random.

6. Conclusions

We presented a robust and efficient numerical method for the Cahn–Hilliard equation which employs adaption both in space and time. The numerical experiments suggest that the methodology is free of stability time stepping constraints. Moreover, it can be orders of magnitude faster than on equivalent uniform grids once the

phase layers are well formed. The time and space discretization on a composite adaptive grid produces a linear system of equations that we solve at optimal cost with a linear multilevel multigrid method. The fully adaptive strategy is capable of capturing accurately both the transient stage and the slow domain coarsening of spinodal decomposition. Using this methodology, we identified several stationary solutions on the 2D torus that, to our knowledge, have not been reported in the literature.

Acknowledgements

Partial support for this research was provided by the National Science Foundation under Grant number DMS 0609996 (HDC), and by the Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP) under Grant numbers 04/13781-1 and 06/57099-5 (AMR). The authors thank Peter Sternberg for insightful discussions about the stationary states and the connection with the isoperimetric problem.

References

[1] G. Gompper, S. Zschocke, Ginzburg–Landau theory of oil–water–surfactant mixtures, Phys. Rev. A 46 (1992) 4836–4851.
[2] G. Gompper, S. Zschocke, Ginzburg–Landau theory of ternary amphiphilic sytems I. Gaussian interface fluctuations, Phys. Rev. E 47 (1993) 4289–4300.
[3] G. Gompper, S. Zschocke, Ginzburg–Landau theory of ternary amphiphilic sytems II. Monte Carlo simulations, Phys. Rev. E 47 (1993) 4301–4312.
[4] P. Español, Thermohydrodynamics for a van der Waals fluid, J. Chem. Phys. 115 (2001) 5392–5403.
[5] F. Boyer, L. Chupin, P. Fabrie, Numerical study of viscoelastic mixtures through a Cahn–Hilliard flow model, Euro. J. Mech. B/Fluids 23 (2004) 759–780.
[6] P. Yue, J.J. Feng, C. Liu, J. Shen, A diffuse-interface method for simulating two-phase flows of complex fluids, J. Fluid Mech. 515 (2004) 293–317.
[7] P. Yue, J.J. Feng, C. Liu, J. Shen, Viscoelastic effects on drop deformation in steady shear, J. Fluid Mech. 540 (2005) 427–437.
[8] P. Yue, J.J. Feng, C. Liu, J. Shen, Diffuse-interface simulations of drop-coalescence and retraction in viscoelastic fluids, J. Non-Newtonian Fluid Dyn. 129 (2005) 163–176.
[9] O. Penrose, P. Fife, Thermodynamically consistent models of phase-field type for the kinetics of phase transitions, Physica D 43 (1990) 44.
[10] J.W. Cahn, J.E. Hilliard, Free energy of a nonuniform system I, J. Chem. Phys. 28 (1958) 258.
[11] J.W. Cahn, J.E. Hilliard, Free energy of a nonuniform system III, J. Chem. Phys. 31 (1959) 688.
[12] P.W. Bates, P.C. Fife, The dynamics of nucleation for the Cahn–Hilliard equation, SIAM J. Appl. Math. 53 (1993) 990.
[13] C.M. Elliot, The Cahn–Hilliard model for the kinetics of phase separation, in: J.F. Rodrigues (Ed.), Mathematical Models for Phase Change Problems, International Series of Numerical Mathematics, vol. 88, Bikhäuser Verlag, Basel, 1989, pp. 35–72.
[14] M. Copetti, C. Elliot, Kinetics of phase decomposition process: numerical solutions to the Cahn–Hilliard equation, Mater. Sci. Technol. 6 (1990) 273.
[15] Q. Du, R.A. Nicolaides, Numerical analysis of a continuum model of phase transition, SIAM J. Numer. Anal. 28 (5) (1991) 1310–1322.
[16] L. Chen, J. Shen, Applications of semi-implicit Fourier-spectral method to phase field equations, Comput. Phys. Comm. 108 (1998) 147–158.
[17] J. Zhu, L.-Q. Chen, J. Shen, V. Tikare, Coarsening kinetics from a variable-mobility Cahn–Hilliard equation: application of a semi-implicit Fourier spectral method, Phys. Rev. E 60 (4) (1999) 3564–3572.
[18] D. Furihata, A stable and conservative finite difference scheme for the Cahn–Hilliard equation, Numer. Math. 87 (4) (2001) 675.
[19] V.E. Badalassi, H.D. Ceniceros, S. Banerjee, Computation of multiphase systems with phase field models, J. Comput. Phys. 190 (2003) 371–397.
[20] J. Kim, K. Kang, J. Lowengrub, Conservative multigrid methods for Cahn–Hilliard fluids, J. Comput. Phys. 193 (2004) 511–543.
[21] H. Garcke, M. Rumpf, U. Weikard, The Cahn–Hilliard equation with elasticity: finite element approximation and qualitative studies, Interf. Free Boundaries 3 (2001) 101–118.
[22] I. Barosan, P. Anderson, H. Meijer, Application of mortar elements to diffuse-interface methods, Comput. Fluids 35 (2006) 1384–1399.
[23] P. Yue, C. Zhou, J.J. Feng, C.F. Ollivier-Gooch, H.H. Hu, Phase-field simulations of interfacial dynamics in viscoelastic fluids using finite elements with adaptive meshing, J. Comput. Phys. 219 (2006) 47–67.
[24] M.J. Berger, P. Colella, Local adaptive mesh refinement for shock hydrodynamics, J. Comput. Phys. 82 (1989) 64–84.
[25] C.M. Elliott, D. French, F.A. Milner, A second order splitting method of the Cahn–Hilliard equation, Numer. Math. 54 (1989) 575.
[26] C. Xu, T. Tang, Stability analysis of large time-stepping methods for epitaxial growth models, SIAM J. Numer. Anal. 44 (4) (2006) 1759–1779.
[27] R. Choksi, P. Sternberg, Periodic phase separation: the periodic Cahn–Hilliard and the isoperimetric problems, Interf. Free Boundaries 8 (2006) 371–392.

[28] L. Modica, The gradient theory of phase transitions and the minimal interface criterion, Arch. Rational Mech. Anal. 98 (1987) 123–142.
[29] U.M. Ascher, S.J. Ruuth, B. Wetton, Implicit–explicit methods for partial differential equations, SIAM J. Numer. Anal. 32 (3) (1995) 797–823.
[30] M.J. Berger, I. Rigoutsos, An algorithm for point clustering and grid generation, IEEE Trans. Syst. Man. Cyb. 21 (5) (1991) 1278–1286.
[31] H.D. Ceniceros, A.M. Roma, Study of the long-time dynamics of a viscous vortex sheet with a fully adaptive non-stiff numerical method, Phys. Fluids. 16 (12) (2004) 4285–4318.
[32] W. Briggs, A Multigrid Tutorial, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1987.