Journal of Computational Physics 227 (2008) 9985–10017

Contents lists available at ScienceDirect

Journal of Computational Physics

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

A time-adaptive finite volume method for the Cahn–Hilliard and Kuramoto–Sivashinsky equations

Luis Cueto-Felgueroso *, Jaume Peraire

Aerospace Computational Design Laboratory, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, 77 Massachusetts Avenue, Cambridge, MA 02139, USA

ARTICLE INFO

Article history:
Received 25 February 2008
Received in revised form 19 June 2008
Accepted 16 July 2008
Available online 15 August 2008

Keywords:
High-order methods
Finite volume method
Fourth order equations
Cahn–Hilliard equation
Kuramoto–Sivashinsky equation
Adaptive time-stepping

ABSTRACT

This paper presents a complete finite volume method for the Cahn–Hilliard and Kuramoto–Sivashinsky type of equations. The spatial discretization is high-order accurate and suitable for general unstructured grids. The time integration is addressed by means of implicit an implicit–explicit fourth order Runge–Kutta schemes, with error control and adaptive time-stepping. The outcome is a practical, accurate and efficient simulation tool which has been successfully applied to accuracy tests and representative simulations.

The use of adaptive time-stepping is of paramount importance in problems governed by the Cahn–Hilliard model; an adaptive method may be several orders of magnitude more efficient than schemes using constant or heuristic time steps. In addition to driving the simulations efficiently, the time-adaptive procedure provides a *quantitative* (not just qualitative) characterization of the rich temporal scales present in phase separation processes governed by the Cahn–Hilliard phase-field model.

1. Introduction

Originally arisen in the context of phase separation after quenching of binary alloys [10], chemical phase turbulence [34] and flame front instabilities [43], the rich dynamical behaviour of the solutions of the Cahn–Hilliard and Kuramoto–Sivashinsky equations has inspired their use as a more general framework to model systems displaying complex spatiotemporal features, such as phase-ordering and coarsening [48,39,47], and pattern formation [42,18]. Under certain conditions, some of these equations may exhibit a chaotic behaviour, or *weak turbulence* [40,19,50,44]. From the simulation perspective, the challenge stems from the fact that non-trivial problems based on this type of equations are potentially very stiff and rich in spatial and temporal scales. The crucial consequence of this multiscale nature is that suitable numerical schemes must address the spatial and temporal discretizations specifically, since critical sources of stiffness may arise from either of them, thus compromising the accuracy and efficiency of the simulations.

For the Cahn–Hilliard equation, a number of finite element formulations have been proposed [20–22,6,7,23], in addition to finite difference schemes [46,25], spectral methods [28] and, more recently, discontinuous Galerkin schemes [49,12,51] and multigrid procedures [31,33]. The analysis of the Kuramoto–Sivashinsky equations has been mostly focused on the characterization of its dynamical properties, rather than on practical numerical simulation aspects, and thus most authors have used finite differences and spectral methods [40,19,42,18,50,44,1,30]. More recently, a discontinuous Galerkin method has been proposed for the one-dimensional KS equation [52]. To the authors' knowledge, there is no previous work on finite volume methods on general grids for these equations.

* Corresponding author.
E-mail address: lcueto@mit.edu (L. Cueto-Felgueroso).

0021-9991/$ - see front matter © 2008 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2008.07.024

This paper presents a *complete* numerical method for the Cahn–Hilliard and Kuramoto–Sivashinsky type of equations. These equations are nonlinear, fourth order PDE's, and their solutions inherently unsteady. The spatial discretization is carried out using a high-order finite volume method, suitable for general, unstructured grids. The time integration is addressed by means of implicit an implicit–explicit fourth order Runge–Kutta schemes, with error control and adaptive time-stepping. The outcome is a practical, accurate and efficient simulation tool, which has been successfully applied to convergence tests and representative simulations.

The proposed spatial discretization is an extension of a high-order unstructured-grid finite volume method for convection-dominated flows developed by the authors [15–17]. The key ingredient of this scheme is a high-order global approximation framework, constructed using Moving Least-Squares (MLS) [36,37], which provides a continuous representation of the solution. The reconstructed solution possesses high regularity, which allows a simple and efficient discretization of nonlinear equations with high-order terms.

A suitable time integrator is required to address two distinctive sources of stiffness in the numerical solution of these equations. Firstly, the presence of high-order terms, and in particular the fourth order ones, which induce a very severe $O(\Delta x^4)$ stability restriction over the explicit time step, thus precluding the use of explicit integrators for representative simulations. Secondly, phase separation processes governed by the Cahn–Hilliard model include another challenging source of stiffness, which stems from their complex dynamical behaviour, and in particular from the fact that the solutions are very rich in temporal scales. Starting from a homogeneous, disordered state, and after an initial transient of intense phase separation, it is easy to identify cycles of short periods of time when the solution changes abruptly, followed by long periods of slow grain coarsening. The severity of this characteristic phenomenology varies with time and the number of spatial dimensions, and can be exacerbated by some model parameters.

The important practical consequence is the efficiency collapse of any time integration strategy that lacks quality *error control* and *time step adaptivity*. Surprisingly, error control has been almost absent in the literature about numerical methods for the Cahn–Hilliard equation. Most of the existing schemes address the stiffness induced by the high-order terms through the use of implicit or semi-implicit integrators but, to the extent of our knowledge, there is no previous work on a systematic approach for error control and adaptive time-stepping in this context. Actually, most schemes assume constant time steps or use heuristic procedures whereby the time step is modified a few times during the simulation. Some authors have proposed to use time integration methods with enhanced stability properties, which may allow larger time steps. This approach, on the other hand, compromises the accuracy and the efficiency of the scheme, due to the lack of error control and to the fact that very long time steps may lead to slow/no convergence of the Newton iterations. The use of adaptive time-stepping is of paramount importance in this context, since an adaptive method may be several orders of magnitude more efficient than other schemes using constant or heuristic time steps.

An interesting feature of the proposed spatial discretization is that the semi-discretized problem reduces to a system of ODE's, and therefore the application of standard implicit an implicit–explicit Runge–Kutta integrators [8,32] is straightforward, which allows systematic and rigorous error control and adaptive time-stepping. In addition, the time-adaptive procedure provides a *quantitative* characterization of the different time scales present in phase separation processes.

The outline of the paper is as follows: Section 2 presents the one- and two-dimensional model equations. Section 3 is a general introduction to the proposed finite volume method, which is particularized for the Cahn–Hilliard and Kuramoto–Sivashinsky equations in Section 4. Section 5 is devoted to the time integration techniques used in this study, as well as to their practical implementation in the present context. Accuracy tests and "canonical" characterizations of the solutions are presented in Section 6, whereas Section 7 presents more representative, two-dimensional simulations. Finally, our main conclusions are drawn in Section 8.

2. Model equations

2.1. The Cahn–Hilliard equation

The Cahn–Hilliard equation was originally proposed as a model for phase separation after quenching of binary alloys [10], and nowadays constitutes a more general framework in the study of phase-ordering and coarsening phenomena. Transport of mass in a binary mixture seems to behave as if it was governed by the parabolic equation [10]

$$\frac{\partial c}{\partial t} = \nabla \cdot (B(c)\nabla(-\gamma \Delta c + \Psi'(c))) \tag{1}$$

where $c \in [0,1]$, or $c \in [-1,+1]$, depending on the interpretation given to the state variable, c. In the above equation, $B(c) \geqslant 0$ is the diffusion mobility, and γ is a positive constant. This problem statement is supplemented with suitable initial and boundary conditions. The model of Cahn and Hilliard is based on a Ginzburg–Landau free energy functional of the form

$$\mathcal{E}(c) = \int_\Omega \left(\frac{\gamma}{2}|\nabla c|^2 + \Psi(c)\right) \mathrm{d}\boldsymbol{x} \tag{2}$$

The first term in (2) penalizes large gradients and models capillary effects, while the second term is the homogeneous free energy. Two common expressions for Ψ are [10]

$$\Psi(c) = \frac{1}{4}(1-c)^2 c^2$$

$$\Psi(c) = \frac{\theta}{2}(c\ln c + (1-c)\ln(1-c)) + F_0(c) \tag{3}$$

with a smooth function $F_0(c)$; typically $F_0(c) = \theta_0 c(1-c)$. Thermodynamical considerations suggest mobilities of the form

$$B(c) = c(1-c) \tag{4}$$

The above definitions correspond to $c \in [0,1]$. Phenomenologically, the Cahn–Hilliard equation is a model for nucleation and growth of droplets of the minority phase in a binary mixture (Ostwald ripening), under cooling from a homogeneous, high-temperature state, to a point within the coexistence region [47].

2.2. Kuramoto–Sivashinsky-type equations

2.2.1. One-dimensional case

Our model problem in 1D is the Kuramoto–Sivashinsky equation

$$\frac{\partial u}{\partial t} + \frac{\partial}{\partial x}\left(\frac{1}{2}u^2\right) + \frac{\partial^2 u}{\partial x^2} + \nu\frac{\partial^4 u}{\partial x^4} = 0 \tag{5}$$

with suitable initial and boundary conditions. Arising in the context of chemical phase turbulence [34] and flame front instabilities [43], this equation has been used as a canonical model to understand the basic features of nonlinear systems displaying complex spatiotemporal features and pattern formation. For sufficiently large domains, the solutions of the above equation display chaotic behaviour, or *weak turbulence* [40,19,50,44].

2.2.2. Two-dimensional model problem

In two dimensions we focus on the damped Kuramoto–Sivashinsky equation

$$\frac{\partial u}{\partial t} + \nabla\cdot(\nabla u + \nabla(\Delta u)) = -\alpha u + |\nabla u|^2 \tag{6}$$

This equation has been studied as a model for systems characterized by the spontaneous appearance of coherent organized patterns (*cellular states*), when the initially unorganized system is driven away from thermodynamic equilibrium [42,18]. An important finding is the characterization of secondary instabilities that may destroy these organized structures, ultimately driving the system to spatiotemporal chaos [40,19,42,18].

3. Spatial discretization: a finite volume method based on multiresolution reproducing kernels

3.1. Overview

There has been extensive interest in the numerical simulation of systems governed by nonlinear, fourth order PDE's. For the Cahn–Hilliard equation, a number of finite element formulations have been proposed [20–22,6,7,23], in addition to finite difference schemes [46,25], spectral methods [28] and, more recently, discontinuous Galerkin schemes [49,12,51]. Advanced topics concerning spatial discretization, such as multigrid approaches, have also been analyzed [31,33]. The analysis of the Kuramoto–Sivashinsky equation has been mostly focused on the characterization of its dynamical properties, rather than on numerical simulation aspects, and thus most authors have used finite differences and spectral methods on simple domains [40,19,42,18,50,44,1,30]. More recently, a discontinuous Galerkin method has been proposed for the one-dimensional KS equation [52].

The following paragraphs introduce the proposed finite volume discretization, as well as the approximation technique used to reconstruct the solution. The specific application of this general formulation to the Cahn–Hilliard and Kuramoto–Sivashinsky equations is subsequently addressed.

The proposed spatial discretization is an extension of a high-order unstructured-grid finite volume method for convection-dominated flows developed by the authors [15–17]. The key ingredient of this scheme is a high-order global approximation framework, which provides a continuous representation of the solution. The reconstructed solution possesses high regularity, which allows a simple and efficient discretization of equations with high-order terms.

The usual approach of high-order finite volume schemes is pragmatic and *bottom-up*. Starting from an underlying piecewise constant representation, a discontinuous reconstruction of the field variables is performed at the cell level. An important practical consequence is that the discretization of higher order terms requires some kind of *recovery* procedure, which is, almost invariably, inconsistent with the aforementioned reconstruction. Our approach is somewhat the opposite. We start from a high-order and highly regular representation of the solution, obtained by means of Moving Least-Squares approximation [36,37], and well suited for general, unstructured grids. This approach is directly suitable for the discretization of elliptic/parabolic equations and high-order spatial terms. For equations with a predominantly hyperbolic character, the global representation is *broken* locally, at the cell level, into a piecewise polynomial reconstruction, which allows to use the powerful finite volume technology of Godunov-type schemes for hyperbolic problems (e.g. Riemann solvers, limiters) [16,17].

The presentation of the scheme will retain this general dual functional representation of the solution, which is fully practical for convection-dominated problems. In the present context, however, the equations to be solved include powerful dissipative mechanisms which, therefore, suggest the direct and *centered* reconstruction of the fluxes at the control volume edges. The proposed discretization of the Cahn–Hilliard and Kuramoto–Sivashinsky equations, which is presented in Section 4, follows this latter approach.

3.2. General formulation

Consider a system of conservation laws of the form

$$\frac{\partial \boldsymbol{u}}{\partial t} + \nabla \cdot (\mathscr{F}^H + \mathscr{F}^E) = \boldsymbol{S} \quad \text{in} \quad \Omega \tag{7}$$

supplemented with suitable initial and boundary conditions. The fluxes have been generically split into a hyperbolic-like part, $\mathscr{F}^H$, and an elliptic-like part, $\mathscr{F}^E$. Consider, in addition, a partition of the domain Ω into a set of non-overlapping *control volumes* or *cells*, $\mathcal{T}^h = I$. Furthermore, we define a reference point (*node*), $\boldsymbol{x}_I$ inside each cell (the cell centroid).

The spatial representation of the solution is as follows: consider a function $\boldsymbol{u}(\boldsymbol{x})$, given by its point values, $\boldsymbol{u}_I = \boldsymbol{u}(\boldsymbol{x}_I)$, at the cell centroids, with coordinates $\boldsymbol{x}_I$. The approximate function $\boldsymbol{u}^h(\boldsymbol{x})$ belongs to the subspace spanned by a set of *basis functions* $\{N_I(\boldsymbol{x})\}$ associated to the nodes, such that $\boldsymbol{u}^h(\boldsymbol{x})$ is given by

$$\boldsymbol{u}^h(\boldsymbol{x}) = \sum_{j=1}^{n_{\boldsymbol{x}}} N_j(\boldsymbol{x}) \boldsymbol{u}_j \tag{8}$$

which states that the approximation at a point $\boldsymbol{x}$ is computed using certain $n_{\boldsymbol{x}}$ surrounding nodes. This set of nodes is referred to as the *cloud* or *stencil* associated to the evaluation point $\boldsymbol{x}$. In particular, the above approximation is constructed using Moving Least-Squares (MLS) approximation [36,37]. An brief introduction to this technique will be presented below. Note that, using MLS, the approximate function $\boldsymbol{u}^h(\boldsymbol{x})$ is not a polynomial in general. An interesting feature of this MLS approach is the *centered* character of the approximation, thus avoiding the spatial bias which is often found in patch-based piecewise polynomial interpolations.

Consider now the integral form of the system of conservation laws (7) which, for each control volume I, reads

$$\int_{\Omega_I} \frac{\partial \boldsymbol{u}}{\partial t} \, \mathrm{d}\Omega + \int_{\Gamma_I} (\mathscr{F}^H + \mathscr{F}^E) \cdot \boldsymbol{n} \, \mathrm{d}\Gamma = \int_{\Omega_I} \boldsymbol{S} \, \mathrm{d}\Omega \tag{9}$$

Introducing the component-wise reconstructed function $\boldsymbol{u}^h$, the spatially discretized counterpart of (9) reads

$$\int_{\Omega_I} \frac{\partial \boldsymbol{u}^h}{\partial t} \, \mathrm{d}\Omega + \int_{\Gamma_I} (\mathscr{F}^{hH} + \mathscr{F}^{hE}) \cdot \boldsymbol{n} \, \mathrm{d}\Gamma = \int_{\Omega_I} \boldsymbol{S}^h \, \mathrm{d}\Omega \tag{10}$$

A direct evaluation of the fluxes in (10) is possible and efficient when the inherent dissipation mechanism is strong enough to overpower the convective terms. In convection-dominated problems, where the character of the equations is predominantly hyperbolic, this centered approach can lead to unstable computations. For this latter type of problems, we introduce a "*broken*" reconstruction, $\boldsymbol{u}_I^{hb}$, which approximates $\boldsymbol{u}^h(\boldsymbol{x})$ (and, therefore, $\boldsymbol{u}(\boldsymbol{x})$) locally inside each cell I, and is discontinuous across cell interfaces [17,16]. In general, we require the order of accuracy of the broken reconstruction to be the same as that of the original continuous reconstruction. One possible choice is to use Taylor series expansions; a quadratic reconstruction inside cell I, for example, would read

$$\boldsymbol{u}_I^{hb}(\boldsymbol{x}) = \boldsymbol{u}_I^h + \nabla \boldsymbol{u}_I^h \cdot (\boldsymbol{x} - \boldsymbol{x}_I) + \frac{1}{2} (\boldsymbol{x} - \boldsymbol{x}_I)^{\mathrm{T}} \boldsymbol{H}^h (\boldsymbol{x} - \boldsymbol{x}_I) \tag{11}$$

where the gradient $\nabla \boldsymbol{u}_I^h$ and the Hessian matrix $\boldsymbol{H}^h$ involve the successive derivatives of the continuous reconstruction $\boldsymbol{u}^h(\boldsymbol{x})$, which are evaluated at the cell centroids using MLS. This dual continuous/discontinuous reconstruction of the solution is crucial in order to obtain accurate and efficient numerical schemes for mixed parabolic/hyperbolic problems. The cell-wise broken reconstruction defined here is actually a piecewise continuous approximation to $\boldsymbol{u}^h$. The advantage is that it allows to make use of Riemann solvers, limiters, and other standard finite volume technologies, while keeping some consistency in terms of functional representation. Thus, the general continuous reconstruction is used to evaluate the viscous (elliptic-like) fluxes, whereas its discontinuous approximation is used to evaluate the inviscid (hyperbolic-like) fluxes.

The final semidiscrete scheme for the continuous/discontinuous approach can be written as

$$\int_{\Omega_I} \frac{\partial \boldsymbol{u}^h}{\partial t} \, \mathrm{d}\Omega + \int_{\Gamma_I} \boldsymbol{H}(\boldsymbol{u}^{hb+}, \boldsymbol{u}^{hb-}) \, \mathrm{d}\Gamma + \int_{\Gamma_I} \mathscr{F}^{hE} \cdot \boldsymbol{n} \, \mathrm{d}\Gamma = \int_{\Omega_I} \boldsymbol{S}^h \, \mathrm{d}\Omega \tag{12}$$

where $\boldsymbol{H}(\boldsymbol{u}^{hb+}, \boldsymbol{u}^{hb-})$ is a suitable numerical flux. Note that our dual reconstruction procedure induces a non-diagonal mass matrix. Most existing finite volume schemes recover, in principle, the diagonal structure of the mass matrix by enforcing reconstructions that preserve the mean. It is not clear whether this argument holds when elliptic terms are present, due to the "pragmatic" recovery approach. In our case, the structure of the consistent mass matrix is of the form $\boldsymbol{M} = \{m_{ij}\}$, where

$$m_{ij} = \int_{\Omega_i} N_j(\boldsymbol{x}) \mathrm{d}\Omega \tag{13}$$

An evaluation of the proposed scheme for convection-dominated problems can be found in [16,17].

3.3. Moving Least-Squares reproducing kernel approximations

This section presents a brief introduction to the theory and implementation of Moving Least-Squares (MLS) approximations. For a more detailed description of the algorithms, the reader is referred to classical references (e.g. [36,37]) and, within the present context, to previous works by the authors [15–17].

Moving Least-Squares approximate a function $u(\boldsymbol{x})$, at a given point $\boldsymbol{x}$ inside a domain Ω, through a weighted least-squares fitting of $u(\boldsymbol{x})$ in a neighborhood of $\boldsymbol{x}$, as

$$u(\boldsymbol{x}) \approx \hat{u}(\boldsymbol{x}) = \sum_{i=1}^{m} p_i(\boldsymbol{x}) \alpha_i(\boldsymbol{z})|_{\boldsymbol{z}=\boldsymbol{x}} = \boldsymbol{p}^{\mathrm{T}}(\boldsymbol{x}) \boldsymbol{\alpha}(\boldsymbol{z})|_{\boldsymbol{z}=\boldsymbol{x}} \tag{14}$$

where $\boldsymbol{p}^{\mathrm{T}}(\boldsymbol{x})$ is an m-dimensional basis of functions (usually polynomials), and $\boldsymbol{\alpha}(\boldsymbol{z})|_{\boldsymbol{z}=\boldsymbol{x}}$ is a set of parameters to be determined, and such that they minimize the error functional

$$J(\boldsymbol{\alpha}(\boldsymbol{z})|_{\boldsymbol{z}=\boldsymbol{x}}) = \int_{\boldsymbol{y}\in\Omega_{\boldsymbol{x}}} W(\boldsymbol{z}-\boldsymbol{y},h)|_{\boldsymbol{z}=\boldsymbol{x}} [u(\boldsymbol{y}) - \boldsymbol{p}^{\mathrm{T}}(\boldsymbol{y}) \boldsymbol{\alpha}(\boldsymbol{z})|_{\boldsymbol{z}=\boldsymbol{x}}]^2 \, \mathrm{d}\Omega_{\boldsymbol{x}} \tag{15}$$

The *kernel* or *weighting function* $W(\boldsymbol{z}-\boldsymbol{y},h)|_{\boldsymbol{z}=\boldsymbol{x}}$, with compact support $\Omega_{\boldsymbol{x}}$ centered at $\boldsymbol{z}=\boldsymbol{x}$, plays an important role in the compactness and other characteristics of the approximation. Note that, even if all the basis functions in $\boldsymbol{p}^{\mathrm{T}}(\boldsymbol{x})$ are polynomials, *the reconstructed function* $\hat{u}(\boldsymbol{x})$ *is not a polynomial in general*.

An important parameter in the above approximation framework is the characteristic length of the reconstruction, the so-called *smoothing length*, h, which represents a certain characteristic measure of the size of the support $\Omega_{\boldsymbol{x}}$ (e.g. kernels with circular supports of radius $2h$). We use the cubic spline

$$W(\boldsymbol{x}-\boldsymbol{y},h) = \begin{cases} 1 - \frac{3}{2}s^2 + \frac{3}{4}s^3 & s \leqslant 1 \\ \frac{1}{4}(2-s)^3 & 1 < s \leqslant 2 \\ 0 & s > 2 \end{cases} \tag{16}$$

where $s = \frac{|\boldsymbol{x}-\boldsymbol{y}|}{h}$. The minimization of J with respect to the set of parameters $\boldsymbol{\alpha}$ leads to the expression

$$\int_{\boldsymbol{y}\in\Omega_{\boldsymbol{x}}} \boldsymbol{p}(\boldsymbol{y}) W(\boldsymbol{z}-\boldsymbol{y},h)|_{\boldsymbol{z}=\boldsymbol{x}} u(\boldsymbol{y}) \mathrm{d}\Omega_{\boldsymbol{x}} = \boldsymbol{M}(\boldsymbol{x}) \boldsymbol{\alpha}(\boldsymbol{z})|_{\boldsymbol{z}=\boldsymbol{x}} \tag{17}$$

where the *moment matrix* $\boldsymbol{M}(\boldsymbol{x})$ is defined as

$$\boldsymbol{M}(\boldsymbol{x}) = \int_{\boldsymbol{y}\in\Omega_{\boldsymbol{x}}} \boldsymbol{p}(\boldsymbol{y}) W(\boldsymbol{z}-\boldsymbol{y},h)|_{\boldsymbol{z}=\boldsymbol{x}} \boldsymbol{p}^{\mathrm{T}}(\boldsymbol{y}) \mathrm{d}\Omega_{\boldsymbol{x}} \tag{18}$$

At the discrete level, the pointwise value of $u(\boldsymbol{x})$ is given at a number of scattered locations in Ω, which are often referred to as *particles* or *nodes*. In order to derive a practical approximation method, the above integrals are evaluated using nodal integration and, given the compact support of the kernel, only those nodes inside $\Omega_{\boldsymbol{x}}$ are involved as quadrature points. Thus, the set of parameters $\boldsymbol{\alpha}$ that minimize the functional J are given by

$$\boldsymbol{\alpha}(\boldsymbol{z})|_{\boldsymbol{z}=\boldsymbol{x}} = \boldsymbol{M}^{-1}(\boldsymbol{x}) \boldsymbol{P}_{\Omega_{\boldsymbol{x}}} \boldsymbol{W}(\boldsymbol{x}) \boldsymbol{u}_{\Omega_{\boldsymbol{x}}} \tag{19}$$

where the vector $\boldsymbol{u}_{\Omega_{\boldsymbol{x}}}$ contains the pointwise values of the function to be reproduced, $u(\boldsymbol{x})$, at the $n_{\boldsymbol{x}}$ particles inside $\Omega_{\boldsymbol{x}}$

$$\boldsymbol{u}_{\Omega_{\boldsymbol{x}}} = \begin{pmatrix} u(\boldsymbol{x}_1) & u(\boldsymbol{x}_2) & \dots & u(\boldsymbol{x}_{n_{\boldsymbol{x}}}) \end{pmatrix}^{\mathrm{T}} \tag{20}$$

The moment matrix, $\mathbf{M}$, which is an $(m \times m)$ matrix, is given by $\mathbf{M}(\boldsymbol{x}) = \mathbf{P}_{\Omega_{\boldsymbol{x}}} \mathbf{W}(\boldsymbol{x}) \mathbf{P}_{\Omega_{\boldsymbol{x}}}^{\mathrm{T}}$, and the matrices $\mathbf{P}_{\Omega_{\boldsymbol{x}}}$ and $\mathbf{W}(\boldsymbol{x})$, whose dimensions are, respectively, $(m \times n_{\boldsymbol{x}})$ and $(n_{\boldsymbol{x}} \times n_{\boldsymbol{x}})$, can be obtained as

$$\boldsymbol{P}_{\Omega_{\boldsymbol{x}}} = \begin{pmatrix} \boldsymbol{p}(\boldsymbol{x}_1) & \boldsymbol{p}(\boldsymbol{x}_2) & \dots & \boldsymbol{p}(\boldsymbol{x}_{n_{\boldsymbol{x}}}) \end{pmatrix} \tag{21}$$

$$\mathbf{W}(\boldsymbol{x}) = \mathrm{diag}\{W_i(\boldsymbol{x}-\boldsymbol{x}_i)\}, \quad i = 1, \dots, n_{\boldsymbol{x}} \tag{22}$$

Introducing (19) in (14), the interpolation structure can be identified as

$$\hat{u}(\boldsymbol{x}) = \sum_{j=1}^{n_{\boldsymbol{x}}} N_j(\boldsymbol{x}) u_j = \boldsymbol{N}^{\mathrm{T}}(\boldsymbol{x}) \boldsymbol{u}_{\Omega_{\boldsymbol{x}}} = \boldsymbol{p}^{\mathrm{T}}(\boldsymbol{x}) \boldsymbol{M}^{-1}(\boldsymbol{x}) \boldsymbol{P}_{\Omega_{\boldsymbol{x}}} \boldsymbol{W}(\boldsymbol{x}) \boldsymbol{u}_{\Omega_{\boldsymbol{x}}} \tag{23}$$

In analogy to finite elements, the approximation was written in terms of the MLS "shape functions"

$$\boldsymbol{N}^{\mathrm{T}}(\boldsymbol{x}) = \boldsymbol{p}^{\mathrm{T}}(\boldsymbol{x}) \boldsymbol{M}^{-1}(\boldsymbol{x}) \boldsymbol{P}_{\Omega_{\boldsymbol{x}}} \boldsymbol{W}(\boldsymbol{x}) \tag{24}$$

where $N_j(\boldsymbol{x})$ can be seen as the shape function associated to node j.

The basis of functions used in this study comprise scaled and *locally* defined monomials. Thus, in order to reconstruct a one-dimensional function at a location x_I, we use basis of the form

$$\boldsymbol{p}(\boldsymbol{x}) = \left(1 \quad \tfrac{x-x_I}{h} \quad \left(\tfrac{x-x_I}{h}\right)^2 \quad \dots \quad \left(\tfrac{x-x_I}{h}\right)^p\right)^{\mathrm{T}} \tag{25}$$

The functional basis $\boldsymbol{p}(\boldsymbol{x})$ is strongly related to the accuracy of the MLS fit. For a pth order MLS fit (pth order complete polynomial basis) and general, irregularly spaced points, the nominal order of accuracy for the approximation of an sth order gradient is roughly $(p-s+1)$. In general, any linear combination of the functions included in the basis is exactly reproduced by the MLS approximation. In multidimensions we follow the same idea of p-complete basis, constructed using products of *scaled* and *locally defined* monomials. The resulting MLS shape functions read

$$\boldsymbol{N}^{\mathrm{T}}(\boldsymbol{x}_I) = \boldsymbol{p}^{\mathrm{T}}(\boldsymbol{0})\boldsymbol{C}(\boldsymbol{x}_I) = \boldsymbol{p}^{\mathrm{T}}(\boldsymbol{0})\boldsymbol{M}^{-1}(\boldsymbol{x}_I)\boldsymbol{P}_{\Omega_{\boldsymbol{x}_I}}\boldsymbol{W}(\boldsymbol{x}_I) \tag{26}$$

where $\boldsymbol{C}(\boldsymbol{x})$ is defined as

$$\boldsymbol{C}(\boldsymbol{x}) = \boldsymbol{M}^{-1}(\boldsymbol{x})\boldsymbol{P}_{\Omega_{\boldsymbol{x}}}\boldsymbol{W}(\boldsymbol{x}) \tag{27}$$

The approximate derivatives of $u(\boldsymbol{x})$ can be expressed in terms of the derivatives of the MLS shape functions, which are functions of the derivatives of the polynomial basis $\boldsymbol{p}(\frac{\boldsymbol{x}-\boldsymbol{x}_I}{h})$ and the derivatives of $\boldsymbol{C}(\boldsymbol{x})$ [15–17]. For example, the first and second order derivatives of $u(\boldsymbol{x})$, evaluated at $\boldsymbol{x}_I$, are given by

$$\begin{aligned}
\left.\frac{\partial u(\boldsymbol{x})}{\partial x_\alpha}\right|_{\boldsymbol{x}=\boldsymbol{x}_I} &\approx \sum_{j=1}^{n_{\boldsymbol{x}_I}} u_j \left.\frac{\partial N_j(\boldsymbol{x})}{\partial x_\alpha}\right|_{\boldsymbol{x}=\boldsymbol{x}_I} \\
\left.\frac{\partial^2 u(\boldsymbol{x})}{\partial x_\alpha \partial x_\beta}\right|_{\boldsymbol{x}=\boldsymbol{x}_I} &\approx \sum_{j=1}^{n_{\boldsymbol{x}_I}} u_j \left.\frac{\partial^2 N_j(\boldsymbol{x})}{\partial x_\alpha \partial x_\beta}\right|_{\boldsymbol{x}=\boldsymbol{x}_I}
\end{aligned} \tag{28}$$

The MLS shape functions are data independent and, therefore, for fixed grids they need to be computed *only once* at the preprocessing phase. In order to prevent the moment matrix $\boldsymbol{M}$ from being singular or ill-conditioned, the cloud of neighbors must fulfill certain "good neighborhood" requirements. The definition of the cloud (the *MLS stencil*) for each evaluation point is a crucial issue that requires careful attention. The selection process must be suitable for general unstructured grids, and the stencil should be as compact as possible for the sake of computational efficiency and physical meaning. Note that these stencils are typically *centered* around the node, and thus the MLS approximation avoids the spatial bias which is often found in patch-based piecewise polynomial approximations.

Once the cloud of neighbor centroids has been determined, the smoothing length h for isotropic kernels (radial weighting) is set to be proportional to the maximum distance between the evaluation point $\boldsymbol{x}_I$ and its neighbors, as

$$h = k \max(\|\boldsymbol{x}_j - \boldsymbol{x}_I\|) \tag{29}$$

where typically $k \approx 0.6$ (recall that, using radial weighting, the support of the kernel expands over a circle of radius $2h$). For anisotropic or highly irregular grids it is more convenient to use anisotropic (tensor-product) kernels [17].

3.4. Stencils

The definition of the clouds or approximation stencils for the different evaluation points is probably the most prominent practical issue in the implementation of the proposed scheme, as it has a direct impact on the *full stencil* of the finite volume method; i.e. how many couplings a given cell will have or, from a more practical perspective, the number of non-zero entries in each row of the Jacobian matrix.

We shall consider two aspects of the stencil definition: firstly, the *MLS stencils*, or which neighbor nodes are used to compute the MLS shape functions at a given location (centroid or edge quadrature point); and, secondly, the full stencil of the finite volume scheme which, in the present context, comprises the union of the stencils associated to all the quadrature points on a given control volume.

The one-dimensional case does not require special attention. Whenever a cloud of n nodes is mentioned, we refer to the n nearest nodes. In multidimensions the choice is not straightforward, and the most reasonable strategy is to design the stencil for each MLS order p separately. In the examples presented below we use *cubic* approximation, and therefore the present analysis is restricted to $p = 3$. The trivial choice is to look for the n nearest neighbors, with n is slightly higher than the minimum to construct a p-complete polynomial basis (10 for $p = 3$ in 2D). Another option, and the one chosen here, is to construct the clouds using grid-based vicinity arguments.

3.4.1. $p = 3$ MLS stencils: I. Centroids

Fig. 1 presents the stencil used to compute the $p = 3$ MLS shape functions at the cell centroids. For and interior cell I, the stencil comprises its first and second neighbors (by neighbors we mean cells that share an edge). This gives a 13-point stencil. For boundary cells the stencil comprises those cells that share a vertex with the cell and their first neighbors. A stronger

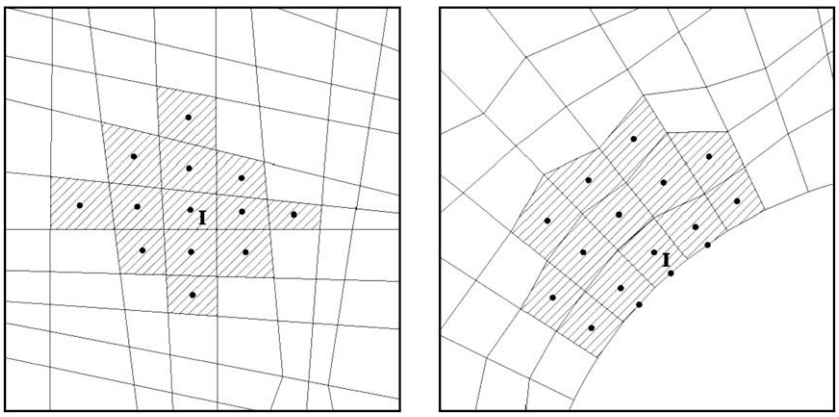

Fig. 1. $p = 3$ MLS stencil: centroids.

enforcement of the boundary conditions is achieved through the introduction of a set of "zero area" cells attached to the boundary (an approach analogous to the use of so-called *ghost cells*). Note that the centroids of these boundary cells, i.e. the midpoints of those edges lying on the boundary, have been included in the above stencils. During the simulation, the variables at these locations will be either extrapolated or assigned a certain value, depending on the type of boundary condition to be enforced.

3.4.2. $p = 3$ MLS stencils: II. Edges

Fig. 2 presents the stencil to compute the $p = 3$ MLS shape functions at the edge quadrature points. Given a quadrature point lying in the interface between cells A and B, its stencil comprises those cells sharing the extremum vertices of the edge,

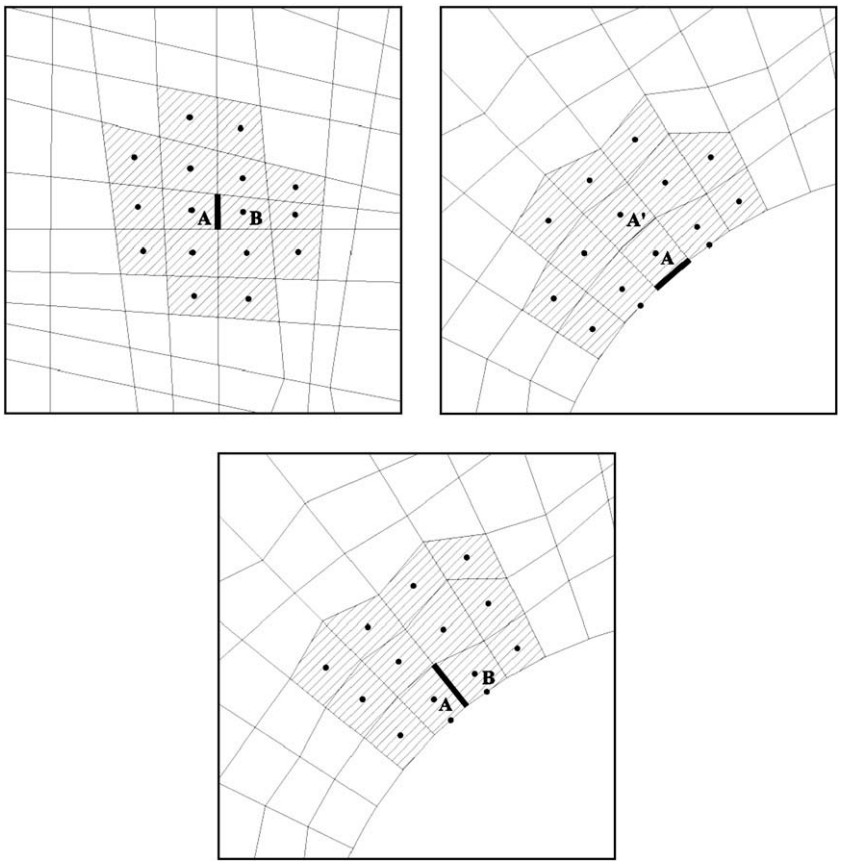

Fig. 2. $p = 3$ MLS stencil: quadrature points on edges.

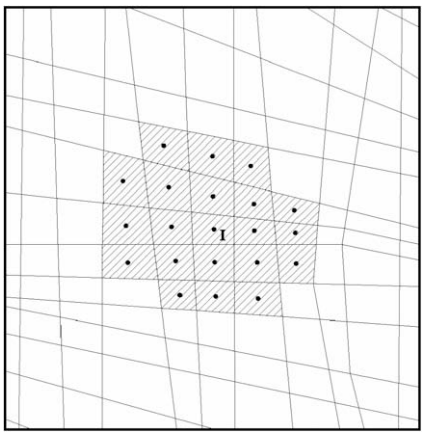

Fig. 3. Full stencil of the finite volume scheme (MLS order $p = 3$).

and their first neighbors. If both vertices are shared by four cells, this is a 16-point stencil. For boundary cells we also include the neighbors of the edge opposite to the boundary, and the corresponding ghost cells.

3.4.3. Full stencil of the finite volume scheme

The proposed finite volume scheme for the Cahn–Hilliard and Kuramoto–Sivashinsky equations introduces couplings only through terms which are computed using quantities evaluated directly at the edge quadrature points. Therefore, the full stencil of the discretization is obtained as the union of the MLS stencils associated to all the edges of cell I. Fig. 3 depicts the $p = 3$ stencil for interior cells, which comprises 21 cells. Note that, quite the opposite to what is usually thought about finite volume schemes, this stencil is actually quite compact for a cubic reconstruction.

3.5. Accuracy of the finite volume scheme

The MLS technique will be used to approximate the successive derivatives of the solution at the edges of the control volumes. Therefore, we expect that, for fluxes involving derivatives up to nth order, and using MLS approximations or order p (order of accuracy $p + 1$), the truncation error of the resulting finite volume scheme will be of order $(p + 1) - (n - 1)$. Thus, in the present context (fourth order equations and third order fluxes, $n = 3$), the finite volume schemes are expected to be of order $p - 1$. These convergence estimates will be checked and confirmed numerically in Section 6.

4. Proposed finite volume discretization of the Cahn–Hilliard and Kuramoto–Sivashinsky type of equations

4.1. Cahn–Hilliard equation

The integral form of the conservation law (1) reads, for each control volume I,

$$\int_{\Omega_I} \frac{\partial c}{\partial t}\,\mathrm{d}\Omega - \int_{\Gamma_I} B(c)\nabla(-\gamma\,\Delta c + \Psi'(c))\cdot\boldsymbol{n}\,\mathrm{d}\Gamma = 0 \tag{30}$$

Making use of the general MLS approximation

$$c^h(\boldsymbol{x}) = \sum_{j=1}^{n_{\boldsymbol{x}}} N_j(\boldsymbol{x})c_j \tag{31}$$

the semidiscrete version of (30) reads

$$\int_{\Omega_I} \frac{\partial c^h}{\partial t}\,\mathrm{d}\Omega - \int_{\Gamma_I} B(c^h)\nabla(-\gamma\,\Delta c^h + \Psi'(c^h))\cdot\boldsymbol{n}\,\mathrm{d}\Gamma = 0 \tag{32}$$

which can be compactly written as a system of ODE's of the form

$$\boldsymbol{M}\frac{\mathrm{d}\boldsymbol{c}}{\mathrm{d}t} + \boldsymbol{R}^h = 0 \tag{33}$$

where the components of the residual (associated to each cell i), R_i^h, which arise from the discretization of the line integral in (32), are given by

$$R_i^h = \sum_{\text{ie}}^{\text{ne}} \sum_{\text{ig}}^{\text{ng}} \boldsymbol{f}_{\text{ig}}^{h} \cdot \boldsymbol{n}_{\text{ig}} \omega_{\text{ig}} \tag{34}$$

in terms of the discrete fluxes

$$\boldsymbol{f}_{\text{ig}}^{h} \cdot \boldsymbol{n}_{\text{ig}} = -[B(c^h)\nabla(-\gamma \Delta c^h + \Psi'(c^h))]_{\text{ig}} \cdot \boldsymbol{n}_{\text{ig}} \tag{35}$$

In the above expressions, ne denotes the number of edges of the control volume, and ng is the number of edge quadrature points, with weights $\{\omega_{\text{ig}},\ \text{ig} = 1, \ldots, \text{ng}\}$. Due to the presence of the third order derivatives of the solution, the challenge for any unstructured-grid finite volume scheme is the approximation of the flux $\boldsymbol{f} \cdot \boldsymbol{n}$ at each quadrature point. This is precisely the crucial feature of our scheme. Using MLS approximation we can compute the numerical fluxes uniquely, and accurately, at the quadrature points, within a quite general geometrical setting which is well suited for unstructured meshes.

Considering a constant value of γ, the two-dimensional discrete flux is given by

$$\boldsymbol{f}^{h} \cdot \boldsymbol{n} = -B(c^h) \begin{pmatrix} -\gamma\left(\frac{\partial^3 c^h}{\partial x^3} + \frac{\partial^3 c^h}{\partial y^2 \partial x}\right) + \frac{\partial \Psi'(c^h)}{\partial x} \\ -\gamma\left(\frac{\partial^3 c^h}{\partial x^2 \partial y} + \frac{\partial^3 c^h}{\partial y^3}\right) + \frac{\partial \Psi'(c^h)}{\partial y} \end{pmatrix} \cdot \boldsymbol{n} \tag{36}$$

where the h superscript denotes MLS-approximated value, in the sense provided by (31). Recall that, for a given quadrature point ig, the general expression (31) particularizes as

$$c_{\text{ig}} = \sum_{j=1}^{n} N_j(\boldsymbol{x}_{\text{ig}}) c_j \tag{37}$$

where the summation extends over the *stencil* associated to the quadrature point, which has coordinates $\boldsymbol{x}_{\text{ig}}$. The discrete mobilities in (36) are directly computed as $B_{\text{ig}} = B(c_{\text{ig}})$. It follows from the MLS approximation (31) that the approximate third order derivatives of the concentrations can be expressed in terms of the third order derivatives of the MLS shape functions, as

$$\left.\frac{\partial^3 c^h}{\partial x^a \partial y^b}\right|_{\text{ig}} = \sum_{j=1}^{n} \left.\frac{\partial^3 N_j}{\partial x^a \partial y^b}\right|_{\text{ig}} c_j, \quad a + b = 3,\ a \geqslant 0,\ b \geqslant 0 \tag{38}$$

We analyzed two different formulations for the free energy terms, $\nabla \Psi'(c^h)$. The first one, which is *consistent* with a weighted residuals formulation, stems from the fact that

$$\nabla \Psi'(c) = \Psi''(c) \nabla c \tag{39}$$

Thus, introducing the approximate concentrations c^h in the above expression, a suitable consistent discretization reads

$$\nabla \Psi'(c^h) = \Psi''(c^h) \nabla c^h \tag{40}$$

Another option is to use a group representation, whereby Ψ' is first computed at the nodes, and then its gradient is evaluated at the edge quadrature points, according to

$$\left.\frac{\partial \Psi'(c)}{\partial x}\right|_{\text{ig}} = \sum_{j=1}^{n} \left.\frac{\partial N_j}{\partial x}\right|_{\text{ig}} \Psi'(c_j), \quad \left.\frac{\partial \Psi'(c)}{\partial y}\right|_{\text{ig}} = \sum_{j=1}^{n} \left.\frac{\partial N_j}{\partial y}\right|_{\text{ig}} \Psi'(c_j) \tag{41}$$

which implies that we assume $\nabla \Psi'(c^h) \approx \nabla \Psi'^h(c)$. This fact induces an inconsistency with the weighted residuals formulation, due to the commutation error, but in practice this latter form yields slightly more accurate results.

The $O(\Delta x^4)$ time step stability limit precludes the use of explicit time-stepping for practical purposes. Fully implicit approaches, as the one adopted in this study, require the linearization of the fluxes (36) The use of MLS approximants, with its shape function structure, allows a simple definition of the Jacobian. In particular, the components of the Jacobian $\boldsymbol{J} = \{J_{ij}\}$ are given by

$$J_{ij} = \sum_{\text{ig}} \frac{\partial \boldsymbol{f}^{\text{ig}}}{\partial c_j} \cdot \boldsymbol{n}_{\text{ig}} \omega_{\text{ig}} \tag{42}$$

where the summation extends over all the quadrature points on all the edges of the control volume. For the consistent fluxes we have

$$\frac{\partial \boldsymbol{f}^{\text{ig}}}{\partial c_j} = -B(c_{\text{ig}}) \begin{pmatrix} -\gamma\left(\frac{\partial^3 N_j}{\partial x^3} + \frac{\partial^3 N_j}{\partial y^2 \partial x}\right) + \Psi''' N_j \frac{\partial c}{\partial x} + \Psi'' \frac{\partial N_j}{\partial x} \\ -\gamma\left(\frac{\partial^3 N_j}{\partial x^2 \partial y} + \frac{\partial^3 N_j}{\partial y^3}\right) + \Psi''' N_j \frac{\partial c}{\partial y} + \Psi'' \frac{\partial N_j}{\partial y} \end{pmatrix}_{\text{ig}} - B'(c_{\text{ig}}) N_j(\boldsymbol{x}_{\text{ig}}) \begin{pmatrix} -\gamma\left(\frac{\partial^3 c}{\partial x^3} + \frac{\partial^3 c}{\partial y^2 \partial x}\right) + \Psi'' \frac{\partial c}{\partial x} \\ -\gamma\left(\frac{\partial^3 c}{\partial x^2 \partial y} + \frac{\partial^3 c}{\partial y^3}\right) + \Psi'' \frac{\partial c}{\partial y} \end{pmatrix}_{\text{ig}} \tag{43}$$

whereas for the inconsistent ones,

$$\frac{\partial \boldsymbol{f}^{\text{ig}}}{\partial c_j} = -B(c_{\text{ig}}) \begin{pmatrix} -\gamma\left(\frac{\partial^3 N_j}{\partial x^3} + \frac{\partial^3 N_j}{\partial y^2 \partial x}\right) + \Psi''(c_j)\frac{\partial N_j}{\partial x} \\ -\gamma\left(\frac{\partial^3 N_j}{\partial x^2 \partial y} + \frac{\partial^3 N_j}{\partial y^3}\right) + \Psi''(c_j)\frac{\partial N_j}{\partial y} \end{pmatrix}_{\text{ig}} - B'(c_{\text{ig}}) N_j(\boldsymbol{x}_{\text{ig}}) \begin{pmatrix} -\gamma\left(\frac{\partial^3 c}{\partial x^3} + \frac{\partial^3 c}{\partial y^2 \partial x}\right) + \frac{\partial \Psi'(c)}{\partial x} \\ -\gamma\left(\frac{\partial^3 c}{\partial x^2 \partial y} + \frac{\partial^3 c}{\partial y^3}\right) + \frac{\partial \Psi'(c)}{\partial y} \end{pmatrix}_{\text{ig}} \tag{44}$$

It follows from the compact support of the MLS shape functions that the above derivatives $\frac{\partial \boldsymbol{f}^{\text{ig}}}{\partial c_j}$ are different from zero if, and only if, the node j belongs to the stencil of the quadrature point ig.

4.2. Kuramoto–Sivashinsky equation

4.2.1. One-dimensional model

Our model problem in 1D is the Kuramoto–Sivashinsky equation

$$\frac{\partial u}{\partial t} + \frac{\partial}{\partial x}\left(\frac{1}{2}u^2\right) + \frac{\partial^2 u}{\partial x^2} + \nu \frac{\partial^4 u}{\partial x^4} = 0, \quad x \in [a,b] \tag{45}$$

with suitable initial and boundary conditions. The finite volume framework stems from the integral form of the conservation law (45) over each control volume Ω_I

$$\int_{\Omega_I} \frac{\partial u}{\partial t}\,\mathrm{d}x + \int_{\Omega_I} \left\{ \frac{\partial}{\partial x}\left(\frac{1}{2}u^2\right) + \frac{\partial^2 u}{\partial x^2} + \nu \frac{\partial^4 u}{\partial x^4} \right\} \mathrm{d}x = 0 \tag{46}$$

Using the divergence theorem, the above expression is equivalent to

$$\int_{\Omega_I} \frac{\partial u}{\partial t}\,\mathrm{d}x + \int_{\Gamma_I} \left\{ \frac{1}{2}u^2 + \frac{\partial u}{\partial x} + \nu \frac{\partial^3 u}{\partial x^3} \right\} n\,\mathrm{d}x = 0 \tag{47}$$

where Γ_I is the boundary of Ω_I. The spatial discretization of (47) is again constructed by means of MLS reconstruction operators, as

$$u^h(x) = \sum_{j=1}^{n_x} N(x) u_j \tag{48}$$

Introducing the above discrete representation of the solution u^h into (47) yields

$$\int_{\Omega_I} \frac{\partial u^h}{\partial t}\,\mathrm{d}x + \int_{\Gamma_I} \left\{ \frac{1}{2}(u^h)^2 + \frac{\partial u^h}{\partial x} + \nu \frac{\partial^3 u^h}{\partial x^3} \right\} \mathrm{d}x = 0 \tag{49}$$

Thus, the semi-discretized finite volume equations read

$$\boldsymbol{M}\frac{\mathrm{d}\boldsymbol{u}}{\mathrm{d}t} + \boldsymbol{R}^h = 0 \tag{50}$$

where the components of the residual (associated to each control volume) can be written as

$$R_i^h = \left[\frac{1}{2}(u^h)^2 + \frac{\partial u^h}{\partial x} + \nu \frac{\partial^3 u^h}{\partial x^3} \right]_{i+\frac{1}{2}} - \left[\frac{1}{2}(u^h)^2 + \frac{\partial u^h}{\partial x} + \nu \frac{\partial^3 u^h}{\partial x^3} \right]_{i-\frac{1}{2}} \tag{51}$$

The discrete derivatives are once more computed in terms of the MLS shape functions and the cell values, as

$$\frac{\partial u^h}{\partial x} = \sum_{j=1}^{n} \frac{\partial N_j}{\partial x} u_j, \quad \frac{\partial^3 u^h}{\partial x^3} = \sum_{j=1}^{n} \frac{\partial^3 N_j}{\partial x^3} u_j \tag{52}$$

4.2.2. Two-dimensional model

Our model problem in two dimensions is the damped Kuramoto–Sivashinsky equation

$$\frac{\partial u}{\partial t} + \nabla \cdot (\nabla u + \nabla(\Delta u)) = -\alpha u + |\nabla u|^2 \tag{53}$$

which has been extensively analyzed as a paradigm in dissipative systems [42,18]. The integral form of (53) reads, after applying the divergence theorem,

$$\int_{\Omega_I} \frac{\partial u}{\partial t}\,\mathrm{d}\boldsymbol{x} + \int_{\Gamma_I} (\nabla u + \nabla(\Delta u)) \cdot \boldsymbol{n}\,\mathrm{d}\boldsymbol{s} = \int_{\Omega_I} (-\alpha u + |\nabla u|^2)\mathrm{d}\boldsymbol{x} \tag{54}$$

Making use again of the MLS approximation, the spatially discretized problem can be expressed as

$$\int_{\Omega_I} \frac{\partial u^h}{\partial t}\,\mathrm{d}\boldsymbol{x} + \int_{\Gamma_I} (\nabla u^h + \nabla(\Delta u^h)) \cdot \boldsymbol{n}\,\mathrm{d}\boldsymbol{s} = \int_{\Omega_I} (-\alpha u^h + |\nabla u^h|^2)\mathrm{d}\boldsymbol{x} \tag{55}$$

which, written in compact notation as a system of ODE's, reads

$$\boldsymbol{M}\frac{\mathrm{d}\boldsymbol{u}}{\mathrm{d}t} + \boldsymbol{R}^h = \boldsymbol{S}^h \tag{56}$$

where the terms arising from the line integrals, R_i^h, are given by

$$R_i^h = \sum_{\mathrm{ie}}^{\mathrm{ne}} \sum_{\mathrm{ig}}^{\mathrm{ng}} \begin{pmatrix} \frac{\partial^3 u^h}{\partial x^3} + \frac{\partial^3 u^h}{\partial y^2 \partial x} + \frac{\partial u^h}{\partial x} \\ \frac{\partial^3 u^h}{\partial x^2 \partial y} + \frac{\partial^3 u^h}{\partial y^3} + \frac{\partial u^h}{\partial y} \end{pmatrix}_{\mathrm{ig}} \cdot \boldsymbol{n}_{\mathrm{ig}} \omega_{\mathrm{ig}} \tag{57}$$

and a second order approximation to the source term can be readily computed as

$$S_i^h = (-\alpha u^h + |\nabla u^h|^2)_i A_i \tag{58}$$

where A_i is the area of the control volume. For higher order schemes, a more accurate quadrature rule is required.

5. Time integration: Runge–Kutta pairs and error control for stiff ODE's

5.1. Overview

The numerical solution of the Cahn–Hilliard and Kuramoto–Sivashinsky type of equations leads to very stiff, unsteady problems. It is interesting to identify two distinctive sources of stiffness, which ultimately determine the characteristics and requirements of practical time integration approaches. Firstly, the presence of high-order terms, and in particular the fourth order ones, which induce a very severe $O(\Delta x^4)$ stability restriction over the explicit time step. This suggests (requires) the use of implicit or implicit–explicit time-stepping procedures, and precludes the use of explicit integrators for representative simulations. Somewhat included in this first source of stiffness is the *grid induced* stiffness, which arises from the presence of areas of intense grid clustering. To a wide extent, all the above can be addressed by means of more or less standard approaches for stiff ODE's (e.g. stiffly accurate implicit or semi-implicit schemes), with all their (real or potential) failures and successes.

Phase separation processes governed by the Cahn–Hilliard model include another challenging source of stiffness, which stems from their complex dynamical behaviour, and in particular from the fact that the solutions are extremely rich in temporal scales. Starting from a homogeneous, disordered state, and after an initial transient of intense phase separation, it is easy to identify cycles of short periods of time when the solution changes abruptly, followed by long periods of slow grain coarsening. The severity of this characteristic phenomenology varies with time and the number of spatial dimensions, and can be exacerbated by some model parameters.

The efficiency and accuracy of any numerical scheme for the Cahn–Hilliard equation can be seriously compromised if its multiscale nature is not properly addressed. Perhaps the most natural approach, and the one adopted in this study, is to rely on *error control* and *adaptive time step selection* techniques. Surprisingly, error control has been almost absent in the literature about numerical methods for the Cahn–Hilliard equation. Most of the existing schemes address the stiffness induced by the high-order terms thorough the use of implicit or semi-implicit integrators but, to the extent of our knowledge, the is no previous work on a systematic approach for error control and adaptive time-stepping in this context. Actually, most schemes assume constant time steps or use heuristic procedures whereby the time step is modified a few times during the simulation. Some authors have proposed to use time integration methods with enhanced stability properties, which may allow larger time steps. This approach, on the other hand, compromises the accuracy and the efficiency of the scheme, due to the lack of error control and to the fact that very long time steps may lead to slow/no convergence of the Newton iterations.

High-order Runge–Kutta methods are particularly interesting in this context. Their theoretical foundations are extensive and sound, allowing the design of stable, high-order accurate, "complete" schemes, in the sense that, in addition to good accuracy and stability properties, the designer also provides embedded methods (which allow for accurate error control and adaptive step size selection), dense output formulas, and stage-value predictors [32]. A key feature for our analysis is that, being multistage methods, they are naturally suited for variable step sizes.

The straightforward group of candidate schemes is that of *implicit* Runge–Kutta methods (IRK). For the sake of efficiency, storage and implementation, we may favor the *diagonally implicit* (DIRK) subset. Of particular interest is the case when the stiff terms are linear, which is the case of the Kuramoto–Sivashinsky equation. Under these conditions, it may be extremely advantageous to split the equations into its *stiff* and *nonstiff* components, and treat each of them separately. In particular, the nonlinear, nonstiff terms are integrated using a suitable *explicit* scheme, whereas the linear, stiff terms and integrated using and *implicit* scheme. The idea of *partitioned or additive Runge–Kutta schemes* (ARK) [13,14,32,38,3,11,5,9] appears in this context, giving rise to the class of *implicit–explicit* (IMEX) Runge–Kutta schemes. More precisely, when the stiff part is linear these methods are oftentimes referred to as *linearly implicit* Runge–Kutta schemes.

Multistep methods are also an obvious choice, both for the fully implicit and implicit–explicit cases [4,24]. In this study, we have only considered Runge–Kutta schemes due to their superior flexibility in terms of error and step size control. The following sections present an introduction to the aforementioned implicit and additive Runge–Kutta schemes, and their specific application to the Cahn–Hilliard and Kuramoto–Sivashinsky equations, respectively. Not that a class of multistep implicit–explicit BDF methods has already been proposed for the time integration of the KS equation [1].

5.2. General formulation

Let us consider a system of ODE's of the form

$$\frac{\mathrm{d}\boldsymbol{U}}{\mathrm{d}t} = \boldsymbol{f}(t, \boldsymbol{U}) \tag{59}$$

One step of an s-stage Runge–Kutta scheme can be written as

$$\boldsymbol{U}^{n+1} = \boldsymbol{U}^n + \Delta t \sum_{i=1}^{s} b_i \boldsymbol{k}_i, \quad \boldsymbol{k}_i = \boldsymbol{f}(t^n + c_i \Delta t, \boldsymbol{U}_i) \tag{60}$$

with stage values

$$\boldsymbol{U}_i = \boldsymbol{U}^n + \Delta t \sum_{j=1}^{s} a_{ij} \boldsymbol{k}_j \tag{61}$$

In the above expressions, Δt is the time step, and $\boldsymbol{A} = \{a_{ij}\} \in \Re^{s\times s}$, $\boldsymbol{b} \in \Re^s$ and $\boldsymbol{c} \in \Re^s$ are the characteristic coefficients of each given Runge–Kutta scheme, which can be compactly written using the so-called Butcher tableau

$$\begin{array}{c|cccc} c_1 & a_{11} & a_{12} & \cdots & a_{1s} \\ c_2 & a_{21} & a_{22} & \cdots & a_{2s} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ c_s & a_{s1} & a_{s2} & \cdots & a_{ss} \\ \hline & b_1 & b_2 & \cdots & b_s \\ \hline & \widehat{b}_1 & \widehat{b}_2 & \cdots & \widehat{b}_s \end{array}$$

The consistency vector $\boldsymbol{c}$ defines the points (in time) at which the method computes approximations to the initial value problem, so that the stage values can be seen as $\boldsymbol{U}_i \approx \boldsymbol{U}(t^n + c_i \Delta t)$. The row sum condition

$$c_i = \sum_{j=1}^{s} a_{ij} \quad \forall i = 1, \ldots, s \tag{62}$$

is usually adopted to simplify the order conditions for high-order methods. Explicit schemes are characterized by $\{a_{ij} = 0 \ \forall j \geqslant i\}$. The second set of coefficients $\{\hat{b}_i,\ i = 1, \ldots, s\}$ corresponds to the embedded scheme, which is used for error estimation.

5.3. Diagonally implicit schemes: application to the Cahn–Hilliard equation

Consider Runge–Kutta schemes with Butcher tableaux of the form [2,8]

$$\begin{array}{c|ccccc} 0 & 0 & 0 & 0 & \cdots & 0 \\ 2\gamma & \gamma & \gamma & 0 & \cdots & 0 \\ c_3 & a_{31} & a_{32} & \gamma & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & a_{s,1} & a_{s,2} & a_{s,3} & \cdots & \gamma \\ \hline & b_1 & b_2 & b_3 & \cdots & \gamma \\ \hline & \widehat{b}_1 & \widehat{b}_2 & \widehat{b}_3 & \cdots & \widehat{b}_s \end{array}$$

These schemes are referred to as *explicit first step, single diagonal coefficient, diagonally implicit Runge–Kutta* (ESDIRK) methods. Each stage value of an ESDIRK scheme is at least second-order accurate. The stiffly accurate assumption ($a_{s,j} = b_j$ $\forall j = 1,\ldots,s$) is frequently adopted, which extends A-stability into L-stability [8]. In this study we use the ESDIRK4 scheme presented in [8], which allows straightforward error control and adaptive time-stepping through the embedded scheme. The efficiency of the integrator for nonlinear problems can be enhanced by the use of good stage value predictors [26,29,35,41].

5.3.1. Implementation

The stage value computation in a DIRK scheme reads

$$\boldsymbol{U}_i = \boldsymbol{U}^n + \Delta t \sum_{j=1}^{i} a_{ij}\boldsymbol{k}_j \tag{63}$$

Given that the $i-1$ previous $\boldsymbol{k}$'s have been previously computed, (63) can be written as

$$\boldsymbol{U}_i = \boldsymbol{E}_i + \Delta t a_{ii}\boldsymbol{k}_i, \quad \boldsymbol{E}_i = \boldsymbol{U}^n + \Delta t \sum_{j=1}^{i-1} a_{ij}\boldsymbol{k}_j \tag{64}$$

The above equation is, in general, a nonlinear system of equations. The $p+1$ Newton iteration associated to (64) is given by

$$\left(\boldsymbol{I} - \Delta t a_{ii}\frac{\partial \boldsymbol{k}_i}{\partial \boldsymbol{U}}\bigg|^p\right)\Delta^p \boldsymbol{U}_i = \boldsymbol{E}_i + \Delta t a_{ii}\boldsymbol{k}_i^p \tag{65}$$

where $\Delta^p \boldsymbol{U}_i = \boldsymbol{U}_i^{p+1} - \boldsymbol{U}_i^p$. As mentioned above, the convergence of the Newton iterations may be improved by the use of good stage value predictors.

The embedded scheme uses the same raw information as the original one, but in this case it is "processed" using the second set of weights, $\{\hat{b}_i,\ i = 1,\ldots,s\}$. Thus, at the end of each step of the RK integrator, we have

$$\boldsymbol{U}^{n+1} = \boldsymbol{U}^n + \Delta t \sum_{i=1}^{s} b_i \boldsymbol{k}_i, \quad \widehat{\boldsymbol{U}}^{n+1} = \boldsymbol{U}^n + \Delta t \sum_{i=1}^{s} \hat{b}_i \boldsymbol{k}_i \tag{66}$$

and the error estimate is given by some suitable norm of the difference between these two solutions, $r^{n+1} = \|\boldsymbol{U}^{n+1} - \widehat{\boldsymbol{U}}^{n+1}\|$.

This class of implicit methods has been chosen in this study for the time integration of the Cahn–Hilliard equation. One may think about the possibility of using semi-implicit techniques (as the one exposed in the next section), due to the fact that the higher order term in the equations (the fourth order one) is linear. However, the nonlinear term is a second order one and, in most circumstances, this term poses too stringent stability limitations to be integrated efficiently using explicit schemes. At least for the examples analyzed in this study, the fully implicit approach proved to be much more efficient than implicit–explicit integrators.

The application of a DIRK scheme to the Cahn–Hilliard equation, within the proposed finite volume framework, corresponds to identifying $\boldsymbol{k} = -\boldsymbol{R}^h$, where $\boldsymbol{R}^h = \{R_I^h\}$ was already defined as

$$R_I^h = \sum_{\mathrm{ig}}^{\mathrm{ng}} B(c_{\mathrm{ig}}) \begin{pmatrix} -\gamma\left(\frac{\partial^3 c^h}{\partial x^3} + \frac{\partial^3 c^h}{\partial y^2 \partial x}\right) + \frac{\partial \Psi'(c^h)}{\partial x} \\ -\gamma\left(\frac{\partial^3 c^h}{\partial x^2 \partial y} + \frac{\partial^3 c^h}{\partial y^3}\right) + \frac{\partial \Psi'(c^h)}{\partial y} \end{pmatrix}_{\mathrm{ig}} \cdot \boldsymbol{n}_{\mathrm{ig}} \omega_{\mathrm{ig}} \tag{67}$$

The above summation extends over all the quadrature points on all the edges of the control volume I. On the other hand, the Jacobian matrix $\frac{\partial \boldsymbol{k}}{\partial \boldsymbol{U}}$ is given by (44) and (45).

5.4. Additive Runge–Kutta schemes: application to the Kuramoto–Sivashinsky equation

Consider systems of ordinary differential equations (59) that can be written in additive form as [5]

$$\frac{\mathrm{d}\boldsymbol{U}}{\mathrm{d}t} = \sum_{v=1}^{N} \boldsymbol{f}^{[v]}(t, \boldsymbol{U}) \tag{68}$$

where $\boldsymbol{f}^{[1]}, \boldsymbol{f}^{[2]}, \ldots, \boldsymbol{f}^{[N]}$ denote certain terms or *components* of $\boldsymbol{f}$, whose distinctive properties are worth being taken into account separately. The above expression (68) is in principle quite loose in terms of the considerations that lead to such splitting. In general, it may be advantageous to exploit the additive structure of the system (68) when either $\boldsymbol{f}$ or the unknowns $\boldsymbol{U}$ themselves present components with significantly different time scales. In the present context, in which our intention is the discretization of PDE's, typical situations of the former case are stiff–nonstiff *a priori* decompositions of the equations, whereas the latter usually refers to grid-induced stiffness. The idea behind additive schemes is to use for each component the integrator that best suits its particular characteristics.

Let us start with the general, N-component case. The integration of (68) can be carried out through the application of N different Runge–Kutta methods, one for each of the components. A step of an s-stage, N-part additive (ARK_N) or partitioned (PRK_N) Runge–Kutta scheme, defined by its generalized Butcher tableau

$$
\begin{array}{c|ccc|c|ccc}
c_1 & a_{11}^{[1]} & \cdots & a_{1s}^{[1]} & \cdots & a_{11}^{[N]} & \cdots & a_{1s}^{[N]} \\
\vdots & \vdots & \ddots & \vdots & \cdots & \vdots & \ddots & \vdots \\
c_s & a_{s1}^{[1]} & \cdots & a_{ss}^{[1]} & \cdots & a_{s1}^{[N]} & \cdots & a_{ss}^{[N]} \\
\hline
 & b_1^{[1]} & \cdots & b_s^{[1]} & \cdots & b_1^{[N]} & \cdots & b_s^{[N]} \\
\hline
 & \widehat{b}_1^{[1]} & \cdots & \widehat{b}_s^{[1]} & \cdots & \widehat{b}_1^{[N]} & \cdots & \widehat{b}_s^{[N]}
\end{array}
$$

is given by

$$\boldsymbol{U}^{n+1} = \boldsymbol{U}^n + \Delta t \sum_{i=1}^{s} \sum_{v=1}^{N} b_i^{[v]} \boldsymbol{f}^{[v]}(t + c_i \Delta t, \boldsymbol{U}_i) \tag{69}$$

with stage values

$$\boldsymbol{U}_i = \boldsymbol{U}^n + \Delta t \sum_{j=1}^{s} \sum_{v=1}^{N} a_{ij}^{[v]} \boldsymbol{f}^{[v]}(t + c_j \Delta t, \boldsymbol{U}_j) \tag{70}$$

where

$$c_j = \sum_{k=1}^{s} a_{jk}^{[v]} \quad \forall v = 1, \ldots, N \tag{71}$$

The Butcher coefficients $\{a_{ij}^{[v]}\}$, $\{b_i^{[v]}\}$, $\{\hat{b}_i^{[v]}\}$, $v = 1, \ldots, N$ and $\{c_i\}$ are constrained by certain accuracy and stability requirements. The order conditions of the combined scheme include those specific to each elemental method, and also certain *coupling* conditions. The growth of the number of coupling conditions for increasingly higher order and number of components N is such that the practical design of ARK_N methods has been typically restricted to $N = 2$. Following this ARK_2 approach, the system (68) can be conceptually written as

$$\frac{\mathrm{d}\boldsymbol{U}}{\mathrm{d}t} = \boldsymbol{f}_{\mathrm{s}}(t, \boldsymbol{U}) + \boldsymbol{f}_{\mathrm{ns}}(t, \boldsymbol{U}) \tag{72}$$

where the right hand side of (68) has been generically split into *stiff* ($\boldsymbol{f}_{\mathrm{s}}$) and *nonstiff* ($\boldsymbol{f}_{\mathrm{ns}}$) terms. Two different Runge–Kutta schemes, specifically designed and coupled, are applied to each term, and the important case in our context is the *implicit–explicit* (IMEX) approach, which acknowledges the fact that the stiff part is more efficiently dealt with by means of an *implicit* integrator, whereas the nonstiff part can be straightforwardly integrated using an *explicit* scheme. In particular, many problems of practical interest are modeled by partial differential equations whose semidiscretization can be expressed in the form of (72), where $\boldsymbol{f}_{\mathrm{s}}(t, \boldsymbol{U})$ is linear but stiff, and $\boldsymbol{f}_{\mathrm{ns}}(t, \boldsymbol{U})$ is nonlinear but nonstiff. The resulting system of ODE's can be very efficiently integrated following the *IMEX* approach. The combined integrators are referred to as *IMEX* ARK_2 methods or, when the stiff terms are linear, *linearly implicit* Runge–Kutta schemes. In our context, the Kuramoto–Sivashinsky equation falls into this category, and therefore linearly implicit integrators are necessarily among the most efficient choices.

There are two main families of *IMEX* ARK_2 schemes, depending on the form of the implicit integrator. The method chosen in this study is one of the schemes introduced by Kennedy and Carpenter [32], which are constructed using stiffly accurate ESDIRK methods, thus taking the form

$$
\begin{array}{c|cccccc}
0 & 0 & 0 & 0 & \cdots & 0 \\
2\gamma & 2\gamma & 0 & 0 & \cdots & 0 \\
c_3 & a_{31}^{[E]} & a_{32}^{[E]} & 0 & \cdots & 0 \\
\vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\
1 & a_{s,1}^{[E]} & a_{s,2}^{[E]} & a_{s,3}^{[E]} & \cdots & 0 \\
\hline
 & b_1^{[E]} & b_2^{[E]} & b_3^{[E]} & \cdots & \gamma \\
\hline
 & \widehat{b}_1^{[E]} & \widehat{b}_2^{[E]} & \widehat{b}_3^{[E]} & \cdots & \widehat{b}_s^{[E]}
\end{array}
\qquad
\begin{array}{c|cccccc}
0 & 0 & 0 & 0 & \cdots & 0 \\
2\gamma & \gamma & \gamma & 0 & \cdots & 0 \\
c_3 & a_{31}^{[I]} & a_{32}^{[I]} & \gamma & \cdots & 0 \\
\vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\
1 & a_{s,1}^{[I]} & a_{s,2}^{[I]} & a_{s,3}^{[I]} & \cdots & \gamma \\
\hline
 & b_1^{[I]} & b_2^{[I]} & b_3^{[I]} & \cdots & \gamma \\
\hline
 & \widehat{b}_1^{[I]} & \widehat{b}_2^{[I]} & \widehat{b}_3^{[I]} & \cdots & \widehat{b}_s^{[I]}
\end{array}
$$

In the above expression, the superscripts [*E*] and [*I*] have been used in reference to the *explicit* and *implicit* components of the additive ARK_2 integrator, respectively. As mentioned before, the stage order of the implicit integrator is two. On the other hand, the schemes proposed by Ascher et al. [3] and Calvo et al. [11] are also based on DIRK methods, but padded with a first row and column of zeroes. They do not possess stage-order of two in the implicit part, but have other interesting properties [32].

5.4.1. Implementation

One step of an *s*-stage two-part additive Runge–Kutta scheme, ARK_2, defined by its Butcher coefficients $(\boldsymbol{A}^{[I]}, \boldsymbol{A}^{[R]}, \boldsymbol{b}^{[I]}, \boldsymbol{b}^{[E]}, \hat{\boldsymbol{b}}^{[I]}, \hat{\boldsymbol{b}}^{[E]}, \boldsymbol{c})$, is given by

$$\boldsymbol{U}^{n+1} = \boldsymbol{U}^n + \Delta t \sum_{i=1}^{s} \left(b_i^{[I]} \boldsymbol{k}_i^{[I]} + b_i^{[E]} \boldsymbol{k}_i^{[E]} \right) \tag{73}$$

where $\boldsymbol{k}_i^{[I]}$ and $\boldsymbol{k}_i^{[E]}$ are the discrete counterparts of the stiff and nonstiff operators in (72), $\boldsymbol{f}_{\mathrm{s}}$ and $\boldsymbol{f}_{\mathrm{ns}}$,

$$\boldsymbol{k}_i^{[I]} = \boldsymbol{f}_{\mathrm{s}}^h(t_i, \boldsymbol{U}_i), \quad \boldsymbol{k}_i^{[E]} = \boldsymbol{f}_{\mathrm{ns}}^h(t_i, \boldsymbol{U}_i) \tag{74}$$

and the stage values are defined as

$$\boldsymbol{U}_i = \boldsymbol{U}^n + \Delta t \sum_{j=1}^{s} \left(a_{ij}^{[I]} \boldsymbol{k}_i^{[I]} + a_{ij}^{[E]} \boldsymbol{k}_i^{[E]} \right) \tag{75}$$

Restricting our analysis on ARK_2 pairs that use DIRK schemes for the implicit part, the above expression can be rearranged to obtain

$$\boldsymbol{U}_i = \boldsymbol{U}^n + \Delta t \sum_{j=1}^{i-1} \left(a_{ij}^{[I]} \boldsymbol{k}_j^{[I]} + a_{ij}^{[E]} \boldsymbol{k}_j^{[E]} \right) + \Delta t a_{ii}^{[I]} \boldsymbol{k}_i^{[I]} \tag{76}$$

We are interested in the linearly implicit case, for which the above expression is a linear system of equations of the form

$$\left(\boldsymbol{I} - \Delta t a_{ii}^{[I]} \boldsymbol{K} \right) \boldsymbol{U}_i = \boldsymbol{U}^n + \Delta t \sum_{j=1}^{i-1} \left(a_{ij}^{[I]} \boldsymbol{k}_j^{[I]} + a_{ij}^{[E]} \boldsymbol{k}_j^{[E]} \right) \tag{77}$$

where $\boldsymbol{k}_i^{[I]} = \boldsymbol{K}\boldsymbol{U}_i$. After solving $\boldsymbol{U}_i$ from (77), we can compute $\boldsymbol{k}_i^{[I]} = \boldsymbol{f}_{\mathrm{s}}(t_i, \boldsymbol{U}_i)$, and $\boldsymbol{k}_i^{[E]} = \boldsymbol{f}_{\mathrm{ns}}(t_i, \boldsymbol{U}_i)$. Similarly to *DIRK* schemes, the solution of the above system of equations may be accelerated by using stage value predictors. These can be constructed

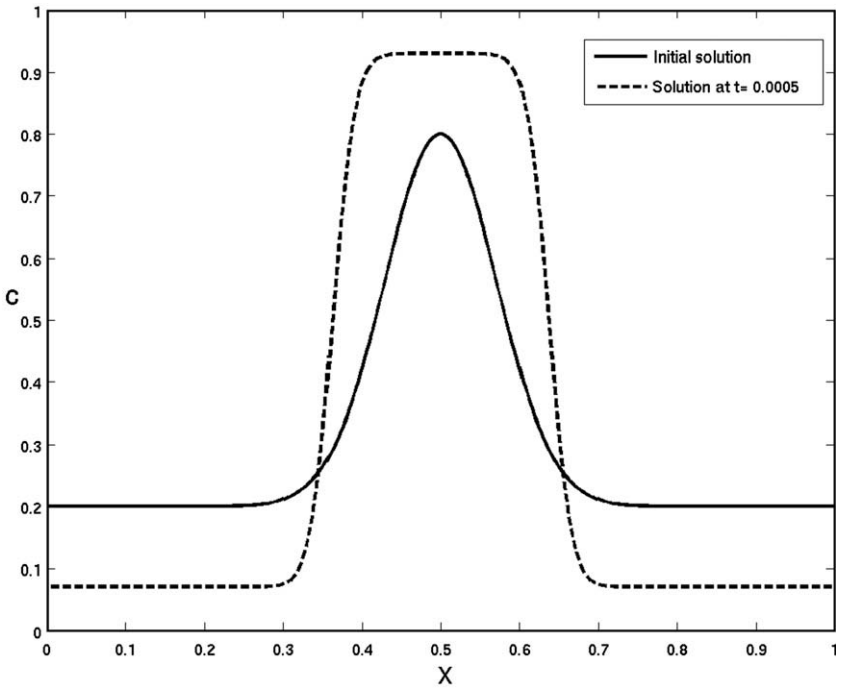

Fig. 4. Cahn–Hilliard equation: model problem for the 1D accuracy test. Initial and final solution profiles.

by means of dense output formulas provided by the designer of the ARK_2 scheme [32], or through more general techniques [26,29,35,41].

The error estimator is constructed again in terms of the solution provided by the embedded scheme,

$$\widehat{\boldsymbol{U}}^{n+1} = \boldsymbol{U}^n + \Delta t \sum_{i=1}^{s} \left(\hat{b}_i^{[I]} \boldsymbol{k}_i^{[I]} + \hat{b}_i^{[E]} \boldsymbol{k}_i^{[E]} \right) \tag{78}$$

and given by some suitable norm of the difference between the original and embedded solutions, $r^{n+1} = \|\boldsymbol{U}^{n+1} - \widehat{\boldsymbol{U}}^{n+1}\|$.

The *IMEX ARK*$_2$ approach exposed above has been used to integrate in time the Kuramoto–Sivashinsky equations analyzed in this study. In particular, we choose the *ARK*4(3)6*L*[2]*SA* method of Kennedy and Carpenter [32]. The following paragraphs present an overview of the specific implementation for the two-dimensional case. Bearing these guidelines in mind, the formulation of the 1D problem is straightforward.

Recall the 2D damped Kuramoto–Sivashinsky equation

$$\frac{\partial u}{\partial t} + \nabla \cdot (\nabla u + \nabla(\Delta u)) = -\alpha u + |\nabla u|^2 \tag{79}$$

A natural splitting into stiff and nonstiff terms, at the continuous level, can be conceptually expressed as

$$\frac{\partial u}{\partial t} = f_{\mathrm{ns}}(u) + f_{\mathrm{s}}(u) \tag{80}$$

where the *linear*, *stiff* terms, are

$$f_{\mathrm{s}}(u) = -\nabla \cdot (\nabla u + \nabla(\Delta u)) \tag{81}$$

while the *nonlinear*, *nonstiff* terms, are

$$f_{\mathrm{ns}}(u) = -\alpha u + |\nabla u|^2 \tag{82}$$

Following the proposed finite volume discretization, the above splitting can be retained at the semidiscrete level. Specifically, we can write

Table 1
Spatial convergence for a Cahn–Hilliard model problem (91)–(94)

Grid	L_2 error	Slope	L_∞ error	Slope
40	2.55×10^{-2}		8.68×10^{-2}	
120	9.22×10^{-4}	3.02	3.14×10^{-3}	3.02
360	1.01×10^{-4}	2.01	3.45×10^{-4}	2.01

MLS approximation of order $p = 3$.

Table 2
Spatial convergence for a Cahn–Hilliard model problem (91)–(94)

Grid	L_2 error	Slope	L_∞ error	Slope
40	2.40×10^{-2}		8.44×10^{-2}	
120	5.12×10^{-5}	5.60	1.89×10^{-4}	5.55
360	6.77×10^{-7}	3.94	2.51×10^{-6}	3.93

MLS approximation of order $p = 5$.

Table 3
Spatial convergence for a Cahn–Hilliard model problem (91)–(94)

Grid	L_2 error	Slope	L_∞ error	Slope
40	2.38×10^{-2}		8.40×10^{-2}	
120	4.79×10^{-6}	7.75	1.83×10^{-5}	7.67
360	7.67×10^{-9}	5.86	3.05×10^{-8}	5.82

MLS approximation of order $p = 7$.

Table 4
Spatial convergence for a Kuramoto–Sivashinsky model problem (95) and (96)

Grid	L_2 error	Slope	L_∞ error	Slope
100	3.06×10^{-1}		1.63×10^{0}	
200	8.18×10^{-2}	1.90	4.35×10^{-1}	1.91
400	2.08×10^{-2}	1.98	1.10×10^{-1}	1.98

MLS approximation of order $p = 3$.

Table 5
Spatial convergence for a Kuramoto–Sivashinsky model problem (95) and (96)

Grid	L_2 error	Slope	L_∞ error	Slope
100	4.20×10^{-2}		2.22×10^{-1}	
200	3.20×10^{-3}	3.71	1.76×10^{-2}	3.66
400	2.12×10^{-4}	3.92	1.17×10^{-3}	3.91

MLS approximation of order $p = 5$.

Table 6
Spatial convergence for a Kuramoto–Sivashinsky model problem (95) and (96)

Grid	L_2 error	Slope	L_∞ error	Slope
100	1.18×10^{-2}		5.68×10^{-2}	
200	2.71×10^{-4}	5.44	1.36×10^{-3}	5.38
400	4.24×10^{-6}	6.00	2.13×10^{-5}	6.00

MLS approximation of order $p = 7$.

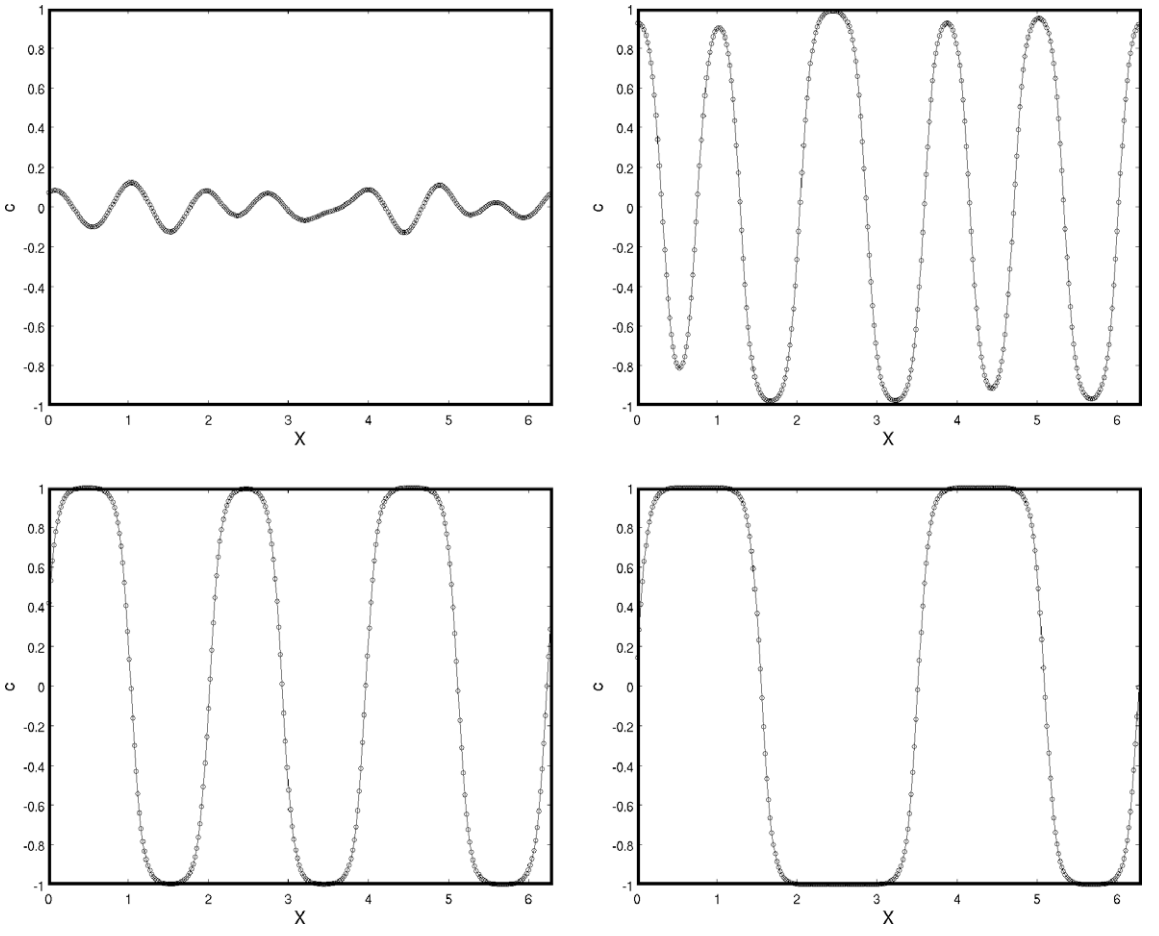

Fig. 5. Snapshots of a coarsening problem governed by the Cahn–Hilliard equation (98). Solution profiles at $t = 0.168$ (top left), $t = 7.15$ (top right), $t = 855$ (bottom left) and $t = 10,000$ (bottom right) time units.

$$\boldsymbol{M}\frac{\mathrm{d}\boldsymbol{u}}{\mathrm{d}t} = \boldsymbol{f}_{\mathrm{s}}^{h}(\boldsymbol{u}^{h}) + \boldsymbol{f}_{\mathrm{ns}}^{h}(\boldsymbol{u}^{h}) \tag{83}$$

where the h superscripts denote discrete values. Following the notation introduced in Eqs. (56)–(58), the above discrete components, $\boldsymbol{f}_{\mathrm{s}}^{h}$ and $\boldsymbol{f}_{\mathrm{ns}}^{h}$, correspond to

$$\boldsymbol{f}_{\mathrm{s}}^{h}(\boldsymbol{u}^{h}) = -\boldsymbol{R}^{h}, \quad \boldsymbol{f}_{\mathrm{ns}}^{h}(\boldsymbol{u}^{h}) = \boldsymbol{S}^{h} \tag{84}$$

In order to make the implementation of the above ARK_2 scheme more clear, we can identify

$$\boldsymbol{k}^{[I]} = \boldsymbol{f}_{\mathrm{s}}^{h}(\boldsymbol{u}^{h}) = -\boldsymbol{R}^{h}, \quad \boldsymbol{k}^{[E]} = \boldsymbol{f}_{\mathrm{ns}}^{h}(\boldsymbol{u}^{h}) = \boldsymbol{S}^{h} \tag{85}$$

As for the system matrix $\boldsymbol{K}$, recall that the linear stiff term can be written as

$$\boldsymbol{R}^{h} = \boldsymbol{K}\boldsymbol{U} \tag{86}$$

and the matrix $\boldsymbol{K} = \{K_{ij}\}$ is given by

$$K_{ij} = \sum_{\mathrm{ig}}^{\mathrm{ng}_i} \left(\begin{array}{c} \frac{\partial^3 N_j}{\partial x^3} + \frac{\partial^3 N_j}{\partial y^2 \partial x} + \frac{\partial N_j}{\partial x} \\ \frac{\partial^3 N_j}{\partial x^2 \partial y} + \frac{\partial^3 N_j}{\partial y^3} + \frac{\partial N_j}{\partial y} \end{array} \right)_{\mathrm{ig}} \cdot \boldsymbol{n}_{\mathrm{ig}} \omega_{\mathrm{ig}} \tag{87}$$

where the summation extends over all the quadrature points ng_i on the edges of the control volume i. The above general expression does not include the boundary conditions, which are specific to each problem set up.

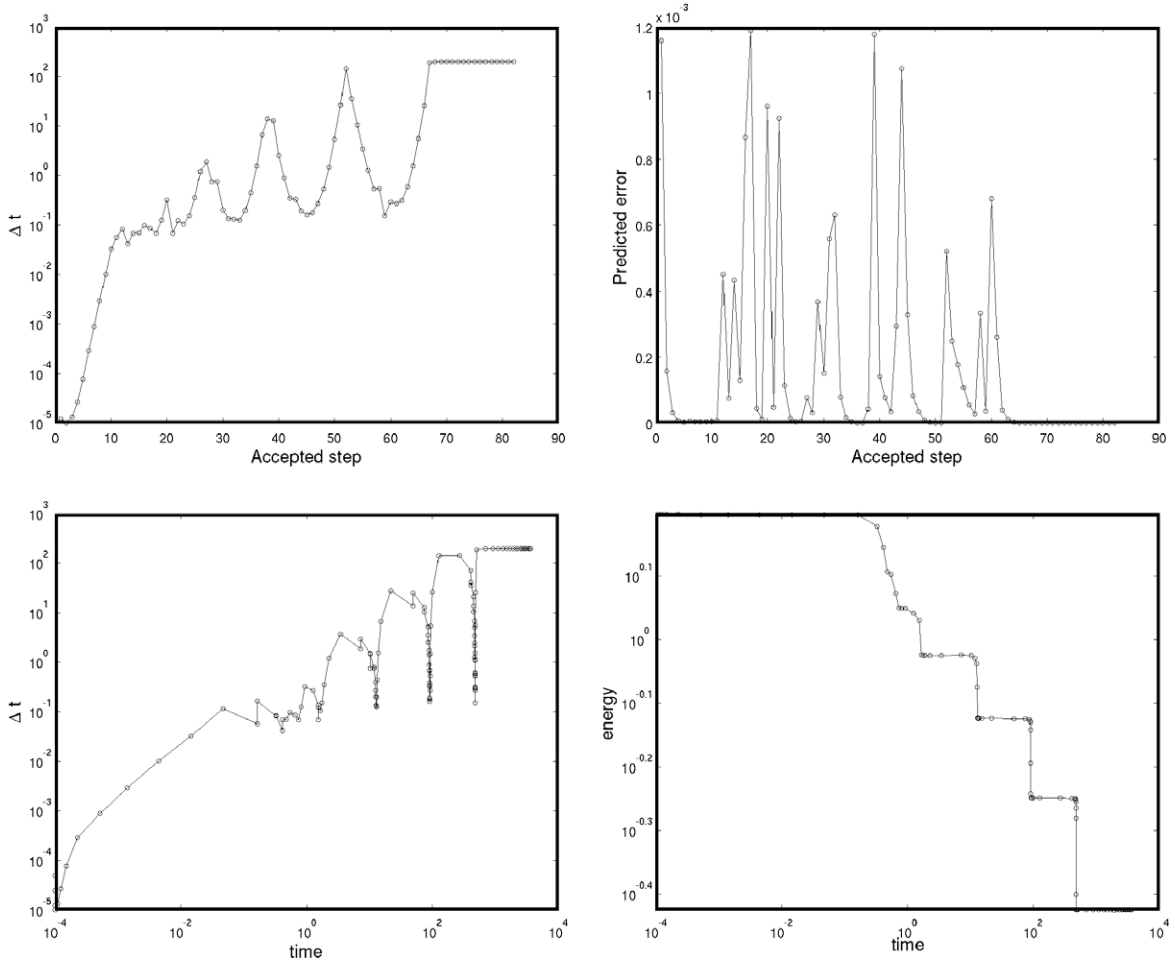

Fig. 6. Dynamics of a phase separation problem governed by the Cahn–Hilliard equation (98). The error tolerance for the time integrator is $\phi = 10^{-3}$. Time step size and predicted error against accepted steps (top left and right, respectively), time step size against time (bottom left), and energy against time (bottom right).

5.5. Step size control

5.5.1. Controller for the implicit ESDIRK scheme

Probably the most basic control rule is to compute the next time step size, Δt^{n+1}, as

$$\Delta t^{n+1} = \kappa \Delta t^n \left(\frac{\epsilon}{r^{n+1}} \right)^{1/k} \tag{88}$$

where ϵ is a fraction of the local error tolerance ϕ, r is a suitable estimation of the time integration error, and k a function of the order of the integration scheme. The idea is to keep the error close to a certain specified tolerance ϕ. Whenever an integration step leads to an estimated error $r > v\phi$, where typically $v = 1.2$ the step is rejected, and a new step is made with a smaller step size. An efficient step size control strategy aims, of course, at predicting the new step sizes in such a way that their values are maximized, while honoring the error constraints, and avoiding an excessive number of rejected steps.

For the ESDIRK4 scheme we have used the above formula with $\kappa = 1,\ \epsilon = 0.5\phi$ and $k = 4$. More sophisticated controllers for implicit Runge–Kutta methods have been proposed [27], although for the analyzed examples we did not find a significant increase in the efficiency of the scheme.

5.5.2. Controller for the implicit–explicit ARK_2 scheme

For IMEX time-stepping we use the PID controller [32,45]

$$\Delta t^{n+1} = \kappa \Delta t^n \left(\frac{\epsilon}{r^{n+1}} \right)^{\alpha} \left(\frac{r^n}{\epsilon} \right)^{\beta} \left(\frac{\epsilon}{r^{n-1}} \right)^{\gamma} \tag{89}$$

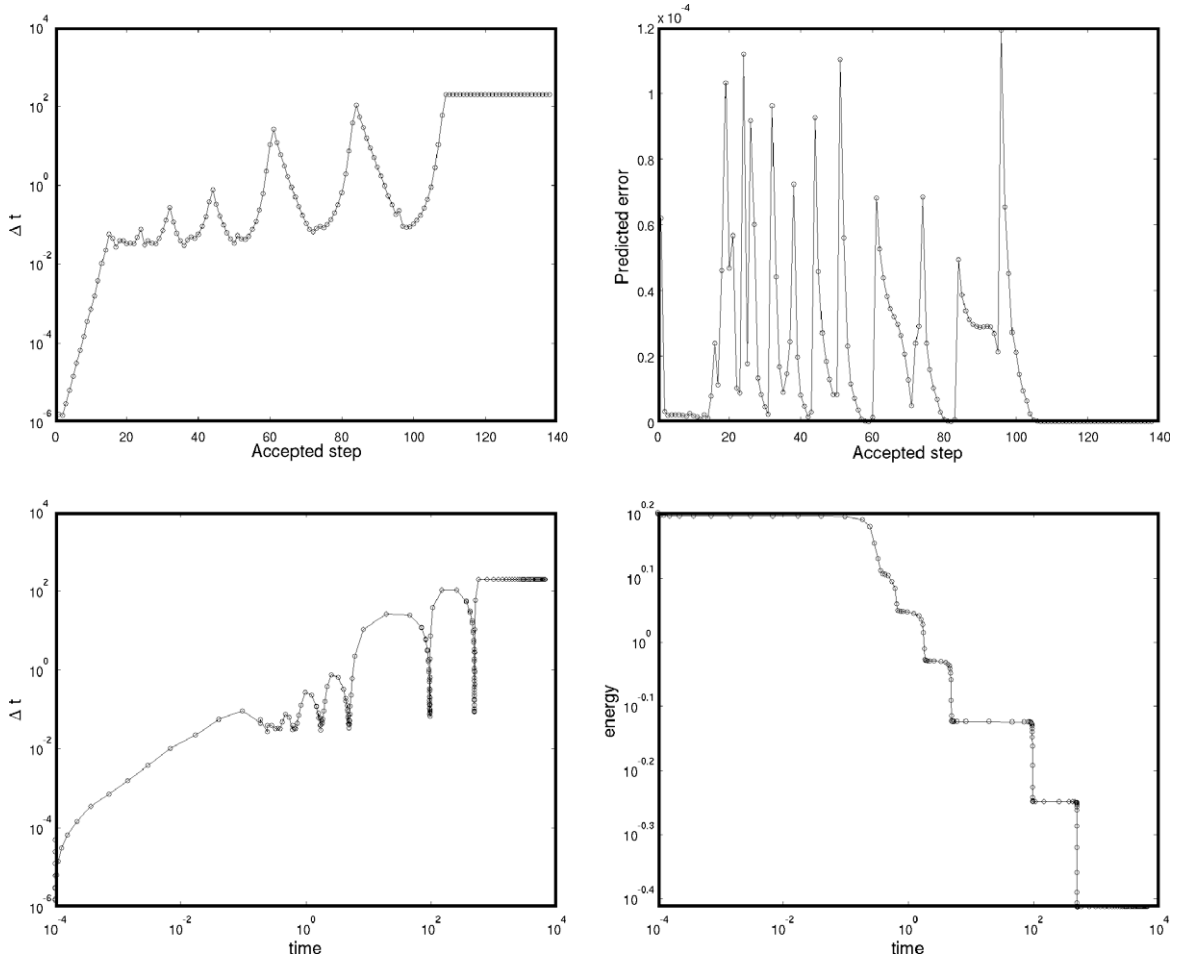

Fig. 7. Dynamics of a phase separation problem governed by the Cahn–Hilliard equation (98). The error tolerance for the time integrator is $\phi = 10^{-4}$. Time step size and predicted error against accepted steps (top left and right, respectively), time step size against time (bottom left), and energy against time (bottom right).

where $r = \|\delta\|_\infty$, $\delta = U - \widehat{U}$, $\kappa = 1$, $v = 1.2$ and $\epsilon = 0.8\phi$. For the remaining parameters we follow [32] and, defining $\Omega^n = \Delta t^n / \Delta t^{n-1}$ and

$$k\alpha = \left[k_I + k_P + \left(\frac{2\omega^n}{1+\omega^n}\right)k_D\right], \quad k\beta = (k_P + 2\omega^n k_D), \quad k\gamma = \left(\frac{2\omega^{2n}}{1+\omega^n}\right)k_D \tag{90}$$

we set $k_I = 0.25$, $k_P = 0.14$ and $k_D = 0.10$, which provide the values of α, β and γ. In this case k is the order of the embedded scheme, $k = 3$.

6. Preliminary 1D analysis: accuracy and error control

6.1. Convergence tests

6.1.1. Cahn–Hilliard

The model problem for the first accuracy test is the Cahn–Hilliard equation

$$c_t = (B(c)(-\gamma c_{xxx} + (\Psi'(c))_x))_x, \quad c \in [0,1] \tag{91}$$

with periodic boundary conditions in $[0,1]$, and initial condition

$$c(x,0) = 0.2 + 0.6\,e^{-100(x-0.5)^2} \tag{92}$$

The energy potential is given by

$$\Psi(c) = 1000(c\ln c + (1-c)\ln(1-c)) + 3000c(1-c) \tag{93}$$

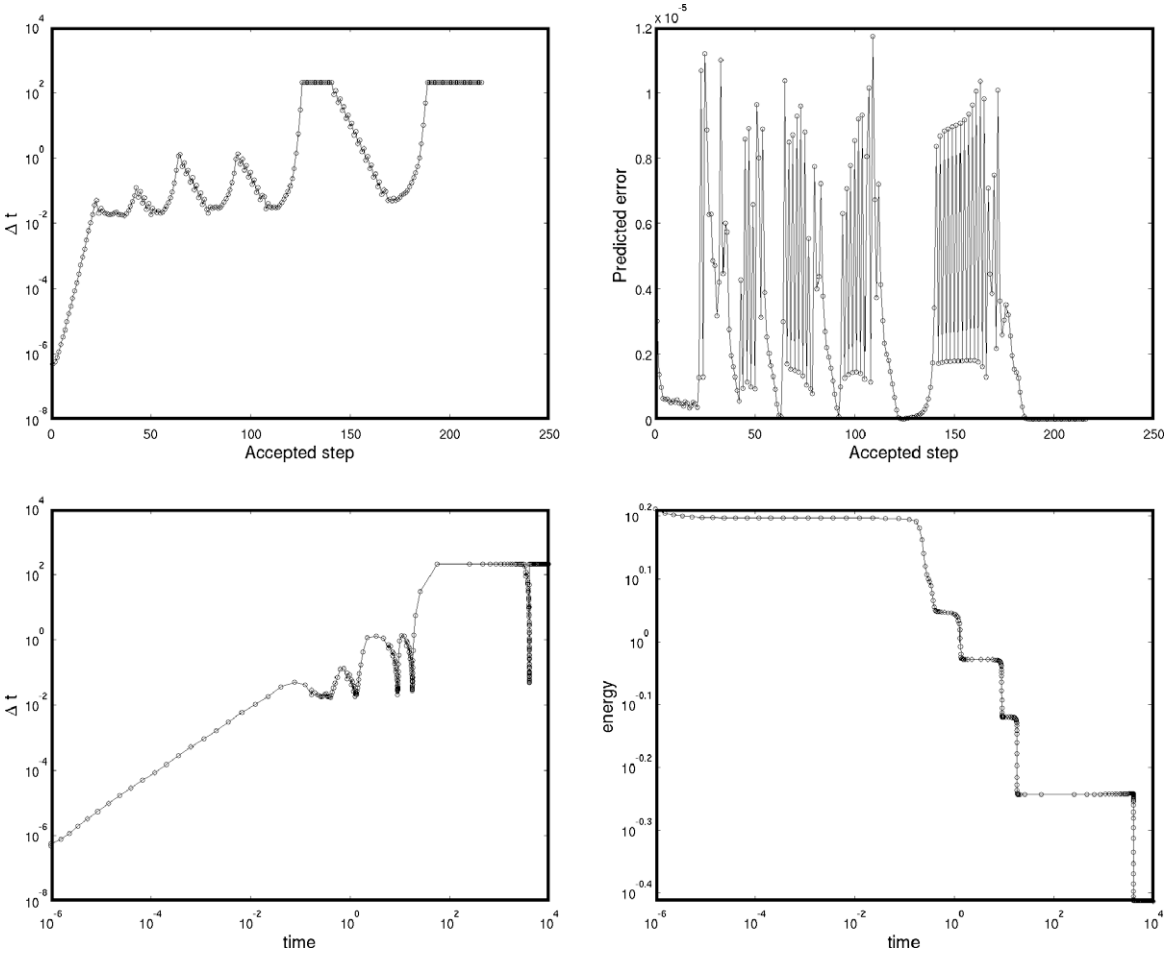

Fig. 8. Dynamics of a phase separation problem governed by the Cahn–Hilliard equation (98). The error tolerance for the time integrator is $\phi = 10^{-5}$. Time step size and predicted error against accepted steps (top left and right, respectively), time step size against time (bottom left), and energy against time (bottom right).

while the mobility is set as the degenerate case

$$B(c) = c(1-c) \tag{94}$$

and $\gamma = 1$. The models adopted for the free energy and the mobility make this problem a representative one in terms of the assessment of the spatial performance of the proposed finite volume method. Its accuracy is evaluated at $t = 5 \times 10^{-4}$; the initial and final concentration profiles are depicted in Fig. 4. The solution is advanced in time using the ESDIRK4 scheme with constant time step, $\Delta t = 10^{-5}$. A *reference solution* was computed using a very fine grid (1080 cells), and the sixth-order scheme (MLS of order 7). All the errors are referred to this solution. Tables 1–3 show the computed errors on a series of successively refined grids, with 40, 120 and 360 cells, respectively. The schemes constructed using MLS approximations of orders $p = 3$, 5 and 7 have orders of converge two, four and six, as expected. It is apparent that the first level of refinement is still far from the asymptotic region.

6.1.2. Kuramoto–Sivashinsky

The second accuracy test is based on the following model problem, extracted from Xu and Shu [52]

$$u_t + \left(\frac{1}{2}u^2\right)_x + u_{xx} + \sigma u_{xxx} + u_{xxxx} = 0 \tag{95}$$

with periodic boundary conditions in $[-30, +30]$, and $\sigma = 4$. The exact solution is

$$u(x,t) = c + 9 - 15(\tanh(z) + \tanh^2(z) - \tanh^3(z)) \tag{96}$$

where $z = k(x - ct - x_0)$, with $c = 6,\ k = \frac{1}{2}$, and $x_0 = -10$. The errors are reported at $t = 1$. The solution is advanced in time with the *ARK*4(3)6*L*[2]*SA* method of Kennedy and Carpenter [32], and constant time step, $\Delta t = 0.005$. The proposed schemes achieve again the correct orders of the truncation error (Tables 4–6).

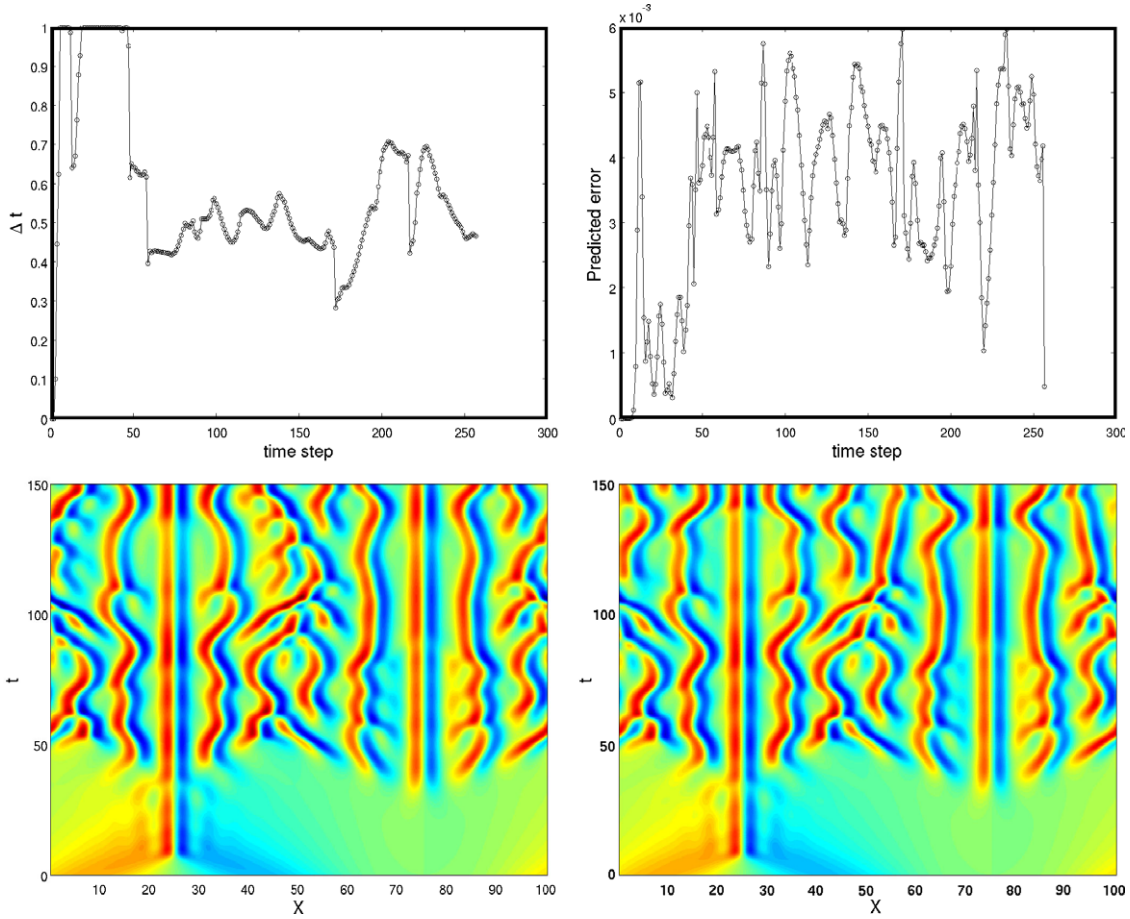

Fig. 9. Dynamical behaviour of the 1D Kuramoto–Sivashinsky, for the model problem (100) and (101). The error tolerance for the time integrator is $\phi = 5 \times 10^{-3}$. Finite volume solution using 600 cells and MLS approximation of order $p = 5$ (fourth order scheme). Time step size and predicted error against accepted steps (top left and right, respectively), computed solution in space–time (bottom left) and reference spectral solution (bottom right).

6.2. Dynamical tests: the error control strategy

Once the accuracy properties of the spatial discretization have been assessed, we turn our attention to more interesting one-dimensional dynamical examples. They are intended to evaluate the performance, and necessity, of the proposed error control and adaptive time-stepping approach.

The first example corresponds to a Cahn–Hilliard problem, and the dynamics of the solution may be regarded as a *canonical* model for phase separation processes. The solution exhibits extreme time scale variations, with cycles of short periods of time when the solution changes abruptly, followed by long periods of slow grain coarsening, until a final equilibrium is reached. The error-control based adaptive time-stepping procedure plays a crucial role in the efficient simulation of this type of problems.

For non-trivial simulations of the Kuramoto–Sivashinsky equation, the ability to control time integration errors, rather than the efficiency gains arising from the adaptive time-stepping strategy, is the crucial feature of our approach. The dynamics of the problem exhibit such sensitivity to small scale perturbations, that even relatively small errors incurred by the spatial or temporal discretizations have a significant impact on the large scale evolution of the system.

6.2.1. A one-dimensional phase separation problem

Consider the Cahn–Hilliard equation

$$c_t = (B(c)(-\gamma c_{xxx} + (\Psi'(c))_x))_x, \quad c \in [-1, +1] \tag{97}$$

where $\Psi(c) = \frac{1}{4}(c^2 - 1)^2$ and $B(c) = 1$, i.e.

$$c_t = (-\gamma c_{xxx} + (c^3 - c)_x)_x, \quad c \in [-1, +1] \tag{98}$$

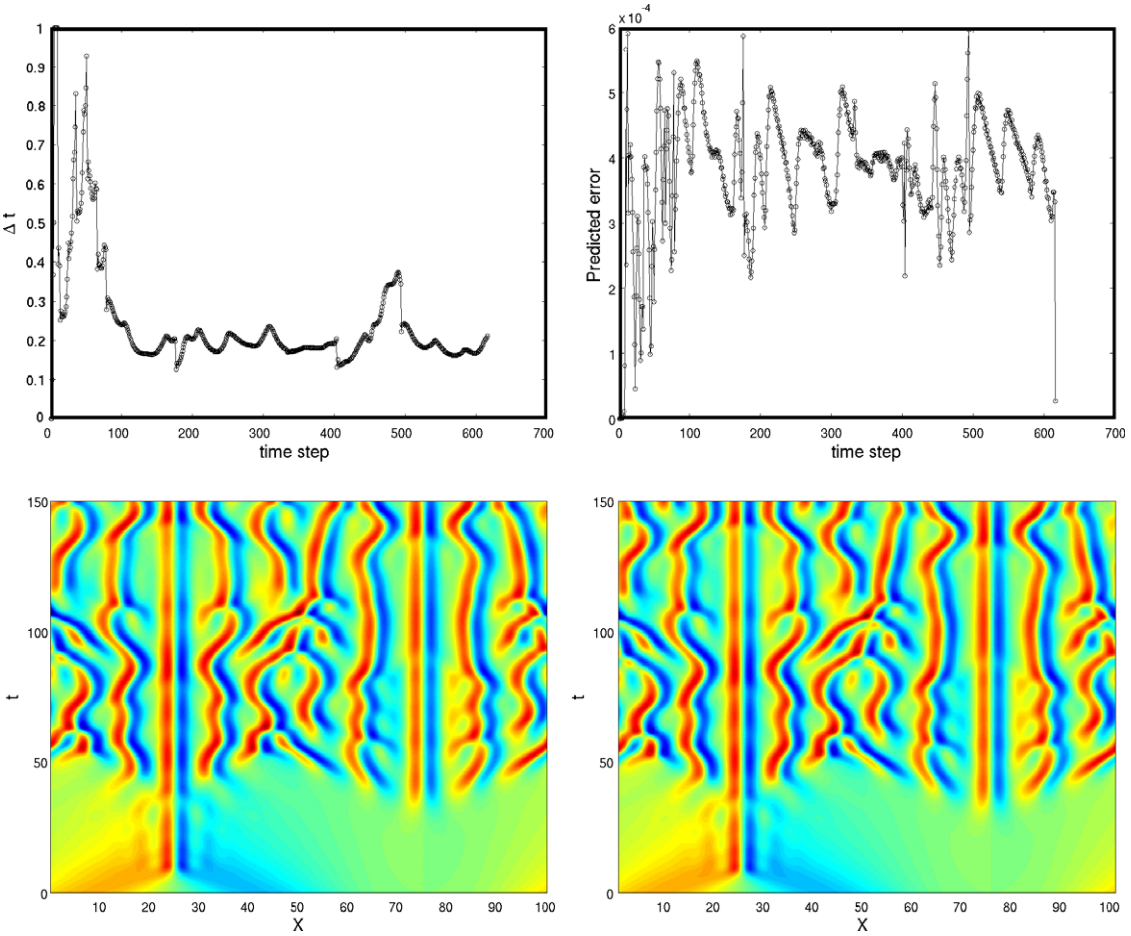

Fig. 10. Dynamical behaviour of the 1D Kuramoto–Sivashinsky, for the model problem (100) and (101). The error tolerance for the time integrator is $\phi = 5 \times 10^{-4}$. Finite volume solution using 600 cells and MLS approximation of order $p = 5$ (fourth order scheme). Time step size and predicted error against accepted steps (top left and right, respectively), computed solution in space–time (bottom left) and reference spectral solution (bottom right).

Furthermore, assume periodic boundary conditions in $[0, 2\pi]$, and $\gamma = 0.01$. The initial state is a random perturbation of maximum amplitude 0.05 around $c = 0$. Fig. 5 shows several snapshots of a realization of the above simulation conditions, computed on a grid comprising 300 cells. The dynamics of the solution reveal the aforementioned cycles of slow/fast coarsening. The crucial observation is that, as the simulation advances, the separation of time scales between the slow and fast phenomena increases dramatically, both in terms of consecutive slow-fast cycles, and with respect to the initial, fast transient.

The important practical consequence is the efficiency collapse of any time integration strategy that lacks quality error control and time step adaptivity. Leaving aside stability considerations, capturing the initial transients with reasonable accuracy requires a time step that is about eight orders of magnitude smaller than the characteristic time scale of the late grain coarsening evolution. Most existing simulation methods for the Cahn–Hilliard equation would require several million (implicit) time steps to complete this simple simulation with acceptable time accuracy. Some authors have tried to develop specific time integrators with enhanced stability properties, which may allow larger time steps. This approach, on the other hand, compromises the accuracy and the efficiency of the scheme, due to the lack of error control and to the fact that very long time steps may lead to slow/no convergence of the Newton iterations.

Note that time step modifications based on more or less heuristic coarsening laws are also likely to fail in this problem, as there is not one single time scale, but at least two distinctive time scales living in the same process, and both appear alternatively as the simulation advances: a *macroscale*, that governs the slow coarsening stages, and a *microscale*, which drives the fast ones. At the late stages of the coarsening simulation these macro–micro scales differ by about three orders of magnitude. In multidimensions the interaction of time scales is even richer, due to the existence of simultaneous aggregation processes at different locations, and therefore a systematic and rigorous approach to time step selection is of paramount importance.

We conduct simulations with 160 cells, the fourth order scheme (MLS order 5), and integrate in time using the fully implicit ESDIRK4 scheme. The error controller (88) was used for step size selection, with error tolerances $\phi_1 = 10^{-3}$, $\phi_2 = 10^{-4}$ and $\phi_3 = 10^{-5}$, and $\{\epsilon_j = 0.5\phi_j,\ j = 1, 2, 3\}$. The time step size and predicted error are plotted against the *accepted* (real) time

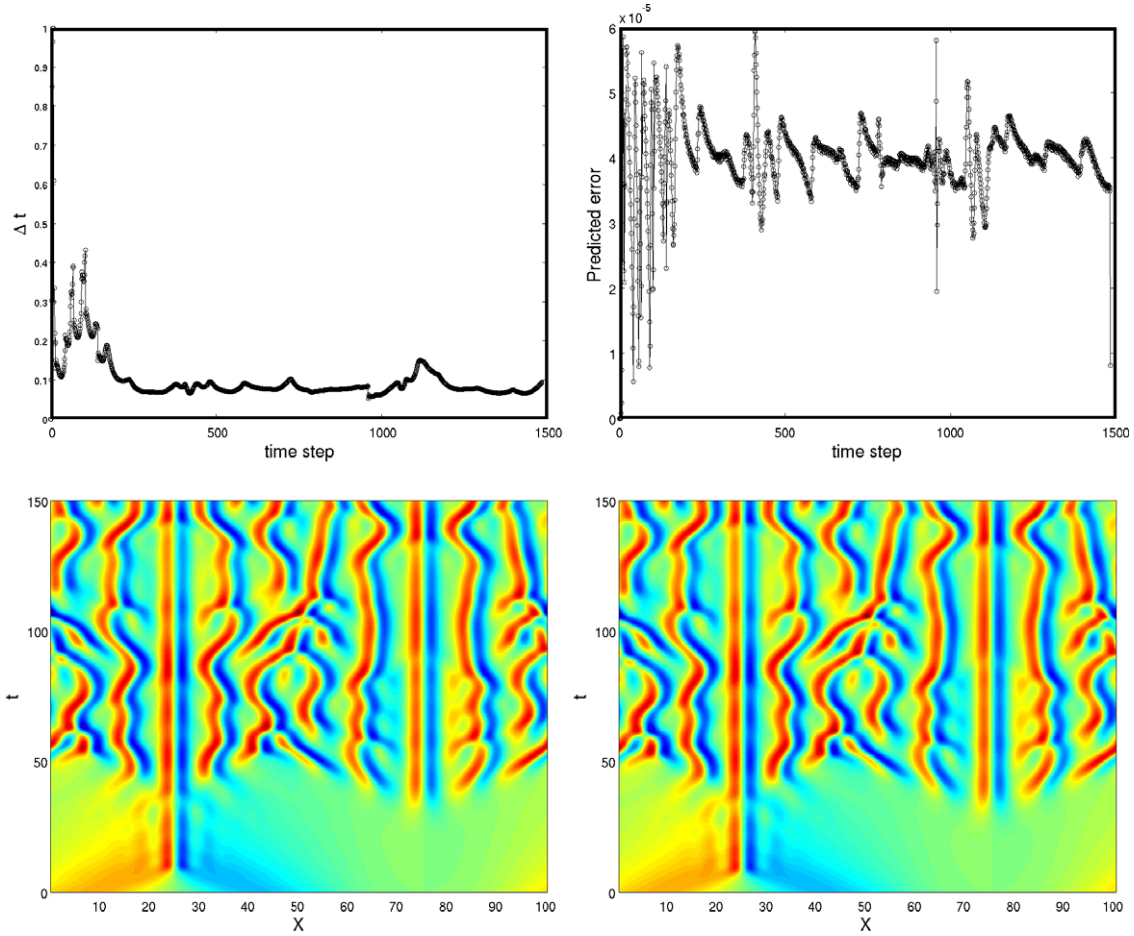

Fig. 11. Dynamical behaviour of the 1D Kuramoto–Sivashinsky, for the model problem (100) and (101). The error tolerance for the time integrator is $\phi = 5 \times 10^{-5}$. Finite volume solution using 600 cells and MLS approximation of order $p = 5$ (fourth order scheme). Time step size and predicted error against accepted steps (top left and right, respectively), computed solution in space–time (bottom left) and reference spectral solution (bottom right).

steps, i.e., the plotted quantities correspond to steps at which the predicted error was lower than 1.2ϕ. We also plot the time step size and energy as a function of time. Note that the time step plot also includes the *rejected* steps. The energy is computed from the functional (2), which is evaluated numerically as

$$\mathcal{E}^h = \sum_{i=1}^{N} \left(\frac{\gamma}{2} |\nabla c_i^h|^2 + \frac{1}{4}(c_i^2 - 1)^2 \right) A_i \tag{99}$$

Figs. 6–8 show the results associated to the three error tolerances. Note that the initial state is different in each simulation, and although the equilibrium state should be statistically identical, the dynamics of the solution may differ significantly. The simulations are run until the equilibrium configuration is reached (around $t = 10^4$). The maximum allowed time step was arbitrarily restricted to $\Delta t = 200$. Note that this example was simulated in [28] using a modified semi-implicit scheme with improved stability properties, and constant time steps up to $\Delta t = 0.01$.

The plots of time step size against time (bottom left) show that several rejected time steps occur when the solution changes abruptly. The relative number of rejected steps is bigger for the larger error tolerance (about 20%), and reduces

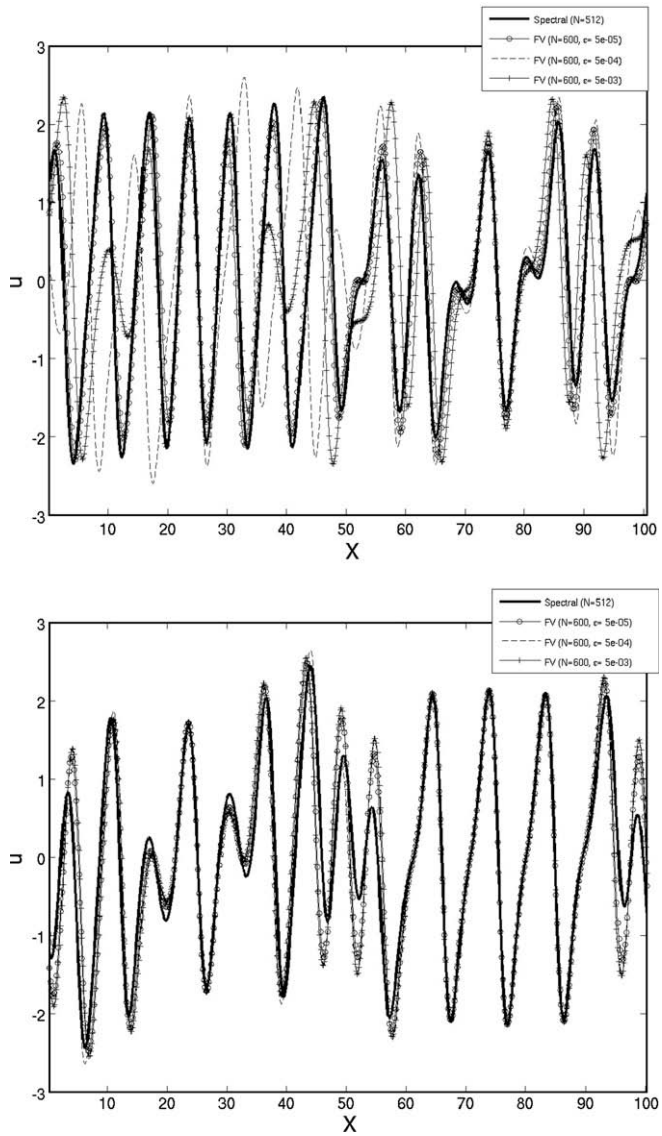

Fig. 12. Dynamical behaviour of the 1D Kuramoto–Sivashinsky, for the model problem (100) and (101). Influence of the time errors on the solution. Finite volume solution using 600 cells and MLS approximation of order $p = 5$ (fourth order scheme). Comparison of the computed profiles at $t = 150$ (top) and $t = 100$ (bottom) with the reference solution, for different error tolerances.

to about 10% for the lower error tolerances. It would be interesting to analyze more sophisticated controllers, capable of minimizing the number of rejected time steps.

We should remark that very small energy increases (about 0.1%) were observed at some point in the simulations. This was the case for the larger error tolerance ϕ_1, and also for the smaller one, ϕ_3. This fact indicates that the proposed time integration strategy does not possess strict gradient stability. Of course, the lack of strict gradient stability does not mean that the scheme is unstable, but rather suggests a conditional stability. As a matter of fact, our experience shows that instabilities may occur, but only at error tolerances that are unacceptably large for practical purposes.

6.2.2. One-dimensional Kuramoto–Sivashinsky equation

Consider the problem

$$u_t + \left(\frac{1}{2}u^2\right)_x + u_{xx} + u_{xxxx} = 0 \tag{100}$$

with periodic boundary conditions in $[0, 32\pi]$, and initial solution

$$u(x,0) = \cos\left(\frac{x}{16}\right)\left(1 + \sin\left(\frac{x}{16}\right)\right) \tag{101}$$

This example was taken from Kassam and Trefethen [30], where the authors demonstrate the convenience of high-order time integration schemes for this kind of problems. They find a modified exponential time differencing (ETD) scheme to be the most efficient one. Their spatial discretization is carried out by means of a spectral method. In our case, however, a finite volume method with general boundary conditions and the requirement of error control and adaptive time-stepping, that choice would not be competitive or even feasible, due to the necessity of computing the exponential of the matrix asso-

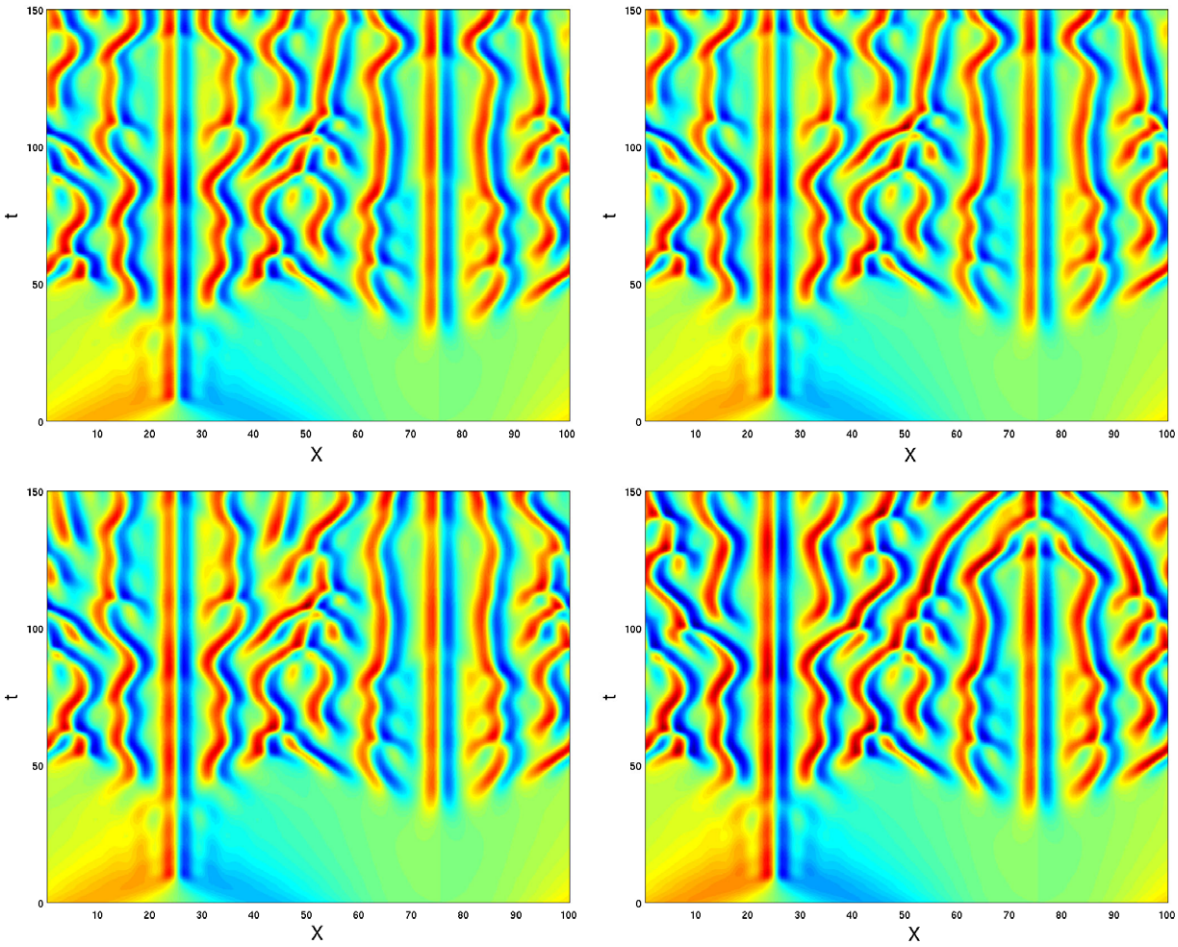

Fig. 13. Dynamical behaviour of the 1D Kuramoto–Sivashinsky, for the model problem (100) and (101). Influence of the spatial errors on the solution. Computed and reference solutions in space–time. The finite volume schemes use 150 cells and MLS approximations of orders $p = 15$ (top right), $p = 9$ (bottom left), and $p = 5$ (bottom right), respectively. The spectral solution uses $N = 512$ modes (top left).

ciated to the discretization of the linear high-order terms multiplied by each Δt. Our choice is the *ARK*4(3)6*L*[2]*SA* implicit–explicit method of Kennedy and Carpenter [32] in combination with the time step controller (89).

In order to assess the accuracy and efficiency of the proposed methodology, we run two sets of simulations of (100) and (101). In the first set of experiments we analyze the influence of time integration errors into the dynamics of the system. In order to focus on time errors, we use a grid of 600 cells, and the fourth order scheme (MLS with $p = 5$). We run three simulations, with error tolerances $\phi_1 = 5 \times 10^{-3}, \phi_2 = 5 \times 10^{-4}$ and $\phi_3 = 5 \times 10^{-5}$. The solution is advanced up to $t = 150$ units. For practical reasons, but rather arbitrarily, the time step size was limited to $\Delta t \leqslant 1$. Note that we did not impose stability restrictions on the time step; therefore, the time step selection is entirely left to the controller, and it is done solely in terms of error control.

For comparison purposes, a reference solution was computed using the spectral method with ETD time integration presented in [30]. We used $N = 512$ Fourier modes and $\Delta t = 1/20$. Figs. 9–11 show the results for the different error tolerances. We plot the evolution of the time step size and the predicted error as functions of the time step number, and the computed solution in space–time. The reference solution is also shown. In all cases the controller does a good job in keeping the predicted error close to the error tolerance (note that in the controller we aim at $\epsilon = 0.8\phi$). Also note that, as the error tolerance becomes smaller, the time step tends to be constant.

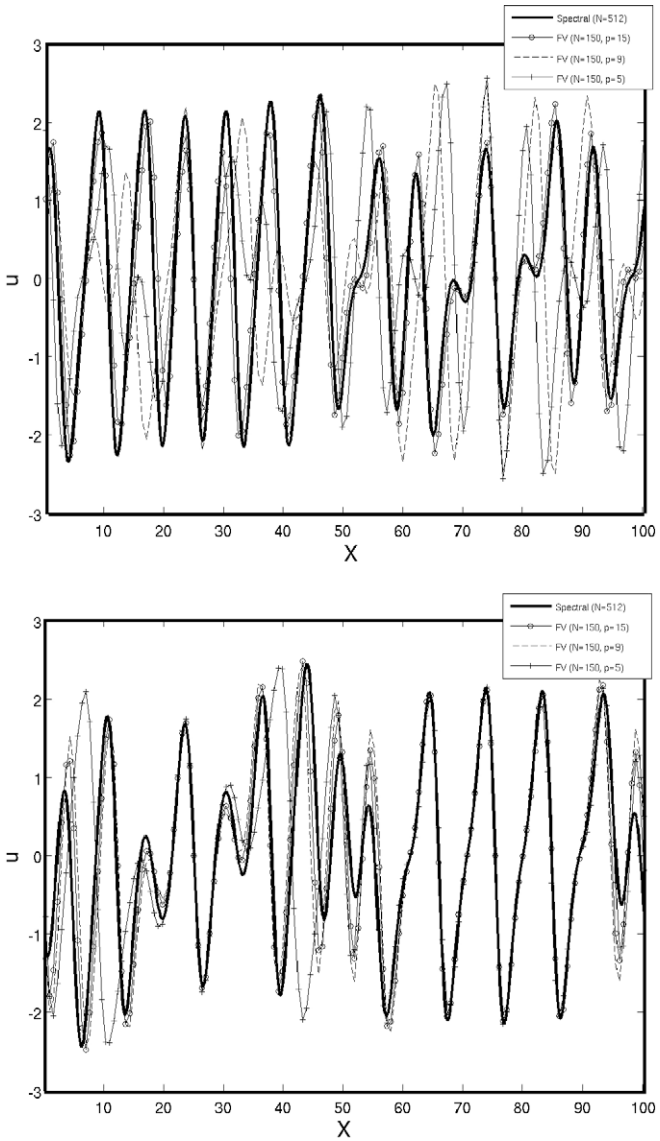

Fig. 14. Dynamical behaviour of the 1D Kuramoto–Sivashinsky, for the model problem (100) and (101). Influence of the spatial errors on the solution. Comparison of the computed profiles at $t = 150$ (top) and $t = 100$ (bottom) with the reference solution. The finite volume schemes use 150 cells and MLS approximations of orders $p = 15$, 9 and 9, respectively. The spectral solution uses $N = 512$ modes.

An important practical observation is that high error tolerances may result into sudden, important deviations from the reference solution. This fact is clearly present in the ϕ_1 solution (Fig. 9). Until around $t = 110$ units the solution is practically identical to the spectral one. Suddenly, some regions of the solution seem to follow a different path, and the final picture retains the basic features, but others are completely lost. This is much less pronounced for the intermediate error tolerance ϕ_2, and for ϕ_3 the solution is practically identical to the reference one. A comparison of the solution profiles at $t = 150$ (top) and $t = 100$ (bottom) is presented in Fig. 12.

We would also like to make some comments about the importance of the spatial discretization in problems which, as this one, exhibit a very complex dynamical behaviour. In particular, and in spite of the obvious differences in terms of the degree of complexity of the solutions, our conclusions will be somewhat extendable to turbulent flows modelled by the Navier–Stokes equations.

The influence of the spatial discretization on the solution dynamics is analyzed by setting $\phi = 5 \times 10^{-5}$ and running the simulation with 150 cells and MLS approximations of orders 15, 9 and 5 (the orders of convergence of the resulting schemes are, therefore, 14, 8 and 4, respectively). Of course, a method of order as high as 14 is not viable in multidimensions and general grids, due to the difficulties of defining appropriate stencils.

Fig. 13 presents a comparison of the computed solutions with the reference, spectral one (top left). The 14th order scheme provides a quite accurate solution, whose main features are perfectly captured by the finite volume method. On the other hand, the eighth and fourth order schemes yield solutions that, at some point, deviate from the spectral computation. In particular, the fourth order solution is reasonably accurate up to around $t = 80$ units, and then most features of the solution follow paths that are quite different from the correct ones. A comparison of the solution profiles at $t = 150$ (top) and $t = 100$ (bottom) is presented in Fig. 14.

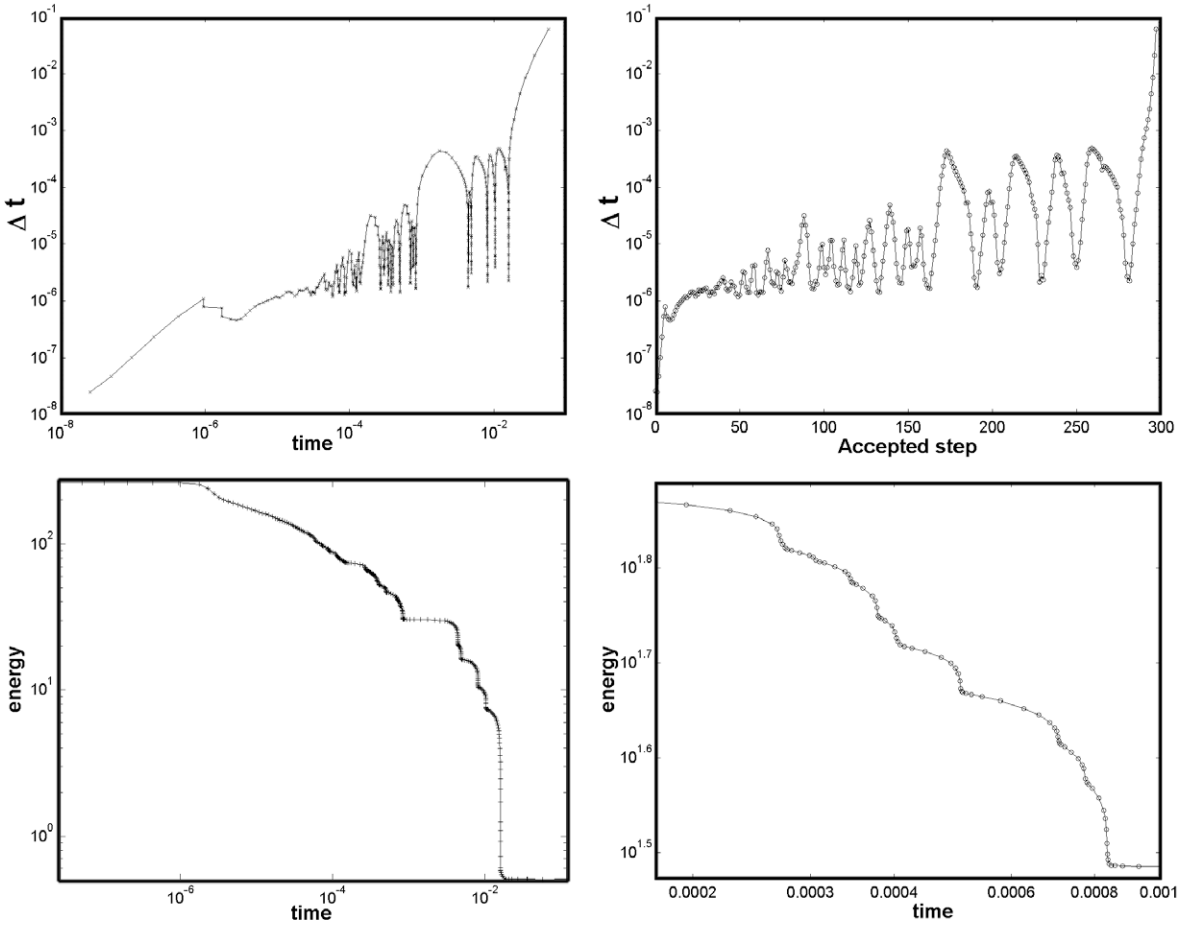

Fig. 15. Dynamics of a 2D phase separation problem governed by the Cahn–Hilliard model (102)–(104), with $A = 3000$. The error tolerance for the time integrator is $\phi = 10^{-4}$. Time step size against time and accepted steps (top left and right, respectively), and energy against time (bottom left). Detail of the rich energy cascade (bottom right).

7. Two-dimensional simulations

7.1. Phase separation and grain coarsening. A potential stiffness leakage

Consider the Cahn–Hilliard equation

$$\frac{\partial c}{\partial t} = \nabla \cdot (B(c)\nabla(-\gamma \Delta c + \Psi'(c))), \quad c \in [0,1] \tag{102}$$

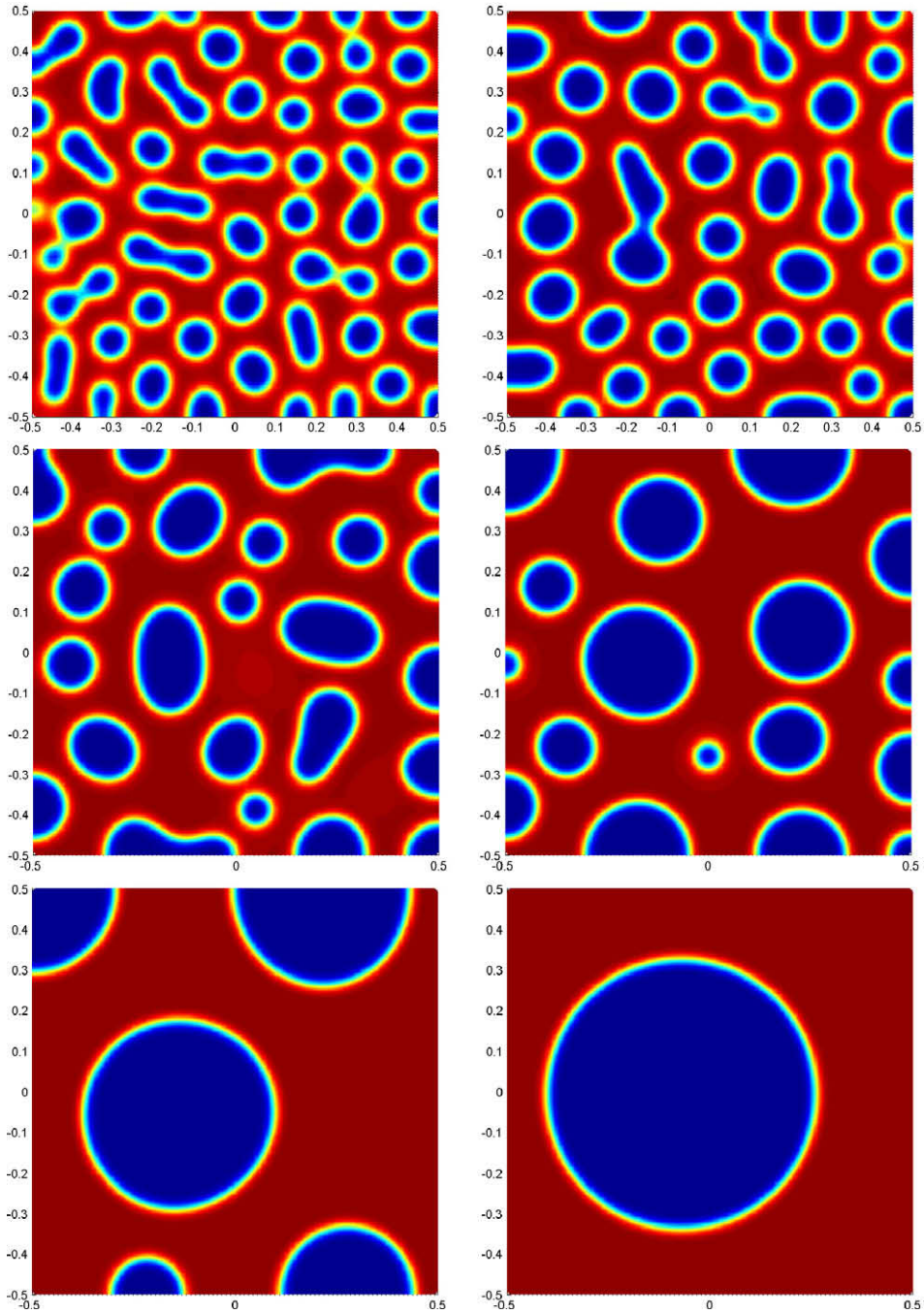

Fig. 16. A 2D phase separation problem governed by the Cahn–Hilliard model (102)–(104), with $A = 3000$. The error tolerance for the time integrator is $\phi = 10^{-4}$. Snapshots of the solution at different times: from top to bottom and from left to right, t = 6.52e−06, 1.91e−05, 6.39e−05, 2.63e−04, 4.70e−03 and 5.70e−02.

particularized for $\gamma = 1$, logarithmic free energy

$$\Psi(c) = A(c\ln c + (1-c)\ln(1-c)) + 3Ac(1-c) \tag{103}$$

parametrized in terms of a coefficient A, and degenerate mobilities

$$B(c) = c(1-c) \tag{104}$$

The dynamics of the solutions to this model problem are somewhat similar to those of the 1D "canonical" example shown above, although the complexity of these systems is significantly higher in multidimensions, due to the presence of multiple droplets that compete in the coarsening phase. As a consequence, the solutions are very rich in spatial and temporal scales, which challenges the efficiency of any numerical scheme.

In our first simulation, we take $A = 3000$. We use an 80×80 grid on $[-0.5, +0.5] \times [-0.5, +0.5]$, and the second order scheme (cubic MLS approximation). The initial state of Wells et al. [49] is reproduced here: a random perturbation around $c = 0.63$, with zero mean and maximum fluctuation of 0.05. At the boundary we impose $\nabla(-\gamma\Delta c + \Psi'(c)) \cdot \boldsymbol{n} = 0$. The time integration was carried out using the *ESDIRK4* scheme, with error tolerance $\phi = 10^{-4}$.

The results are presented in Figs. 15 and 16. The simulation runs up to $t = 0.057$, when the solution has already reached its final stage. The evolution of the time step size as a function of time and time step (Fig. 15, top left and top right, respectively) exhibits a pattern that is somewhat similar to the 1D canonical problem. The higher complexity of the 2D simulation is also apparent from these plots. Note the extremely wide range of time scales, which vary from $\Delta t \approx 10^{-8}$ at the beginning of the simulation, to $\Delta t \approx 10^{-2}$ at the final stages. The adaptive nature of the time-stepping strategy has a clear impact in the richness of the computed energy cascade (Fig. 15, bottom left and right). While one may identify a single time scale in the initial nucleation process, there is a clear separation of scales in the grain coarsening phase. The bifurcation takes place around $t = 10^{-5}$, and at least two distinctive time scales can be characterized. The *microscale*, which is of order 10^{-6}, seems to grow very slowly in time, whereas the *macroscale* exhibits a much faster growth, to the point that the difference between

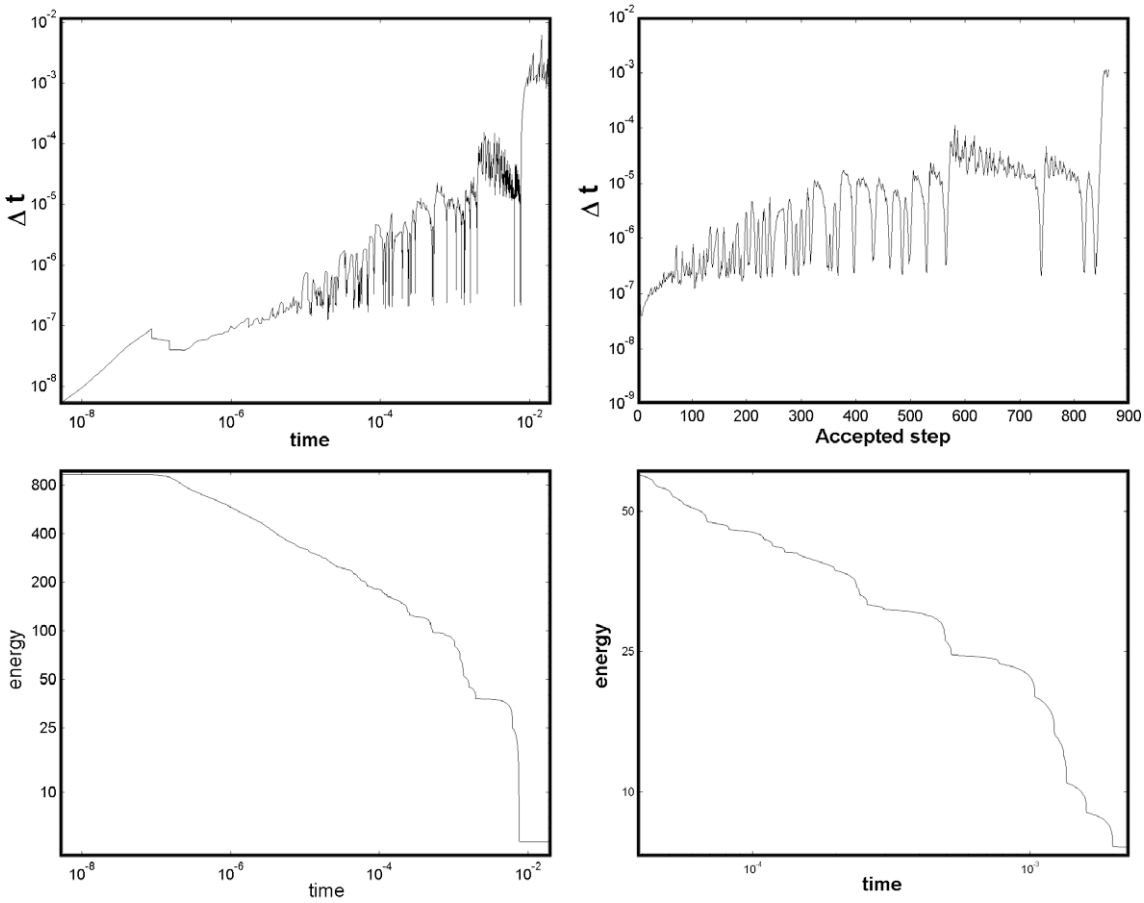

Fig. 17. Dynamics of a 2D phase separation problem governed by the Cahn–Hilliard model (102)–(104), with $A = 10,000$. The error tolerance for the time integrator is $\phi = 10^{-4}$. Time step size against time and accepted steps (top left and right, respectively), and energy against time (bottom left). Detail of the rich energy cascade (bottom right).

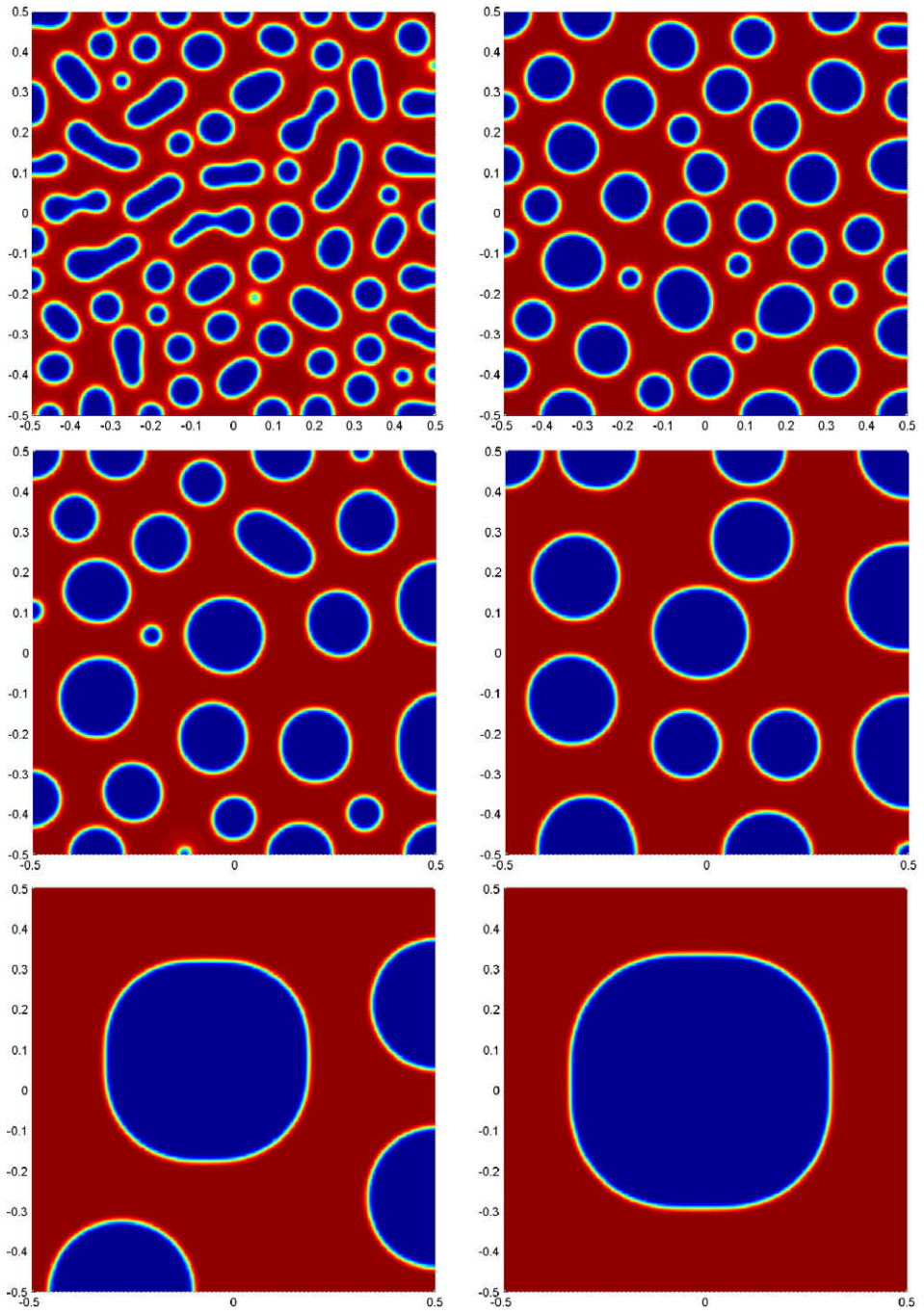

Fig. 18. A 2D phase separation problem governed by the Cahn–Hilliard model (102)–(104), with $A = 10,000$. The error tolerance for the time integrator is $\phi = 10^{-4}$. Snapshots of the solution at different times: from top to bottom and from left to right, t = 4.30e−06, 1.87e−05, 6.59e−05, 2.94e−04, 4.70e−03 and 2.04e−02.

the micro- and macroscales is about four orders of magnitude at the end of the simulation. Several snapshots of the solution are plotted in Fig. 16.

Note that about 30% of the steps were rejected. More than a half of this amount corresponds to *unsuccessful* or early discarded steps, by which we mean steps where slow/no convergence of the Newton or Krylov solvers was found. These unsuccessful steps are usually detected within the first stage of the Runge–Kutta step, and therefore their associated cost is small compared to that of the rejected steps. It is quite likely that the number of rejected steps can be significantly reduced with a more careful design of the controller.

Increasing the model parameter A in the free energy (103) results in a reduction of the thickness of the interfaces, and in the increase of the global stiffness of the problem. In our second example, we take $A = 10,000$ and a finer grid of 120×120 cells. The results are presented in Figs. 17 and 18. The simulation runs up to $t = 0.0204$, when the solution has already reached its final stage. The evolution of the time step size (Fig. 17, top left and right) reveals the increased complexity of the dynamics of the system, and its multiscale nature. Note that, compared to the $A = 3000$ case, we require about three times more time steps to complete the simulation, and that the time step evolution is quite rough. Several snapshots of the solution are plotted in Fig. 18.

An important remark is that, under increasingly stiffer conditions, the adaptive time-stepping strategy may suffer from a reduction in its efficiency. Thus, the time step size would be constantly "pulled" by the small scales, slowing down the advance of the simulation, and increasing the risk of producing rejected steps. This arises the possibility of a *stiffness leakage* of the adaptive procedure, in the limit of stiffness of the free energy model. This stiffness leakage is independent from other stiffness issues arising from the time integrator itself.

7.2. States of the damped Kuramoto–Sivashinsky equation

Recall the 2D damped Kuramoto–Sivashinsky equation

$$\frac{\partial u}{\partial t} + \nabla \cdot (\nabla u + \nabla(\Delta u)) = -\alpha u + |\nabla u|^2 \tag{105}$$

This equation has been studied as a model for systems characterized by the spontaneous appearance of coherent organized patterns (*cellular states*), when the initially unorganized system is driven away from thermodynamic equilibrium [42,18]. An

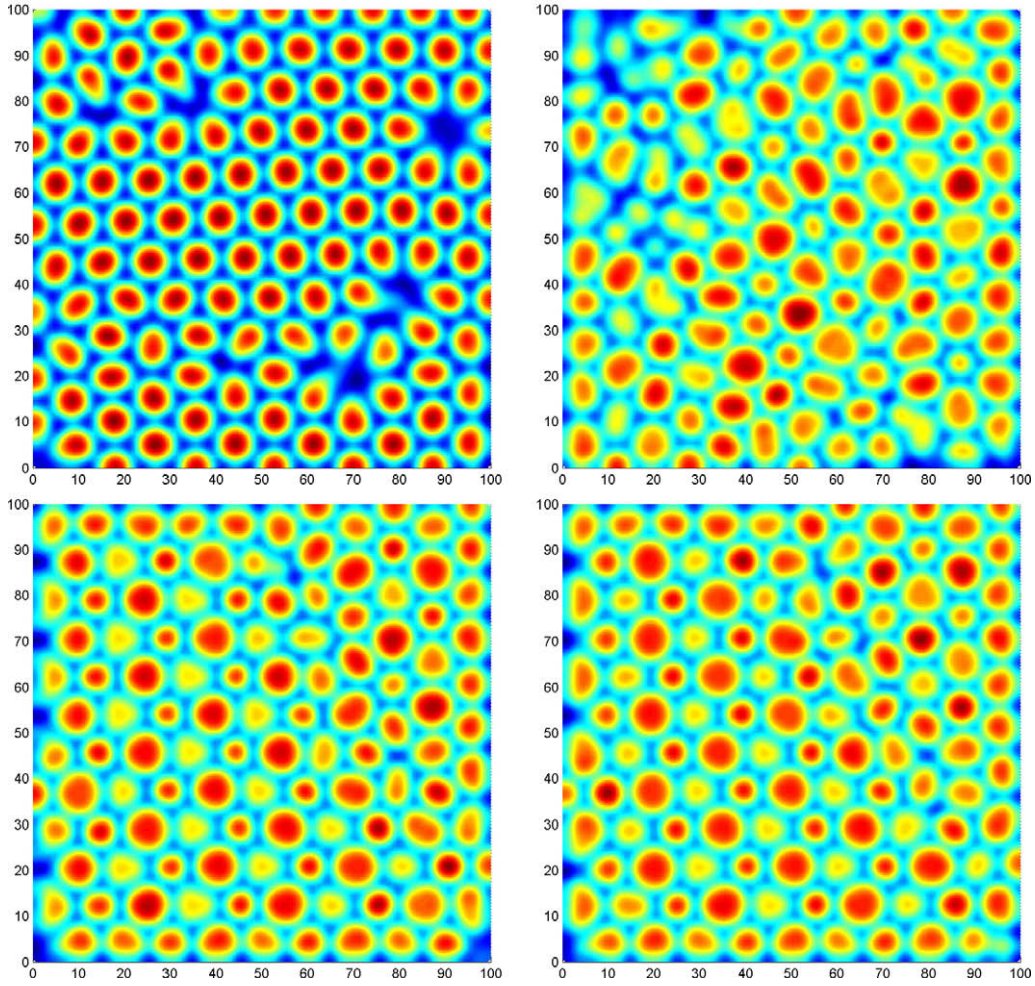

Fig. 19. States of the 2D damped Kuramoto–Sivashinsky equation as a function of the damping parameter α. Hexagonal state ($\alpha = 0.24$, top left), chaotic state ($\alpha = 0.17$, top right) and two configurations of the oscillating, breathing hexagonal state ($\alpha = 0.19$, bottom left and right).

important observation is that there exist secondary instabilities that can destroy the organized structure. A one-dimensional analysis of these secondary instabilities was presented in [40,19], whereas two-dimensional phenomenological and statistical studies can be found in [42,18].

The important finding in [42] is the characterization of an asymptotic state, at intermediate values of the damping parameter α, which exhibits a marked hexagonal structure, and oscillates or *breathes* in time. For larger values of α an hexagonal state is found, whereas smaller values lead to spatiotemporal chaotic solutions. In their numerical simulations they use a second order finite difference discretization with backward Euler time integration. They focus on the large aspect ratio limit, and therefore simulate large domains ($L = 512$). As mentioned before, they characterize three distinct states in the late time limit, depending on the value of the damping parameter α. At large values of α ($0.2176 < \alpha < 0.25$), a periodic hexagonal morphology dominates, with clear defects in the pattern. As α is decreased ($0.207 < \alpha < 0.2176$), the hexagonal state begins to oscillate and breathe. Each cell oscillates out of phase with its nearest neighbors. When α is decreased further, the breathing eventually drives the system to a spatiotemporal chaotic state.

The proposed finite volume scheme was used to reproduce these phenomena, but the set up is different from that of Paniconi and Elder [42]. Thus, the DKS problem (105) was solved in $[0, 100]$ with $\nabla(u + \Delta u) \cdot \boldsymbol{n} = 0$ on the boundary, and a grid of 120×120 cells. The implicit–explicit $ARK4(3)6L[2]SA$ scheme was used for time integration, with error tolerance $\phi = 10^{-4}$. The simulations are run up to $t = 15,000$.

The aforementioned states were also found in the present simulations, but at lower values of the damping parameter α. The results are presented in Fig. 19. For $\alpha = 0.24$ (top left) the hexagonal state is clearly observed. For low values of α, such as $\alpha = 0.17$ (top right), a characteristic spatiotemporal chaotic state is found. The breathing hexagonal state, on the other hand, appears already for $\alpha = 0.19$, while in [42] it is not found until $\alpha = 0.207$. This difference can be due to the fact that we use a smaller domain and different boundary conditions, which may result in a more constrained problem, thus increasing the stability of the solutions.

8. Conclusions

This paper presented a complete numerical method for the Cahn–Hilliard and Kuramoto–Sivashinsky type of equations. The spatial discretization is carried out using a high-order finite volume method, suitable for general, unstructured grids. The key ingredient of this scheme is a high-order global approximation framework, constructed using Moving Least-Squares (MLS). The reconstructed solution possesses high regularity, which allows a simple and efficient discretization of nonlinear equations with high order terms. The time integration is addressed by means of implicit an implicit–explicit Runge–Kutta schemes, with error control and adaptive time-stepping. The outcome is a practical, accurate and efficient tool which has been successfully applied to accuracy tests and representative simulations.

The use of adaptive time-stepping is of paramount importance in problems governed by the Cahn–Hilliard model, since an adaptive method may be several orders of magnitude more efficient than other schemes using constant or heuristic time steps. In addition, the time-adaptive procedure provides a *quantitative* characterization of the different time scales present in phase separation processes, which is another contribution of this study.

Acknowledgment

L. Cueto-Felgueroso gratefully acknowledges the funding received from "Ministerio de Educación y Ciencia" of Spain, through its program of postdoctoral scholarships.

References

[1] G. Akrivis, Y.-S. Smyrlis, Implicit–explicit BDF methods for the Kuramoto–Sivashinsky equation, Appl. Numer. Math. 51 (2004) 151–169.
[2] R. Alexander, Design and implementation of DIRK integrators for stiff systems, Appl. Numer. Math. 46 (2003) 1–17.
[3] U.M. Ascher, S.J. Ruuth, R.J. Spiteri, Implicit–explicit Runge–Kutta methods for time-dependent partial differential equations, Appl. Numer. Math. 25 (1997) 151–167.
[4] U.M. Ascher, S.J. Ruuth, B.T.R. Wetton, Implicit–explicit Runge–Kutta methods for time-dependent partial differential equations, SIAM J. Numer. Anal. 32 (1995) 797–823.
[5] A.L. Araújo, A. Murúa, J.M. Sanz-Serna, Symplectic methods based on decompositions, SIAM J. Numer. Anal. 34 (1997) 1926–1947.
[6] J.W. Barret, J.F. Blowey, Finite element approximation of a model for phase separation of a multi-component alloy with non-smooth free energy, Numer. Math. 77 (1997) 1–34.
[7] J.W. Barret, J.F. Blowey, H. Garcke, Finite element approximation of the Cahn–Hilliard equation with degenerate mobility, SIAM J. Numer. Anal. 37 (1999) 286–318.
[8] H. Bijl, M.H. Carpenter, V.N. Vatsa, C. Kennedy, Time integration schemes for the unsteady compressible Navier–Stokes equations: laminar flow, J. Comput. Phys. 179 (2002) 313–329.
[9] B. Bujanda, J.C. Jorge, Additive Runge–Kutta methods for the resolution of linear parabolic problems, J. Comput. Appl. Math. 140 (2002) 99–117.
[10] J.W. Cahn, J.E. Hilliard, Free energy of non-uniform systems. I. Interfacial free energy, J. Chem. Phys. 28 (1958) 258–267.
[11] M.P. Calvo, J. de Frutos, J. Novo, Linearly implicit Runge–Kutta methods for advection–reaction–diffusion equations, Appl. Numer. Math. 37 (2001) 535–549.
[12] S.M. Choo, Y.J. Lee, A discontinuous Galerkin method for the Cahn–Hilliard equation, J. Comput. Phys. 18 (2005) 113–126.
[13] G.J. Cooper, A. Sayfy, Additive methods for the numerical solution ordinary differential equations, Math. Comput. 35 (1980) 1159–1172.
[14] G.J. Cooper, A. Sayfy, Additive Runge–Kutta methods for stiff ordinary differential equations, Math. Comput. 40 (1983) 207–218.
[15] L. Cueto-Felgueroso, I. Colominas, J. Fe, F. Navarrina, M. Casteleiro, High order finite volume schemes on unstructured grids using Moving Least-Squares reconstruction. Application to shallow water dynamics, Int. J. Numer. Method Eng. 65 (2006) 295–331.

[16] L. Cueto-Felgueroso, I. Colominas, X. Nogueira, F. Navarrina, M. Casteleiro, Finite volume solvers and Moving Least-Squares approximations for the compressible Navier–Stokes equations on unstructured grids, Comput. Method Appl. Mech. Eng. 196 (2007) 4712–4736.
[17] L. Cueto-Felgueroso, I. Colominas, High-order finite volume methods and multiresolution reproducing kernels, Arch. Comput. Method Eng. 15 (2008) 185–228.
[18] I. Daumont, K. Kassner, C. Misbah, A. Valance, Cellular self-propulsion of two-dimensional dissipative structures and spatial-period tripling Hopf bifurcation, Phys. Rev. E 55 (1997) 6902–6906.
[19] K.R. Elder, J.D. Gunton, N. Goldenfeld, Transition to spatiotemporal chaos in the damped Kuramoto–Sivashinsky equation, Phys. Rev. E 56 (1997) 1631–1634.
[20] C.M. Elliott, D.M. French, A nonconforming finite element method for the two-dimensional Cahn–Hilliard equation, SIAM J. Numer. Anal. 26 (1989) 884–903.
[21] C.M. Elliott, D.M. French, F.A. Milner, A second order splitting method for the Cahn–Hilliard equation, Numer. Math. 54 (1989) 575–590.
[22] C.M. Elliott, H. Garcke, On the Cahn–Hilliard equation with degenerate mobility, SIAM J. Math. Anal. 27 (1996) 404–423.
[23] X. Feng, A. Prohl, Analysis of a fully discrete finite element method for the phase field model and approximation of its sharp interface limits, Math. Comput. 73 (2004) 541–567.
[24] J. Frank, W. Hundsdorfer, J.G. Verwer, On the stability of implicit–explicit linear multistep methods, Appl. Numer. Math. 25 (1997) 193–205.
[25] D. Furihata, A stable and conservative finite difference scheme for the Cahn–Hilliard equation, Numer. Math. 87 (2001) 675–699.
[26] S. González-Pinto, J.I. Montijano, S. Pérez-Rodríguez, Variable-order starting algorithms for implicit Runge–Kutta methods on stiff problems, Appl. Numer. Math. 44 (2003) 77–94.
[27] K. Gustafsson, Control theoretic techniques for stepsize selection in implicit Runge–Kutta methods, ACM Trans. Math. Softw. 20 (1994) 496–517.
[28] Y. He, Y. Liu, T. Tang, On large time-stepping methods for the Cahn–Hilliard equation, Appl. Numer. Math. 57 (2007) 616–628.
[29] I. Higueras, T. Roldán, Starting algorithms for some DIRK methods, Numer. Algor. 23 (2000) 357–369.
[30] A.-K. Kassam, L.N. Trefethen, Fourth order timestepping for stiff PDEs, SIAM J. Sci. Comput. 26 (2005) 1214–1233.
[31] D. Kay, R. Welford, A multigrid finite element solver for the Cahn–Hilliard equation, J. Comput. Phys. 212 (2006) 288–304.
[32] C.A. Kennedy, M.H. Carpenter, Additive Runge–Kutta schemes for convection–diffusion–reaction equations, Appl. Numer. Math. 44 (2003) 139–181.
[33] D. Kim, K. Kang, J. Lowengrub, Conservative multigrid methods for Cahn–Hilliard fluids, J. Comput. Phys. 193 (2004) 511–543.
[34] Y. Kuramoto, T. Tsuzuki, Persistent propagation of concentration waves in dissipative media far from thermal equilibrium, Prog. Theor. Phys. 55 (1976) 356–369.
[35] M.P. Laburta, Starting algorithms for IRK methods, J. Comput. Appl. Math. 83 (1997) 269–288.
[36] P. Lancaster, K. Salkauskas, Surfaces generated by moving least squares methods, Math. Comput. 155 (1981) 141–158.
[37] W.K. Liu, S. Li, T. Belytschko, Moving least-squares reproducing kernel methods: (I). Methodology and convergence, Comput. Method Appl. Mech. Eng. 143 (1997) 113–154.
[38] H. Liu, J. Zou, Some new additive Runge–Kutta methods and their applications, J. Comput. Appl. Math. 190 (2006) 74–98.
[39] E.V.L. de Mello, O. Teixeira da Silveira Filho, Numerical study of the Cahn–Hilliard equations in one, two and three dimensions, Physica A 347 (2005) 429–433.
[40] C. Misbah, A. Valance, Secondary instabilities in the stabilized Kuramoto–Sivashinsky equation, Phys. Rev. E 49 (1994) 166–182.
[41] H. Olsson, G. Söderlind, Stage value predictors and efficient Newton iterations in implicit Runge–Kutta methods, SIAM J. Sci. Comput. 20 (1998) 185–202.
[42] M. Paniconi, K.R. Elder, Stationary, dynamical and chaotic states of the two-dimensional damped Kuramoto–Sivashinsky equation, Phys. Rev. E 56 (1997) 2713–2721.
[43] G.I. Sivashinsky, Nonlinear analysis of hydrodynamic instability in laminar flames. I. Derivation of basic equations, Acta Astron. 4 (1977) 1177–1206.
[44] Y.S. Smyrlis, D.T. Papageorgiou, Computational study of chaotic and ordered solutions of the Kuramoto–Sivashinsky equation. ICASE Report 96-12, 1999.
[45] G. Söderlind, Automatic control and adaptive time-stepping, Numer. Algor. 31 (2002) 281–310.
[46] Z.Z. Sun, A second-order accurate linearized finite difference scheme for the two-dimensional Cahn–Hilliard equation, Math. Comput. 64 (1995) 1463–1471.
[47] R. Toral, A. Chakrabarti, J.D. Gunton, Droplet distribution for the two-dimensional Cahn–Hilliard model: comparison of theory with large-scale simulations, Phys. Rev. A 45 (1992) R2147.
[48] B.P. Vollmayr-Lee, A.D. Rutenberg, Fast and accurate coarsening simulation with an unconditionally stable time step, Phys. Rev. E 68 (2003) 066703.
[49] G.N. Wells, E. Kuhl, K. Garikipati, A discontinuous Galerkin method for the Cahn–Hilliard equation, J. Comput. Phys. 218 (2006) 860–877.
[50] R.W. Wittenberg, P. Holmes, Scale and space localization in the Kuramoto–Sivashinsky equation, Chaos 9 (1999) 452–465.
[51] Y. Xia, Y. Xu, C.-W. Shu, Local discontinuous Galerkin methods for the Cahn–Hilliard type equations, J. Comput. Phys. 227 (2007) 472–491.
[52] Y. Xu, C.-W. Shu, Local discontinuous Galerkin methods for the Kuramoto–Sivashinsky equations and the Ito-type coupled KdV equations, Comput. Method Appl. Mech. Eng. 195 (2006) 3430–3447.