Commun Nonlinear Sci Numer Simulat 17 (2012) 4930–4946

Contents lists available at SciVerse ScienceDirect

Commun Nonlinear Sci Numer Simulat

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

A new space–time discretization for the Swift–Hohenberg equation that strictly respects the Lyapunov functional

Hector Gomez *, Xesús Nogueira

Group of Numerical Methods in Engineering, University of A Coruña, Department of Mathematical Methods, Campus de Elviña, s/n, 15192 A Coruña, Spain

ARTICLE INFO

Article history:
Received 13 October 2011
Received in revised form 14 May 2012
Accepted 17 May 2012
Available online 30 May 2012

Keywords:
Nonlinear stability
Time-integration
Unconditionally stable
Isogeometric Analysis
Swift–Hohenberg
Rayleigh–Bénard convection
Pattern formation

ABSTRACT

The Swift–Hohenberg equation is a central nonlinear model in modern physics. Originally derived to describe the onset and evolution of roll patterns in Rayleigh–Bénard convection, it has also been applied to study a variety of complex fluids and biological materials, including neural tissues. The Swift–Hohenberg equation may be derived from a Lyapunov functional using a variational argument. Here, we introduce a new fully-discrete algorithm for the Swift–Hohenberg equation which inherits the nonlinear stability property of the continuum equation irrespectively of the time step. We present several numerical examples that support our theoretical results and illustrate the efficiency, accuracy and stability of our new algorithm. We also compare our method to other existing schemes, showing that is feasible alternative to the available methods.

1. Introduction

The Swift–Hohenberg model is an evolutive nonlinear higher-order partial differential equation (PDE) which develops complicated dynamics. Since it was proposed in the late seventies [1] as a model for the description of Rayleigh–Bénard convection [2,3], it has become one of the paradigms of nonlinear dynamical system leading to complex pattern formation [4,5]. Apart from fluid convection, the Swift–Hohenberg equation has also been employed to describe complex fluids and biological tissues [6]. The Swift–Hohenberg equation may be derived from a Lyapunov functional using a variational argument, which endows the theory with a nonlinear stability property. If inadequate algorithms are employed, this important property of the model can be lost after numerical discretization, leading to non-physical solutions. For example, an standard explicit method would require $\Delta t \sim \Delta x^4$ for the discrete solution to be energy-decreasing. This imposes a severe restriction over the periods of time that can be simulated. Here we present a fully discrete algorithm that inherits the nonlinear stability of the continuum model irrespectively of the mesh and time step sizes (in what follows, an algorithm verifying this property will be called unconditionally stable or thermodynamically consistent). Thus, our algorithm eliminates the restriction over the time step and opens the possibility to perform simulations over very large periods of time with the additional guarantee that the numerical solution will satisfy an important physical property of the model.

Thermodynamically consistent algorithms have been extensively studied in solid [7–10] and fluid mechanics [11–14], but remain rather unexplored for complex pattern-forming PDE's (some exceptions may be found in [15–24]). Previous work on the numerical simulation of the Swift–Hohenberg equation includes [25–31]. Remarkably, thermodynamically consistent algorithms for the Swift–Hohenberg equation have been proposed in [32,20]. Here we present a new space–time

* Corresponding author.
E-mail address: hgomez@udc.es (H. Gomez).

1007-5704/$ - see front matter © 2012 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.cnsns.2012.05.018

discretization for the Swift–Hohenberg equation that is unconditionally stable and second-order time-accurate. The space discretization of our algorithm is based on variational forms whose well-posedness requires the use of globally $\mathcal{C}^1$-continuous basis functions. We satisfy this requirement using Isogeometric Analysis [33,34], a recently proposed generalization of the Finite Element Method, that permits generating higher-order and higher-continuity basis functions. Compared to standard finite element formulations based on mixed methods, our method leads to half of the global number of degrees of freedom and exhibits better approximability properties than widely used $\mathcal{C}^0$-continuous piecewise polynomials [35]. For these reasons, our method is very attractive for the simulation of extended systems over large periods of time.

We present several numerical examples that support our theorems proving second-order time accuracy and unconditional stability. These computations are related to fluid convection on square and circular domains. The outline of this paper is as follows: In Section 2, we describe the Swift–Hohenberg equation. Section 3 presents our algorithm for this equation. We present numerical examples in Section 4. Finally, we draw conclusions in Section 5.

2. The Swift–Hohenberg equation

The Swift–Hohenberg equation describes the onset and evolution of roll patterns in Rayleigh–Bénard convection [36–39]. It may be derived from the fundamental equations of fluid mechanics in the limit of large Prandtl number, under the assumption of the Boussinesq approximation [1]. However, for the purpose of this work, it is more useful to derive it as a dissipative evolution equation of a non-conserved phase variable. By dissipative equation, we understand one for which a Lyapunov functional exists. In what follows, we introduce the Lyapunov functional of the Swift–Hohenberg equation, and derive an evolution equation whose solutions lead to a time-decreasing Lyapunov functional.

2.1. Lyapunov functional

Let u be a scalar phase variable defined on Ω, an open subset of $\mathbb{R}^3$. Let us call Γ the boundary of Ω. We assume Γ to have a continuous unit outward normal $\mathbf{n}$. We define the following free-energy functional

$$\mathcal{F}(u) = \int_\Omega \left\{ \Psi(u) + \frac{D}{2}\left[(\Delta u)^2 - 2k^2|\nabla u|^2 + k^4 u^2 \right] \right\} \mathrm{dx} \tag{1}$$

where D and k are real constants, and Ψ is a nonlinear function of u, defined as

$$\Psi(u) = -\frac{\epsilon}{2}u^2 - \frac{g}{3}u^3 + \frac{1}{4}u^4 \tag{2}$$

Here ϵ and g are positive constants, which represent physically relevant quantities. In what follows, we introduce the Swift–Hohenberg model, and show that $\mathcal{F}$ is indeed a Lyapunov functional of the equation.

2.2. The Swift–Hohenberg equation

The Swift–Hohenberg equation may be written as

$$\frac{\partial u}{\partial t} = -\frac{\delta \mathcal{F}}{\delta u} \tag{3}$$

where

$$\frac{\delta \mathcal{F}}{\delta u} = \Psi'(u) + Dk^4 u + 2Dk^2 \Delta u + D\Delta^2 u \tag{4}$$

denotes the variational derivative of $\mathcal{F}$ with respect to variations δu that verify $\delta u = \nabla(\delta u)\cdot \mathrm{n} = 0$ on Γ. Let us introduce the real-valued function F defined as $F(t) = \mathcal{F}(u(\cdot, t))$. Multiplying Eq. (3) with $\delta\mathcal{F}/\delta u$, and integrating over the domain Ω, we obtain the expression

$$\frac{\mathrm{d}F}{\mathrm{d}t} = -\int_\Omega \left(\frac{\delta \mathcal{F}}{\delta u} \right)^2 \mathrm{dx} \tag{5}$$

which leads to the inequality

$$\frac{\mathrm{d}F}{\mathrm{d}t} \leqslant 0. \tag{6}$$

The expression (6) may be thought of as a purely mechanical version of the Clausius–Duhem inequality [40] (continuum version of the second law of thermodynamics), and we consider it the fundamental stability property of the Swift–Hohenberg equation. The objective of this paper is to develop a fully discrete numerical method which inherits this property irrespectively of the mesh and time step sizes.

2.3. Initial/boundary-value problem

We state the following initial/boundary-value problem for the Swift–Hohenberg equation over the spatial domain Ω and the time interval $(0,T)$: given $u_0 : \overline{\Omega} \mapsto \mathbb{R}$, find $u : \overline{\Omega} \times [0,T] \mapsto \mathbb{R}$ such that

$$\frac{\partial u}{\partial t} = -\mu(u) - Dk^4 u - 2Dk^2 \Delta u - D\Delta^2 u \quad \text{in } \Omega \times (0,T) \tag{7}$$

$$\nabla(2Dk^2 u + D\Delta u) \cdot \mathrm{n} = 0 \quad \text{on } \Gamma \times [0,T] \tag{8}$$

$$\nabla u \cdot \mathrm{n} = 0 \quad \text{on } \Gamma \times [0,T] \tag{9}$$

$$u(\mathbf{x},0) = u_0(\mathbf{x}) \quad \text{in } \overline{\Omega} \tag{10}$$

where $\mu(u)$ stands for $\Psi'(u)$. Eqs. (8) and (9) may be considered natural boundary conditions of the Swift–Hohenberg equation in a variational formulation. In practice, most calculations involving the Swift–Hohenberg equation are performed using periodic boundary conditions in all directions.

3. Numerical formulation for the Swift–Hohenberg equation

In this section we present our numerical formulation for the Swift–Hohenberg equation. We first derive a semidiscrete form, and then introduce our unconditionally stable time-integration scheme.

3.1. Semidiscrete formulation

Our starting point is the weak formulation of the continuous problem. At this point we assume periodic boundary conditions in all directions. Let us call $\mathcal{V}$ the space of trial and weighting functions which are assumed to be the same. We suppose $\mathcal{V} \subset \mathcal{H}^2$, where $\mathcal{H}^2$ is the Sobolev space of square integrable periodic functions with square integrable first and second derivatives. The problem may be stated as follows: find $u \in \mathcal{V}$ such that for all $w \in \mathcal{V}$

$$\left(w, \frac{\partial u}{\partial t} + \mu(u) + Dk^4 u\right) - \left(\nabla w, 2Dk^2 \nabla u\right) + (\Delta w, D\Delta u) = 0 \tag{11}$$

where $(\cdot,\cdot)$ is the $\mathcal{L}^2$-inner product with respect to the domain Ω.To perform the space discretization of (11) we employ Galerkin's method. We approximate (11) by the following finite-dimensional problem over the finite element space $\mathcal{V}^h \subset \mathcal{V}$: find $u^h \in \mathcal{V}^h$ such that for all $w^h \in \mathcal{V}^h$

$$\left(w^h, \frac{\partial u^h}{\partial t} + \mu(u^h) + Dk^4 u^h\right) - \left(\nabla w^h, 2Dk^2 \nabla u^h\right) + \left(\Delta w^h, D\Delta u^h\right) = 0 \tag{12}$$

We define the discrete space $\mathcal{V}^h$ as $\mathcal{V}^h = \mathrm{span}\{N_A\}_{A=1,\ldots,n_b}$, where the N_A's are basis functions yet to be defined, and n_b is the dimension of the discrete space. As a consequence, u^h in Eq. (12) may be written as,

$$u^h(\mathrm{x},t) = \sum_{A=1}^{n_b} u_A(t) N_A(\mathrm{x}) \tag{13}$$

where the u_A's are the coordinates of u^h on $\mathcal{V}^h$. The function w^h is defined analogously. We emphasize that the condition $\mathcal{V}^h \subset \mathcal{V}$ requires the discrete space to be a subset of $\mathcal{H}^2$. Standard $\mathcal{C}^0$-continuous finite elements do not satisfy this requirement, and may not be utilized directly in the variational formulation (12). To handle this situation, we employ Isogeometric Analysis [33,34], which is a generalization of Finite Element Analysis [41] possessing several advantages [42–52]. Isogeometric Analysis is a recently introduced technology that is based on the developments of Computer Aided Design (CAD). Ideally, it would permit generating computational meshes directly from geometrical models encapsulated in CAD files, making use of the underlying parametrization of the CAD design. This holds promise to simplify, or even eliminate altogether, the mesh generation and refinement process, currently the major bottleneck of analysis. Following the isoparametric concept, the geometrical parametrization is also employed to generate the discrete space used to approximate the solution. Geometrical models in CAD files are usually parametrized using Non-Uniform Rational B-Splines (NURBS). NURBS are projective transformations of B-Splines, which, in turn, are piecewise polynomials [53,54]. NURBS not only permit generating computational models from CAD designs, but have also shown superior approximation capabilities compared to classical piecewise polynomials [50]. Even more importantly for this paper, the use of NURBS permits generating globally $\mathcal{C}^1$-continuous basis functions easily, which leads to simple treatment of higher-order operators and has proved significantly accurate and robust [18,35,55,56]. In what follows we show how to generate our basis functions and discrete spaces.

3.2. Basis functions and discrete space

Here we show how to generate the NURBS basis functions that we employ for spatial discretization. The first step is to define a one-dimensional B-Spline basis in parametric space. A B-Spline basis is a set of n piecewise polynomial functions

of order p denoted by $\{B_{i,p}\}_{i=1,\dots,n}$. These functions are defined from a knot vector, which is an array containing $n+p+1$ non-decreasing coordinates in parametric space called knots. We consider the knot vector

$$\mathbf{K}_\xi = \{\xi_1, \xi_2, \dots, \xi_{n+p+1}\} \tag{14}$$

which defines the parametric space $[\xi_1, \xi_{n+p+1}]$. Since the functions will be eventually mapped into physical space, we may assume without loss of generality $\xi_1 = 0$ and $\xi_{n+p+1} = 1$. Given a knot vector, the B-Spline basis functions of order p are defined recursively from their lower-order counterparts. The process is started with the zero-th order functions $\{B_{i,0}\}_{i=1,\dots,n}$ given by

$$B_{i,0}(\xi) = \begin{cases} 1 & \text{if} \quad \xi_i \leqslant \xi \leqslant \xi_{i+1} \\ 0 & \text{otherwise} \end{cases} \tag{15}$$

Then, the following algorithm is applied

$$B_{i,a}(\xi) = \frac{\xi - \xi_i}{\xi_{i+p} - \xi_i} B_{i,a-1}(\xi) + \frac{\xi_{i+p+1} - \xi}{\xi_{i+p+1} - \xi_{i+1}} B_{i+1,a-1}(\xi); \quad i = 1, \dots, n;\ a = 1, \dots, p \tag{16}$$

The functions $\{B_{i,p}\}_{i=1,\dots,n}$ are $\mathcal{C}^\infty$ everywhere except at knots. At a non-repeated knot, the functions have $p-1$ continuous derivatives. If a knot is repeated k times the number of continuous derivatives at that point is $p-1-k$.

A three-dimensional B-Spline basis is defined taking tensor products of one-dimensional basis in three orthogonal parametric directions. Therefore, given three polynomial orders p, q, r, and three knot vectors, $\mathbf{K}_\xi$, $\mathbf{K}_\eta$, $\mathbf{K}_\zeta$ of lengths $n+p+1,\ m+q+1,\ l+r+1$, we can compute

$$B_{ijk}(\xi, \eta, \zeta) = B_{i,p}(\xi) B_{j,q}(\eta) B_{k,r}(\zeta) \tag{17}$$

Analogously to the one-dimensional case, the knot vectors define the parametric space, which we denote Ξ. Again, for simplicity, we take $\Xi = [0,1]^3$. Using the three-dimensional B-Spline basis functions we can generate a geometric mapping $\mathbf{F} : \Xi \mapsto \Omega$

$$\mathbf{F}(\xi, \eta, \zeta) = \sum_{i=1}^{n+p+1} \sum_{j=1}^{m+q+1} \sum_{k=1}^{l+r+1} \mathbf{C}_{ijk} B_{ijk}(\xi, \eta, \zeta) \tag{18}$$

which defines the geometric object Ω. The values $\mathbf{C}_{ijk} \in \mathbb{R}^3$ are called control variables.At this point, we may define NURBS geometrical objects in $\mathbb{R}^d$, which are projective transformations of B-Spline geometrical entities in $\mathbb{R}^{(d+1)}$. Let $\widehat{\mathbf{C}}_{ijk} \in \mathbb{R}^3$ be a set of control points in three-dimensional space and ω_{ijk} a set of positive real numbers called weights such that $(\widehat{\mathbf{C}}_{ijk}, w_{ijk}) \in \mathbb{R}^4$. We define the following B-Spline geometrical object in $\mathbb{R}^4$ as,

$$\widehat{\Omega} = \widehat{\mathbf{F}}(\Xi) \tag{19}$$

where

$$\widehat{\mathbf{F}}(\xi, \eta, \zeta) = \sum_{i=1}^{n+p+1} \sum_{j=1}^{m+q+1} \sum_{k=1}^{l+r+1} (\widehat{\mathbf{C}}_{ijk}, w_{ijk}) B_{ijk}(\xi, \eta, \zeta), \quad (\xi, \eta, \zeta) \in \Xi \tag{20}$$

The NURBS object Ω_R is defined as

$$\Omega_R = \mathbf{F}_R(\Xi) \tag{21}$$

where the geometrical mapping $\mathbf{F}_R$ takes the form,

$$\mathbf{F}_R(\xi, \eta, \zeta) = \sum_{i=1}^{n+p+1} \sum_{j=1}^{m+q+1} \sum_{k=1}^{l+r+1} \frac{\widehat{\mathbf{C}}_{ijk}}{w_{ijk}} \frac{w_{ijk} B_{ijk}(\xi, \eta, \zeta)}{\sum_{a=1}^{n+p+1} \sum_{b=1}^{m+q+1} \sum_{c=1}^{l+r+1} w_{abc} B_{abc}(\xi, \eta, \zeta)}, \quad (\xi, \eta, \zeta) \in \Xi \tag{22}$$

Denoting,

$$\mathbf{C}_{ijk} = \frac{\widehat{\mathbf{C}}_{ijk}}{w_{ijk}}, \quad W(\xi, \eta, \zeta) = \sum_{i=1}^{n+p+1} \sum_{j=1}^{m+q+1} \sum_{k=1}^{l+r+1} w_{ijk} B_{ijk}(\xi, \eta, \zeta), \quad R_{ijk}(\xi, \eta, \zeta) = \frac{w_{ijk} B_{ijk}(\xi, \eta, \zeta)}{W(\xi, \eta, \zeta)} \tag{23}$$

we have that

$$\mathbf{F}_R(\xi, \eta, \zeta) = \sum_{i=1}^{n+p+1} \sum_{j=1}^{m+q+1} \sum_{k=1}^{l+r+1} \mathbf{C}_{ijk} R_{ijk}(\xi, \eta, \zeta), \quad (\xi, \eta, \zeta) \in \Xi \tag{24}$$

We will call R_{ijk} NURBS functions in parametric space.NURBS functions in physical space are defined as the push forward of the functions R_{ijk}. Thus, the discrete space that we use for our numerical method is the space spanned by those functions, namely

$$\mathcal{V}^h = \text{span}\{R_{ijk} \circ \mathbf{F}_R^{-1}\} \tag{25}$$

Note that we invoke the isoparametric concept, because the geometrical mapping $\mathbf{F}_R$ is defined in terms of NURBS functions.

3.3. Time integration

Here we present our time integration algorithm for the Swift–Hohenberg equation. Let us divide the time interval $[0,T]$ into N subintervals $\mathcal{I}_n = (t_n, t_{n+1});\ n = 0,\dots,N-1$, where $t_0 = 0$ and $t_N = T$. We call u_n^h the discrete approximation of $u^h(t_n)$, where we have omitted the dependence on the spatial coordinate for simplicity. Our time stepping algorithm is defined as follows: given u_n^h, find $u_{n+1}^h \in \mathcal{V}^h$ such that for all $w^h \in \mathcal{V}^h$

$$\left(w^h, \frac{[\![u_n^h]\!]}{\Delta t_n}\right) + \left(w^h, \frac{1}{2}\left(\mu(u_{n+1}^h) + \mu(u_n^h)\right) - \frac{[\![u_n^h]\!]^2}{12}\mu''(u_n^h)\right) + \left(w^h, Dk^4 u_{n+1/2}^h\right) - \left(\nabla w^h, 2Dk^2 \nabla u_{n+1/2}^h\right) + \left(\Delta w^h, D\Delta u_{n+1/2}^h\right) = 0 \tag{26}$$

where

$$\Delta t_n = t_{n+1} - t_n; \quad [\![u_n^h]\!] = u_{n+1}^h - u_n^h; \quad u_{n+1/2}^h = \frac{1}{2}\left(u_{n+1}^h + u_n^h\right) \tag{27}$$

We summarize the main properties of our time integration scheme in the following theorem.

Theorem 1. *The fully-discrete variational formulation (26):*

(1) *Verifies the nonlinear stability condition*

$$\mathcal{F}(u_n^h) \leqslant \mathcal{F}(u_{n-1}^h) \quad \forall n = 1,\dots,N$$

irrespectively of the time step.

(2) *Gives rise to a local truncation error τ that may be bounded as $|\tau(t_n)| \leqslant K\Delta t_n^2$ for all $t_n \in [0,T]$, where K is a constant independent of Δt_n.*

Proof.

(1) Let $f : [a,b] \mapsto \mathbb{R}$ be a sufficiently smooth function. We will make use of the following quadrature formula:

$$\int_a^b f(x)\mathrm{d}x = \frac{b-a}{2}(f(a)+f(b)) - \frac{(b-a)^3}{12}f''(a) - \frac{(b-a)^4}{24}f''''(\xi);\ \xi \in (a,b) \tag{28}$$

which was introduced in [18]. Let us apply the quadrature formula (28) to the right-hand side of the identity

$$\int_{u_n^h}^{u_{n+1}^h} \Psi'(z)\mathrm{d}z = \int_{u_n^h}^{u_{n+1}^h} \mu(z)\mathrm{d}z \tag{29}$$

and rearrange the resulting equation. It follows that

$$\frac{[\![\Psi(u_n^h)]\!]}{[\![u_n^h]\!]} + \frac{[\![u_n^h]\!]^3}{24}\mu'''(u_{n+\xi}^h) = \frac{1}{2}\left(\mu(u_n^h) + \mu(u_{n+1}^h)\right) - \frac{[\![u_n^h]\!]^2}{12}\mu''(u_n^h); \quad \xi \in (0,1) \tag{30}$$

Taking $w^h = [\![u_n^h]\!]$ in Eq. (26), applying Eq. (30), and making use of the identities

$$\left([\![u_n^h]\!], u_{n+1/2}^h\right) = \frac{1}{2}\int_\Omega [\![(u_n^h)^2]\!]\mathrm{d}x; \quad \left(\nabla[\![u_n^h]\!], \nabla u_{n+1/2}^h\right) = \frac{1}{2}\int_\Omega [\![|\nabla u_n^h|^2]\!]\mathrm{d}x \tag{31}$$

$$\left(\Delta[\![u_n^h]\!], \Delta u_{n+1/2}^h\right) = \frac{1}{2}\int_\Omega [\![(\Delta u_n^h)^2]\!]\mathrm{d}x \tag{32}$$

we obtain the following relation

$$\frac{1}{\Delta t_n}\int_\Omega [\![u_n^h]\!]^2\mathrm{d}x + \int_\Omega [\![\Psi(u_n^h)]\!]\mathrm{d}x + \int_\Omega \frac{[\![u_n^h]\!]^4}{12}\mu'''(u_{n+\xi}^h)\mathrm{d}x + \int_\Omega \frac{Dk^4}{2}[\![(u_n^h)^2]\!]\mathrm{d}x - \int_\Omega Dk^2[\![|\nabla u_n^h|^2]\!]\mathrm{d}x + \int_\Omega \frac{D}{2}[\![(\Delta u_n^h)^2]\!]\mathrm{d}x = 0 \tag{33}$$

which may be rewritten as

$$[\![\mathcal{F}(u_n^h)]\!] = -\frac{1}{\Delta t_n}\int_\Omega [\![u_n^h]\!]^2 \mathrm{dx} - \int_\Omega \frac{[\![u_n^h]\!]^4}{12}\mu'''(u_{n+\xi}^h)\mathrm{dx} \tag{34}$$

Since $\mu'''(u) \geqslant 0$, it follows that

$$[\![\mathcal{F}(u_n^h)]\!] \leqslant 0 \tag{35}$$

which completes the proof.

(2) We derive a bound on the local truncation error by replacing the time-continuous solution $u^h(t_n)$ into the algorithm (26). The time-continuous solution does not satisfy the algorithm, giving rise to the local truncation error τ, which is defined by the expression

$$\begin{aligned}(w^h, \tau(t_n)) = &\left(w^h, \frac{[\![u^h(t_n)]\!]}{\Delta t_n}\right) + \left(w^h, \frac{1}{2}\left(\mu(u^h(t_{n+1})) + \mu(u^h(t_n))\right)\right) - \left(w^h, \frac{[\![u^h(t_n)]\!]^2}{12}\mu''(u^h(t_n))\right)\\ &+ \left(w^h, Dk^4 u^h(t_{n+1/2})\right) - \left(\nabla w^h, 2Dk^2\nabla u^h(t_{n+1/2})\right) + \left(\Delta w^h, D\Delta u^h(t_{n+1/2})\right)\end{aligned} \tag{36}$$

where $t_{n+1/2} = (t_{n+1} + t_n)/2$. Assuming sufficient smoothness, Taylor series can be utilized to prove that

$$\frac{[\![u^h(t_n)]\!]}{\Delta t_n} = \frac{\partial u^h}{\partial t}(t_{n+1/2}) + \mathcal{O}(\Delta t_n^2) \tag{37}$$

$$\frac{1}{2}\left(\mu(u^h(t_{n+1})) + \mu(u^h(t_n))\right) = \mu(u^h(t_{n+1/2})) + \mathcal{O}(\Delta t_n^2) \tag{38}$$

$$\frac{[\![u^h(t_n)]\!]^2}{12}\mu''(u^h(t_n)) = \mathcal{O}(\Delta t_n^2) \tag{39}$$

where we have made use of the Landau notation. Eqs. (36)–(39), together with (12) lead to the identity

$$(w^h, \tau(t_n)) = (w^h, \mathcal{O}(\Delta t_n^2)) \tag{40}$$

which indicates that $|\tau(t_n)| \leqslant K\Delta t_n^2$ where K is a constant independent of Δt_n. □

4. Numerical examples

In this section we present some numerical examples for the Swift–Hohenberg equation. Our calculations also provide numerical corroboration of the theoretical results presented in the previous sections. We also present a comparison of the performance of our new algorithm with other existing techniques. Finally, we present two examples related to the formation of roll patterns in Rayleigh–Bénard convection both in square and circular domains.

4.1. Accuracy test

This example provides numerical evidence for our time integration scheme being second-order accurate. The setup of this accuracy test is based on that presented in [20]. We solve the one-dimensional Swift–Hohenberg equation on the domain $\overline{\Omega} = [0, 32]$. The parameters of the Swift–Hohenberg equation are $D = k = 1,\ \epsilon = 0.025$, and $g = 0$. The initial condition is defined as:

$$u(x) = 0.07 - 0.02\cos\left(\frac{2\pi(x-12)}{32}\right) + 0.0171\cos^2\left(\frac{2\pi(x+10)}{32}\right) - 0.0085\sin^2\left(\frac{4\pi x}{32}\right) \tag{41}$$

We computed a reference solution at time $t = 1$ using a spatial mesh composed of 256 $\mathcal{C}^1$ quadratic elements, and a time step $\Delta t = 7.8125 \cdot 10^{-3}$. We assume that this space–time discretization is fine enough as to suppose that the reference solution is exact. Then, we repeated the computation using larger time steps, and studied how the $\mathcal{L}^2([0, 32])$ spatial error norm evolved as a function of Δt. The results are presented on a doubly logarithmic scale in Fig. 1. The data defines a straight line with slope 2.67, which confirms the results proven in Theorem 1.

4.2. Comparison with other methods

In this section, we compare the performance of our new numerical scheme with other well-established techniques. The performance of our spatial discretization, that is, NURBS functions in a variational formulation, has been shown superior to standard finite elements on a per-degree-of-freedom basis in several publications [42,48,50,51]. For this reason, we will focus on the time discretization algorithm. We take a spatial mesh which is sufficiently fine as to suppose that all the error can be attributed to time integration. Then, we compare the performance of our new time integration method with the following algorithms: (1) a semi-implicit method that treats the nonlinear terms explicitly and the linear terms implicitly;

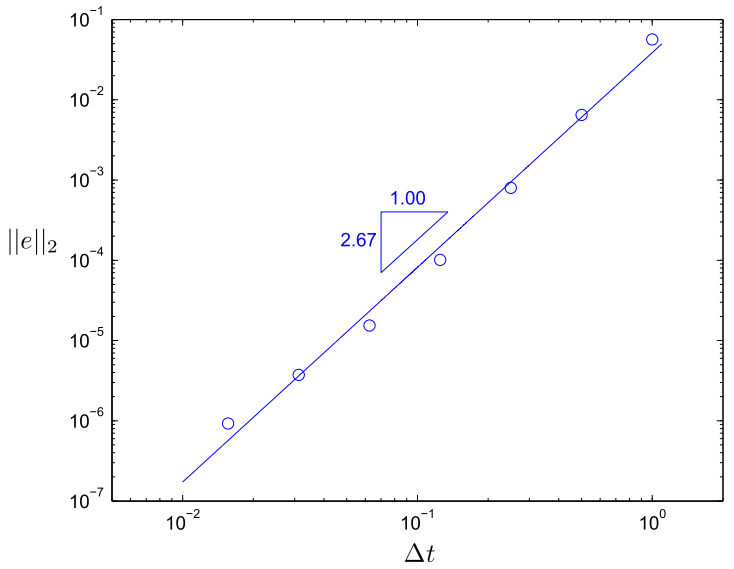

Fig. 1. Accuracy test. $\mathcal{L}^2([0,32])$ spatial error norm with respect to the time step Δt. This plot confirms that our algorithm is second-order time-accurate.

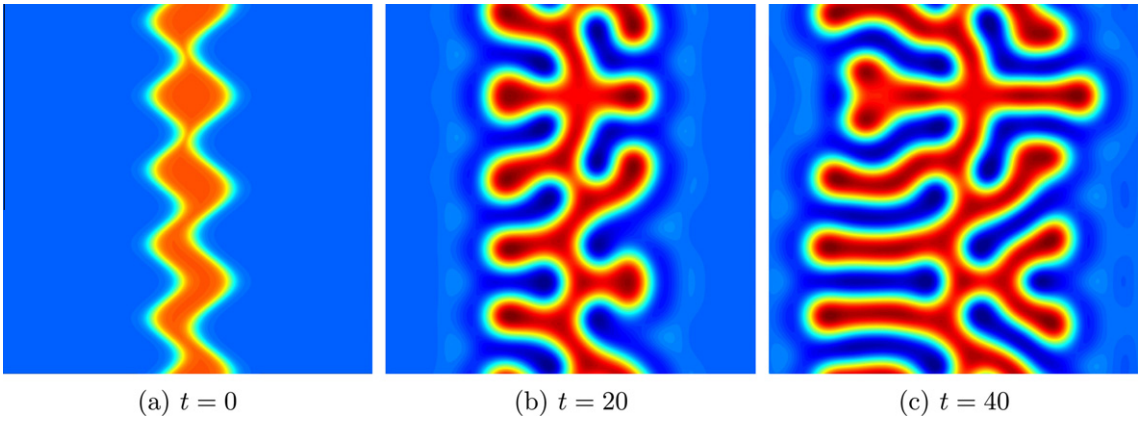

Fig. 2. Comparison with other methods. Initial condition and reference solution at different times. The computational domain is $\overline{\Omega} = [0,40]^2$. This solution has been computed on a sufficiently small space–time mesh.

(2) the explicit first-order accurate exponential time integrator presented in [57]; (3) the convex-splitting method proposed in [20] that has been shown to be unconditionally stable for the Swift–Hohenberg equation; (4) the midpoint rule.

To perform the comparison we solve the Swift–Hohenberg equation on the domain $\overline{\Omega} = [0,40]^2$. The parameters are $D = k = 1,\ \epsilon = 2$, and $g = 0$. The initial condition is a constant state ($u = -1$) in which we embed a curvy vertical stripe with the phase variable taking the value $u = 1$. The initial pattern evolves developing horizontal fingers that might bifurcate. Fig. 2 shows the initial condition and solution at times $t = 20$ and $t = 40$ on a sufficiently fine space–time mesh. The color scale ranges from -1.75 (blue[1]) to $+1.75$ (red), and will be maintained for all the examples in this section. We consider the solution in Fig. 2 as our reference solution. We will compare the results produced by all the above-mentioned methods at time $t = 40$.

Fig. 3 shows the results produced by the semi-implicit scheme at time $t = 40$ using different time steps. For $\Delta t = 0.25$ the solution is totally incorrect. Qualitative agreement is obtained for $\Delta t = 0.125,\ \Delta t = 0.0625$, and $\Delta t = 0.03125$. For smaller time steps, the solution is almost indistinguishable from the reference solution.

Fig. 4 presents the free energy evolution for the semi-implicit method using different time steps. For $\Delta t = 0.25$, the free energy is oscillating, which indicates that the computed solution is incorrect. For smaller time steps, the energy does decrease in time, but the dissipation rate is somewhat underestimated. For the two smallest time steps ($\Delta t = 0.015625$ and $\Delta t = 0.0078125$), the energy curves are one on top of each other, which is a sign of numerical convergence.

[1] For interpretation of colour in Fig. 2, the reader is referred to the web version of this article.

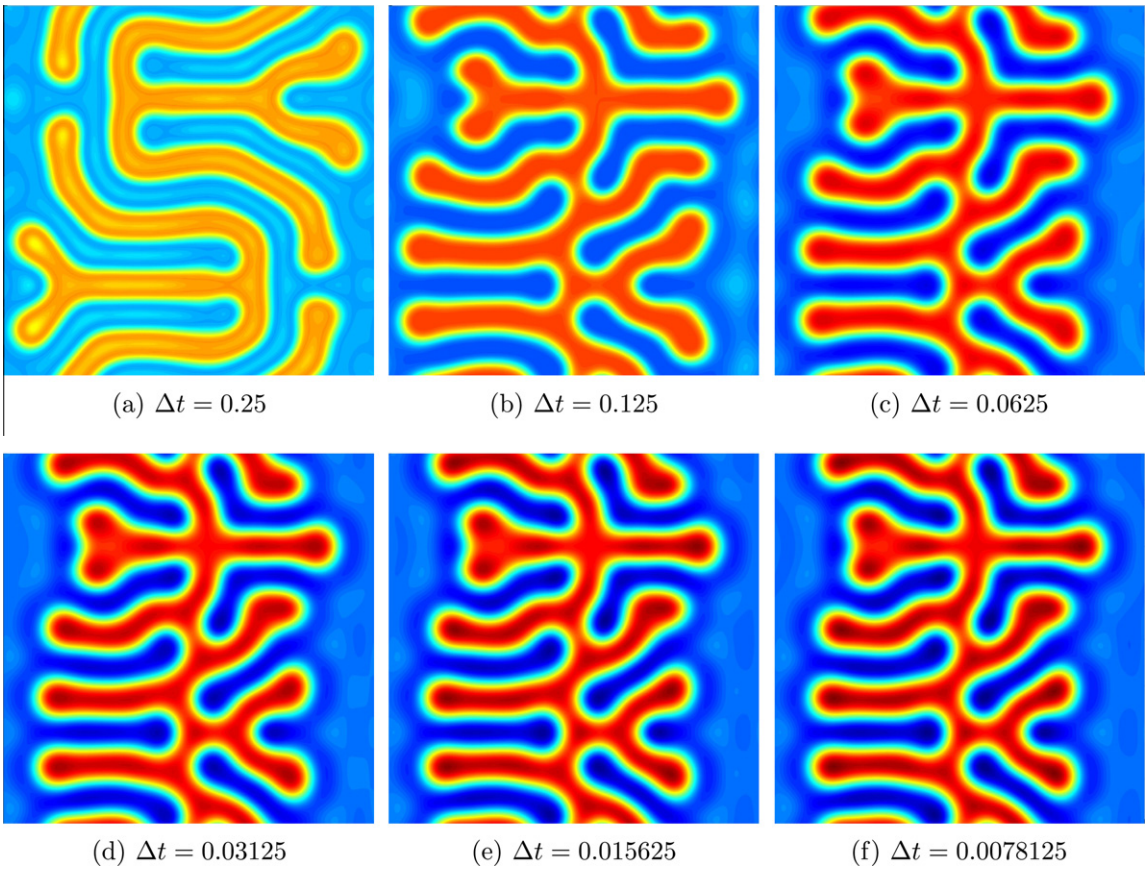

Fig. 3. Comparison with other methods. Numerical solution at $t = 40$ with the semi-implicit time integrator using different time steps.

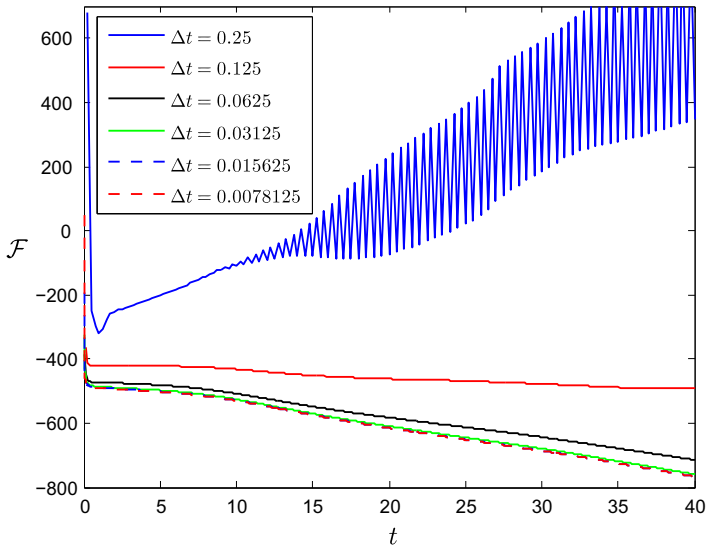

Fig. 4. Comparison with other methods. Energy evolution for several time steps using the semi-implicit algorithm.

Fig. 5 shows the solution at time $t = 40$ using the exponential time integrator and different time steps. This method permits taking time steps larger than those used for the semi-implicit algorithm. The solution for $\Delta t = 0.5$ (Fig. 5(a)) is incorrect. This is also reflected in the energy plot (Fig. 6), which shows an oscillating evolution. For smaller time steps, the solution

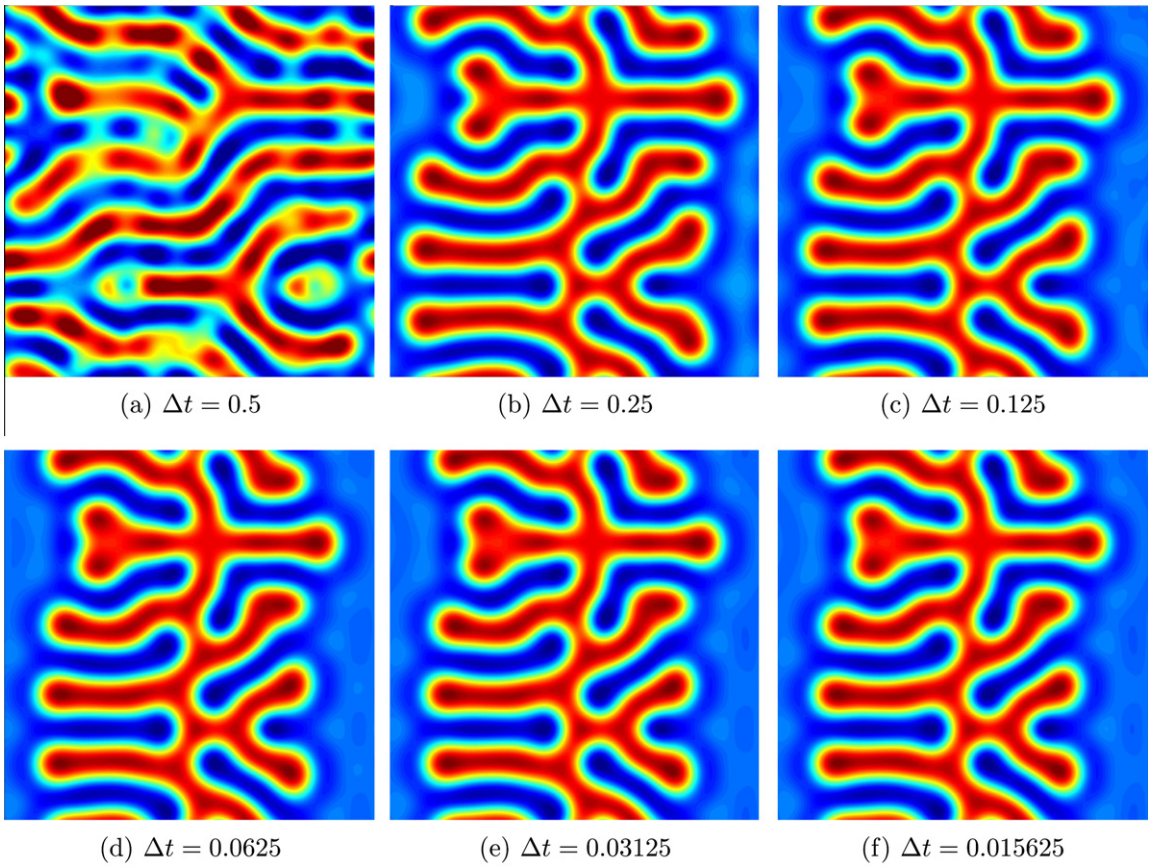

Fig. 5. Comparison with other methods. Numerical solution at $t = 40$ with the exponential time integrator using different time steps.

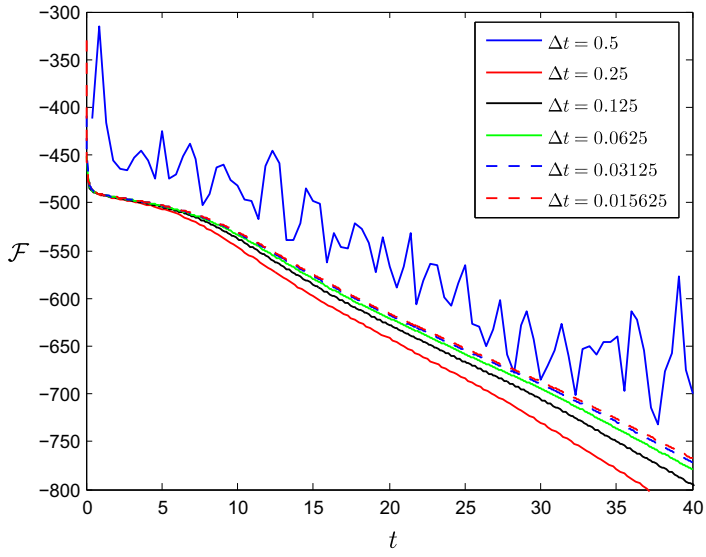

Fig. 6. Comparison with other methods. Energy evolution for several time steps using the exponential algorithm.

looks qualitatively correct, but quantitative match is only achieved for the two smallest time steps ($\Delta t = 0.03125$ and $\Delta t = 0.015625$). This can also be observed in the energy plot (Fig. 6) in which the curves corresponding to those time steps are superposed.

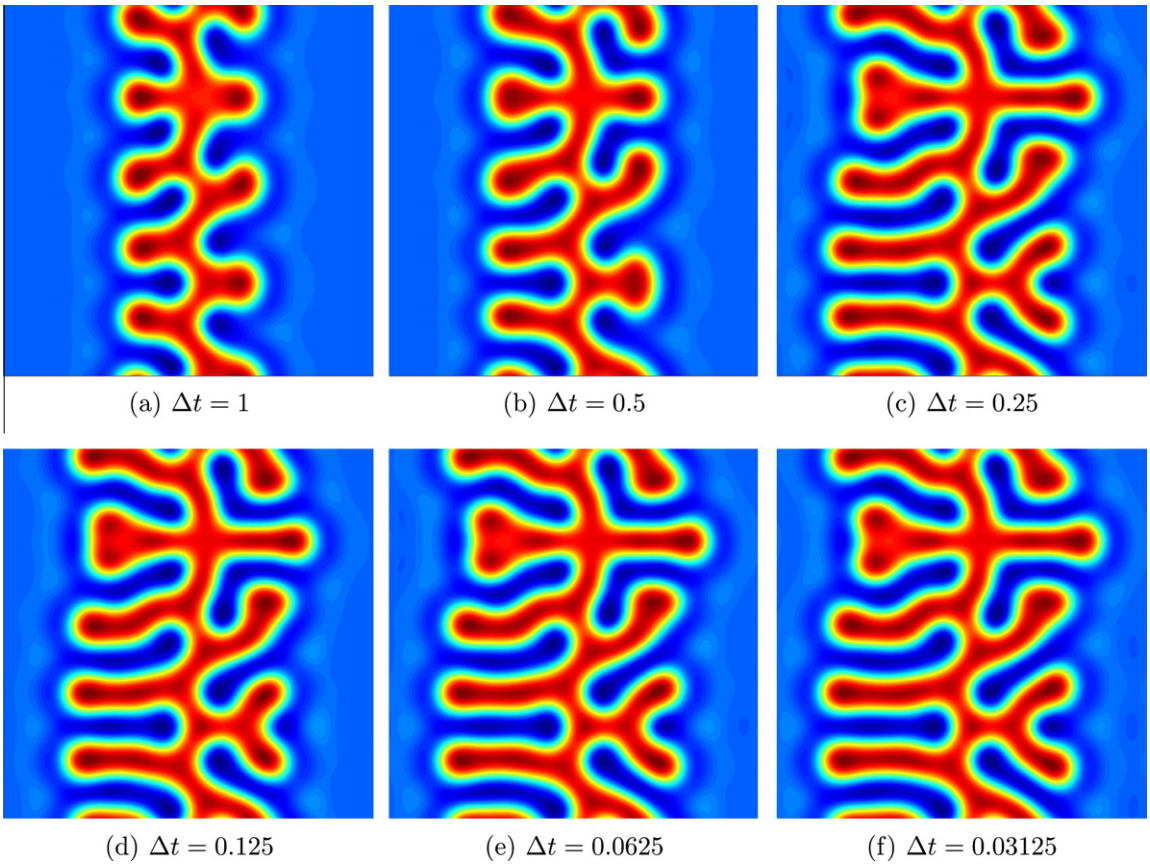

Fig. 7. Comparison with other methods. Numerical solution at $t = 40$ with the convex splitting algorithm using different time steps.

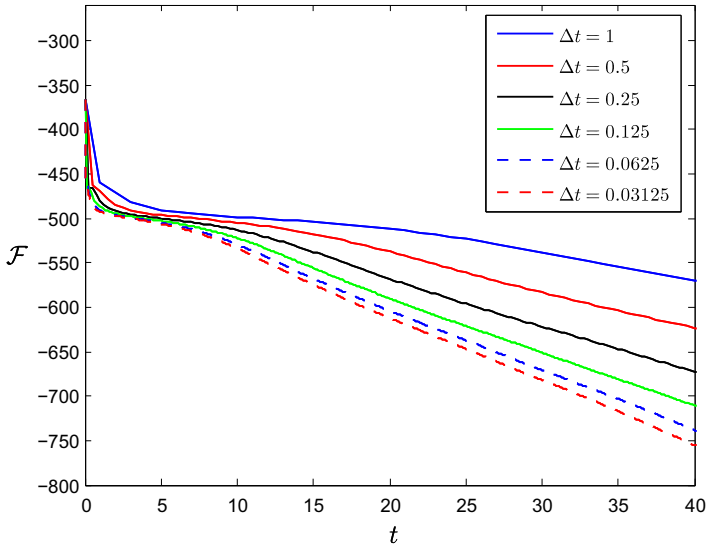

Fig. 8. Comparison with other methods. Energy evolution for several time steps using the convex splitting method.

Fig. 7 shows the solution using the convex-splitting algorithm proposed in [20]. This method has been shown to be unconditionally stable for the Swift–Hohenberg equation. This property is certainly reflected in the energy plot (Fig. 8), which shows time-decreasing energies for all time steps. However, it is know that algorithms based on the convex-splitting

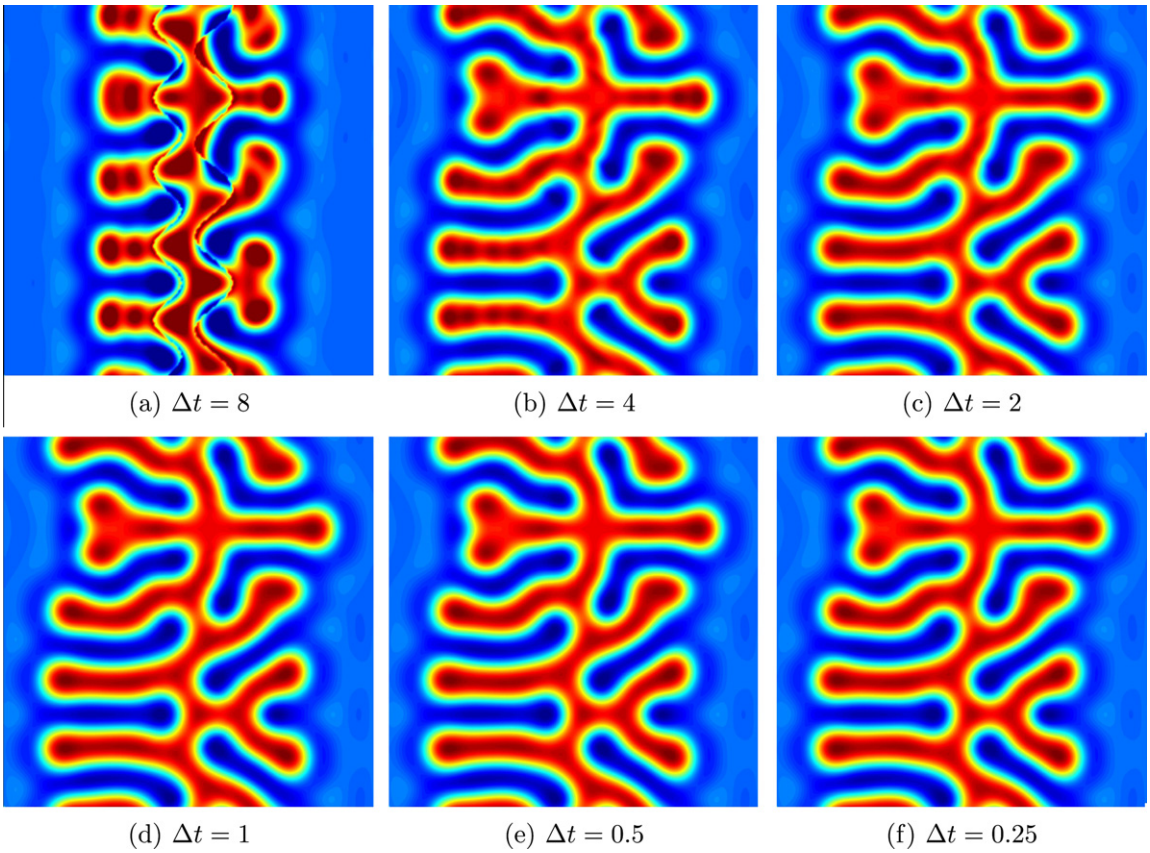

Fig. 9. Comparison with other methods. Numerical solution at $t = 40$ with the midpoint rule using different time steps.

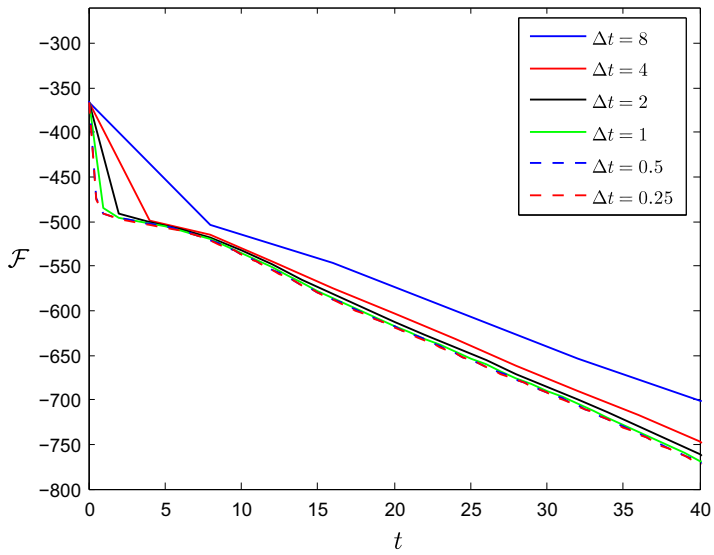

Fig. 10. Comparison with other methods. Energy evolution for several time steps using the midpoint rule.

concept tend to be significantly inaccurate for large time steps [18]. This statement is consistent with the results in Fig. 7, which show inaccurate solutions at least for $\Delta t = 1$ and $\Delta t = 0.5$. As we reduce the time step, the solution becomes more accurate, exhibiting good agreement with the reference solution for $\Delta t = 0.0625$.

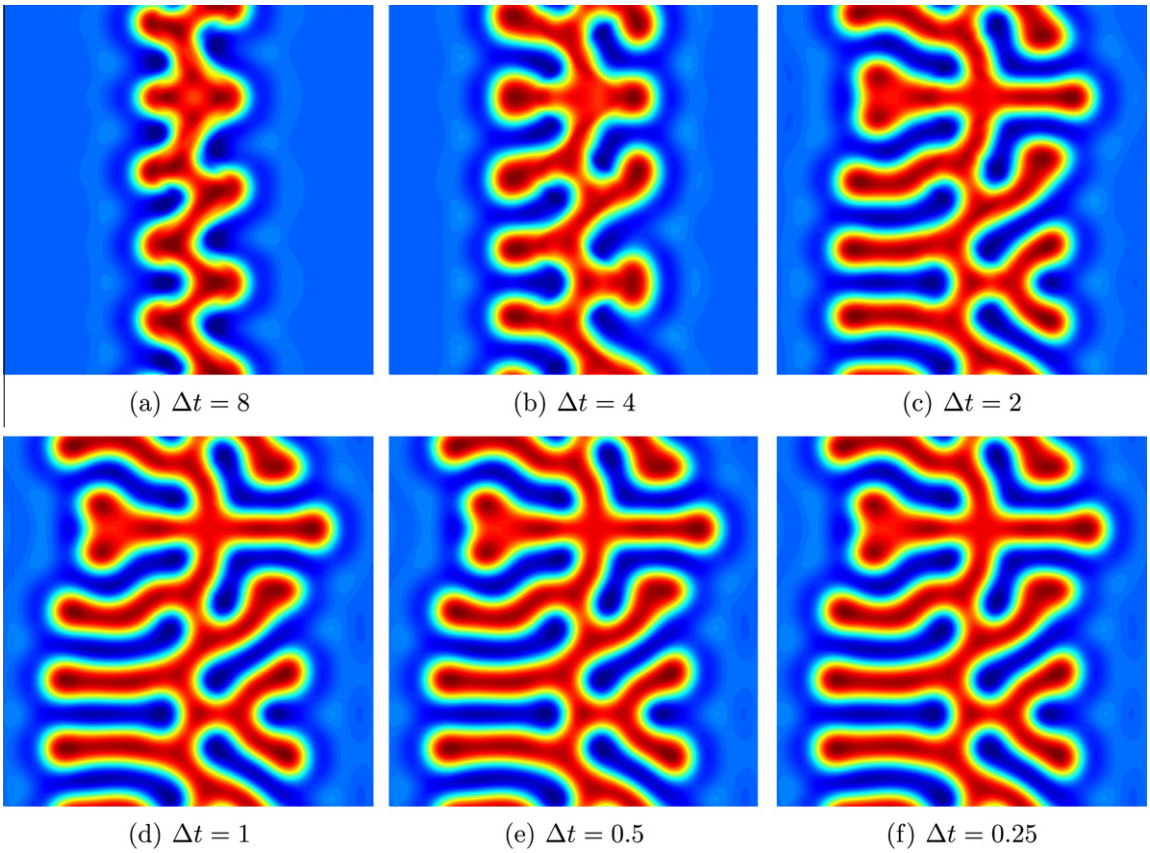

Fig. 11. Comparison with other methods. Numerical solution at $t = 40$ with the proposed algorithm using different time steps.

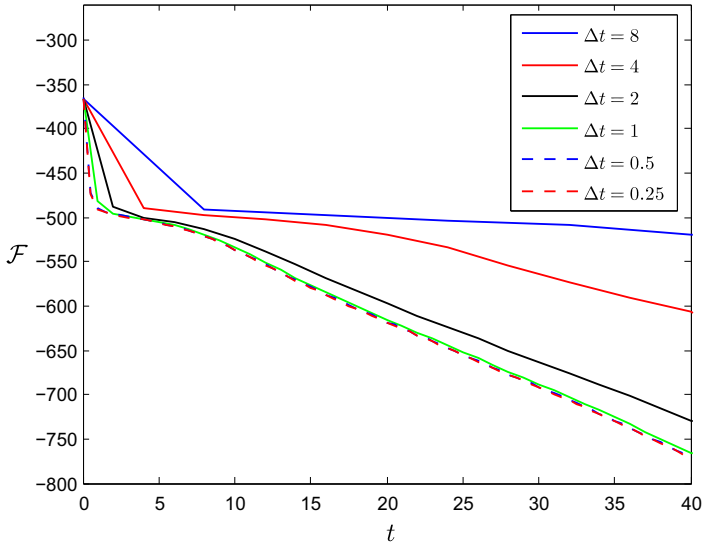

Fig. 12. Comparison with other methods. Energy evolution for several time steps using the proposed method. We observe that for $\Delta t = 1$, $\Delta t = 0.5$ and $\Delta t = 0.25$ the curves are almost superposed, which indicates numerical convergence.

Now, we analyze the results produced by the midpoint rule. This algorithm is an standard second-order accurate method. For a linear problem it is known to have the lowest truncation error of all second-order accurate *A*-stable linear multistep methods [59]. Another feature of the midpoint rule that holds for linear problems is that the algorithm preserves the highest

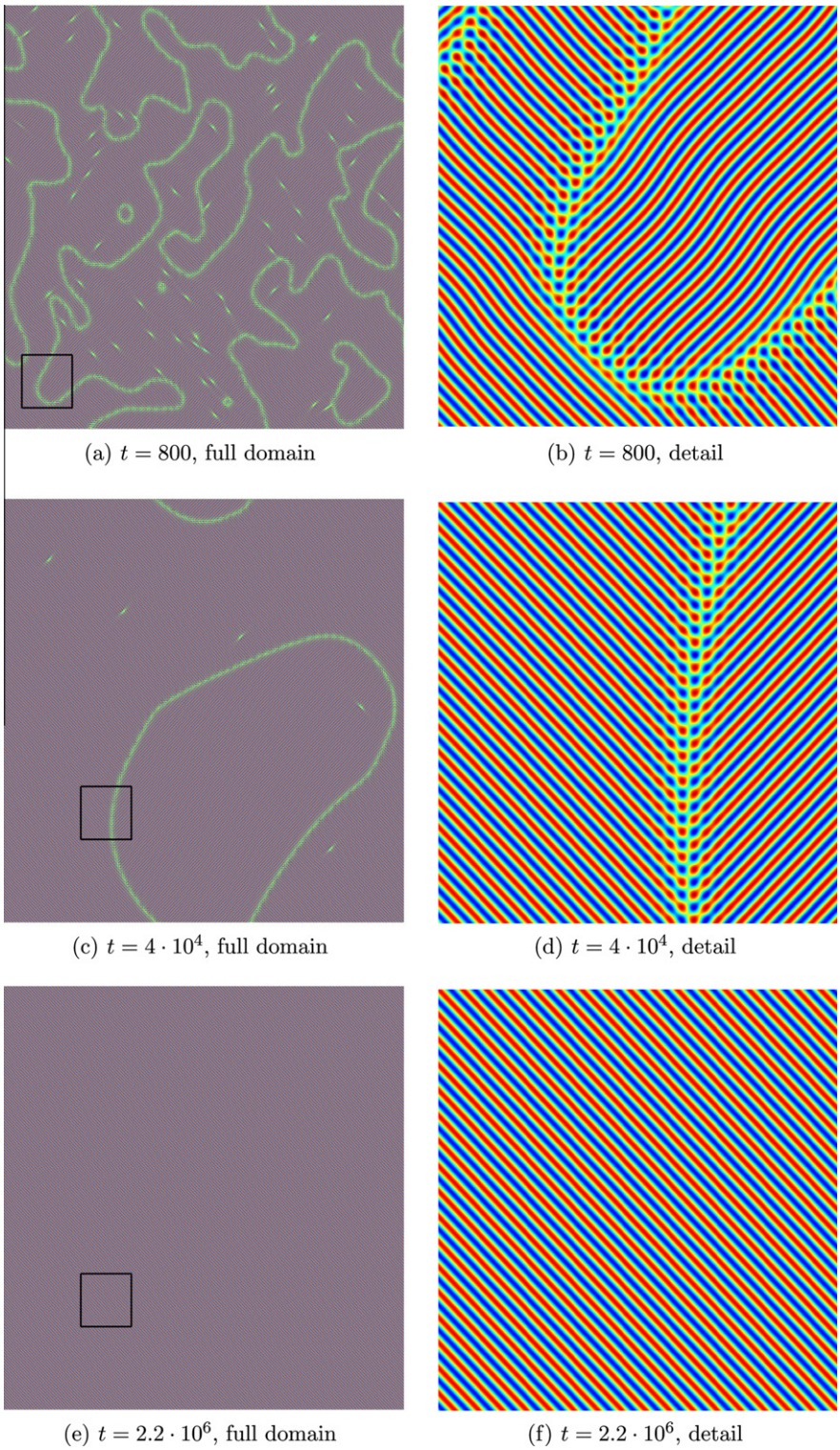

Fig. 13. Swift–Hohenberg equation on a periodic square. Snapshots of the numerical approximation to the phase variable u at different computational times. The parameters of the equation are $D = k = 1$, $\epsilon = 0.1$, and $g = 0$. The spatial mesh is composed of 2048^2 C^1-quadratic elements. The time step is $\Delta t = 20$.

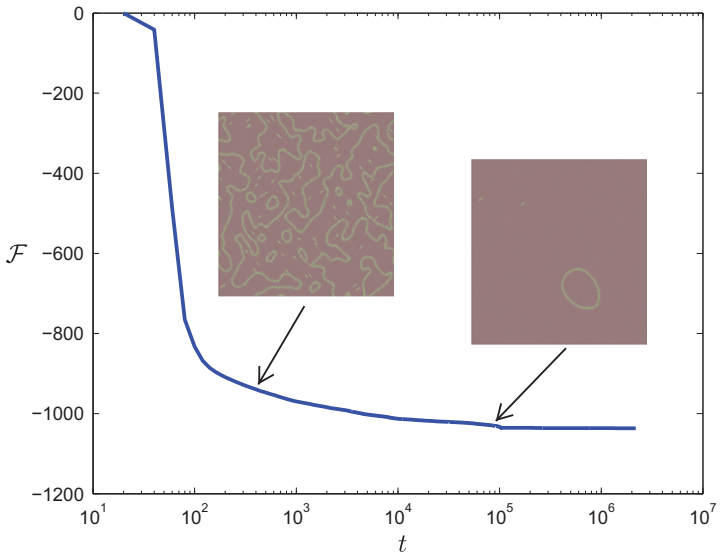

Fig. 14. Swift–Hohenberg equation on a periodic square. Time evolution of the energy functional. We appended some snapshots of the solution to the free-energy curve. We observe that the energy functional diminishes at all times, which confirms our theoretical predictions.

frequency of the numerical solution at each time step, which makes it difficult to damp out the spurious modes of the approximate solution. This feature of the method may be observed in Fig. 9(a), which represents the numerical solution at $t = 40$ using the time step $\Delta t = 8$. This image clearly shows how the high frequencies contained in the initial condition remain in the solution after several time steps, leading to a significantly inaccurate solution. For $\Delta t = 4$, Fig. 9(b), the solution looks still shaky, showing again the inability of the algorithm to handle the high frequencies of the solution. For smaller time steps the solution is accurate and smooth, and appears almost indistinguishable from the reference solution.

Fig. 10 shows the energy evolution produced by the midpoint rule for the analyzed time steps. All the curves are monotonically decreasing, which indicates that, for this particular problem and the selected time steps, the midpoint rule respects the stability property of the Swift–Hohenberg equation. However, this result does not necessarily hold for other problems or time steps, because the midpoint rule does not achieve unconditional stability for the Swift–Hohenberg equation.

Finally, Fig. 11 shows the solution using our new algorithm. We observe that our method permits using significantly larger time steps than the semi-implicit, the exponential and the convex-splitting algorithms. The time steps employed are comparable to those utilized for the midpoint rule. It may be said that for small and intermediate time steps our method produces results similar to those achieved by the midpoint rule. However, for very large time steps ($\Delta t = 8$), although the solution is inaccurate, is still smooth and its associated energy decreases with time. The numerical solution actually looks like the exact solution at an earlier time (compare Fig. 2 with Fig. 11(a)). Physically speaking, it may be said that the method defers the dynamics of the equation for large time steps. This seems to be a feature of unconditionally stable methods for nonlinear dynamics [18,20,58], and may be also observed in the results produced by the convex splitting method (Fig. 7). However, the convex-splitting scheme is significantly less accurate than our method. In all, we may conclude that our algorithm represents a good balance between accuracy and stability, with the additional guarantee of energy-decreasing solutions, as shown in Fig. 12. This plot also shows that the dissipation rate is underestimated for large time steps, which is consistent with the statement that the dynamics of the equation is deferred for large time steps.

4.3. Swift–Hohenberg equation on a periodic square

Here we present the numerical solution to the Swift–Hohenberg equation on the domain $\overline{\Omega} = [0, 1200]^2$. We assume periodic boundary conditions in both directions. We take the parameters $D = k = 1,\ \epsilon = 0.1,\ g = 0$. To define our initial condition, we set all control variables to zero, and then, randomly perturb their values with a pseudo-random number which is uniformly distributed on $[-0.005, 0.005]$. We employ an uniform spatial mesh composed of 2048^2 $\mathcal{C}^1$-quadratic elements. The time step is $\Delta t = 20$.

Fig. 13 shows snapshots of the time-history of the phase variable. We observe that the initial condition induces an instability into the equation that leads to amplification of the solution and to the emergence of spatial patterns. Those patterns are composed of stripped regions with zero- and one-dimensional defects as observed in Fig. 13(a) and (b). Global reorganization of the patterns leads to larger defect-free regions (Fig. 13(c) and (d)) which eventually fill up the whole domain. The simulation results suggest that for this set of parameters the equation presents a stationary solution corresponding to an ordered stripped pattern as shown in Fig. 13(e) and (f).

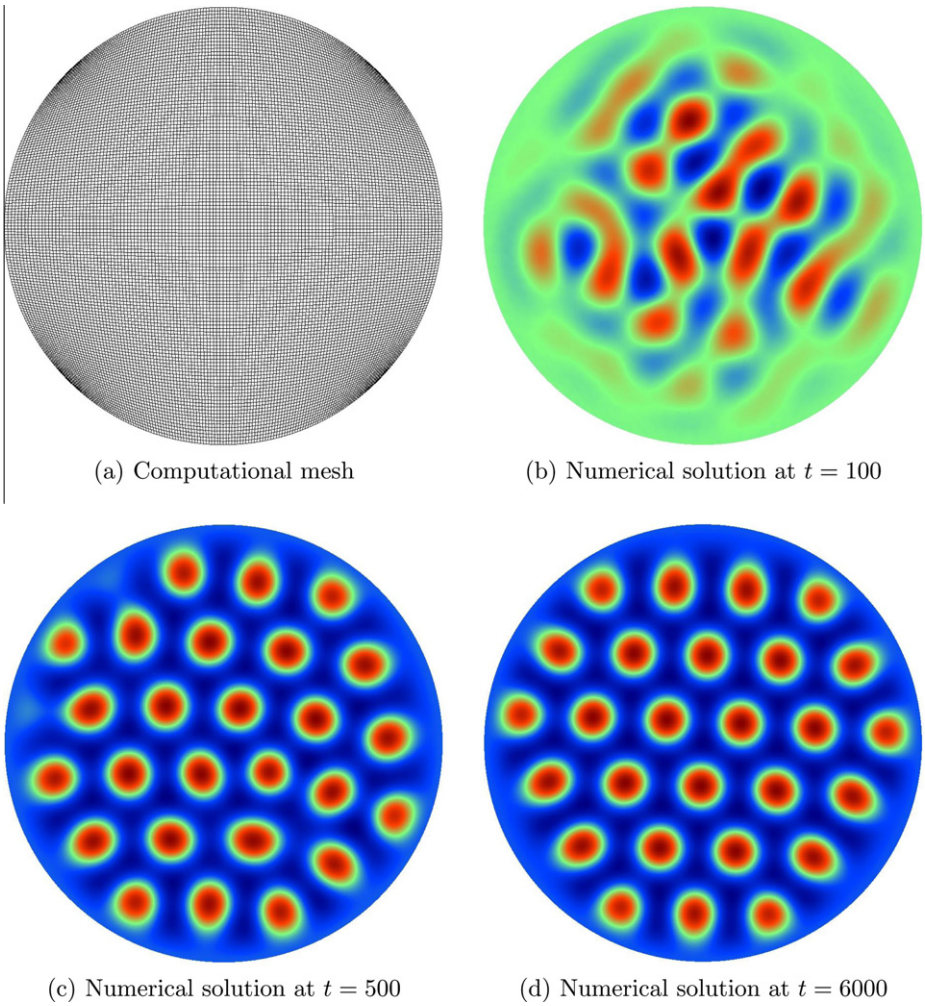

Fig. 15. Swift–Hohenberg equation on a disk. Computational mesh (a), and snapshots of the numerical approximation to the phase variable u at different computational times (b)–(d). The parameters of the equation are $D = k = 1,\ \epsilon = 0.1$, and $g = 1$. The spatial mesh is composed of 128^2 $\mathcal{C}^1$-quadratic elements. The time step is $\Delta t = 5$.

Fig. 14 shows the time evolution of the energy functional (1). We appended some snapshots of the solution to the free-energy curve. Note that the time scale is logarithmic, because the dynamics of the equation becomes slower as time evolves. We observe that the energy functional diminishes at all times, which confirms our theoretical predictions.

4.4. Swift–Hohenberg equation on a disk

We present the numerical solution to the Swift–Hohenberg equation on a disk. This geometry belongs to the class of conic sections that can be exactly reproduced by NURBS. To generate this geometry we employ quadratic basis functions and the parametrization defined in [60]. This leads to a mapping with four singular points on the boundary, as shown in the mesh picture presented in Fig. 15(a). These singularities did not produce any issues in the calculations. The radius disk is 15, and the computational mesh is composed of 128^2 $\mathcal{C}^1$ quadratic elements. On the boundary we set homogeneous Dirichlet boundary conditions. The time step is $\Delta t = 5$.

The parameters of the Swift–Hohenberg equation are $D = k = 1,\ \epsilon = 0.1$, and $g = 1$. The initial condition was generated following the same process as in the last example. Fig. 15(b)–(d) shows the time evolution of the phase variable u until the steady state is reached. We notice that the choice of $g = 1$ leads to the formation of circular structures, rather than stripes. Note that the circular structures are distorted near the boundary, due to the effect of Dirichlet boundary conditions.

In Fig. 16 we plot the time evolution of the energy functional. We notice that the energy decreases at all times, which supports our theoretical result about the unconditional stability of the presented space–time discretization.

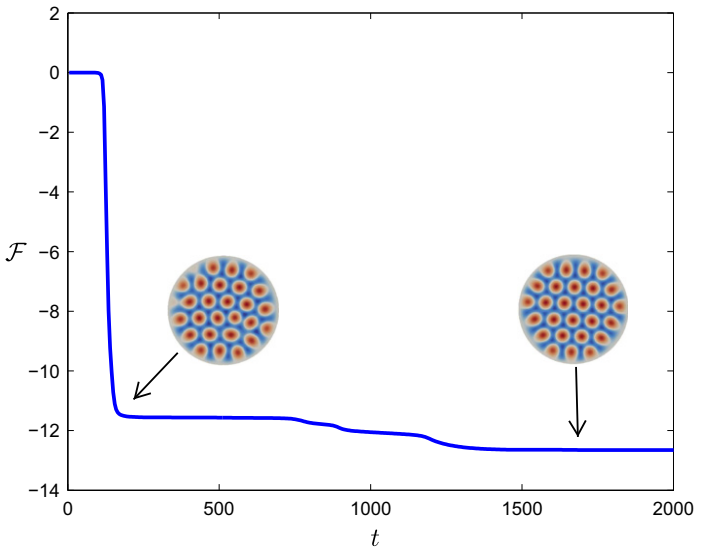

Fig. 16. Swift–Hohenberg equation on a disk. Time evolution of the energy functional. We appended some snapshots of the solution to the free-energy curve. These snapshots show that fast time-variations of the free-energy correspond to abrupt variations in the spatial pattern. We observe that the energy functional diminishes at all times, which confirms our theoretical predictions.

5. Conclusions

The Swift–Hohenberg equation is a higher-order nonlinear partial differential equation endowed with a nonlinear stability property. This equation governs the formation and evolution of roll patterns in Rayleigh–Bénard convection. We introduce a new space–time discretization that inherits the nonlinear stability relationship of the continuous equation irrespectively of the mesh and time step sizes, and that is second-order time-accurate. We present several numerical examples dealing with fluid convection on square and circular domains. These examples support our theoretical results and show the accuracy, efficiency and robustness of our new method. We also compare our method to other existing schemes, showing that is feasible alternative to the available methods.

Acknowledgements

The authors were partially supported by Xunta de Galicia (Grants # 09REM005118PR and #09MDS00718PR), Ministerio de Ciencia e Innovación (Grant # DPI2009-14546-C02-01) cofinanced with FEDER funds, and Universidad de A Coruña.

References

[1] Swift J, Hohenberg PC. Hydrodynamic fluctuations at the convective instability. Phys Rev A 1977;15:319–28.
[2] Evstigneev NM, Magnitskii NA, Sidorov SV. Nonlinear dynamics of laminar-turbulent transition in three dimensional Rayleigh–Benard convection. Commun Nonlinear Sci Numer Simulat 2010;15:2851–9.
[3] Wen B, Dianati N, Lunasin E, Chini GP, Doering CR. New upper bounds and reduced dynamical modeling for Rayleigh–Bénard convection in a fluid saturated porous layer. Commun Nonlinear Sci Numer Simulat 2012;17(5):2191–9.
[4] Rosa RR, Pontes J, Christov CI, Ramos FM, Rodrigues Neto C, Rempel EL, Walgraef D. Gradient pattern analysis of Swift–Hohenberg dynamics: phase disorder characterization. Physica A 2000;283:156–9.
[5] Kudryashov NA, Sinelshchikov DI. Exact solutions of the Swift–Hohenberg equation with dispersion. Commun Nonlinear Sci Numer Simulat 2012;17:26–34.
[6] Hutt A, Atay FM. Analysis of nonlocal neural fields for both general and gamma-distributed connectivities. Physica D 2005;203:30–54.
[7] Armero F, Zambrana-Rojas C. Volume-preserving energy-momentum schemes for isochoric multiplicative plasticity. Comput Methods Appl Mech Eng 2007;196:41304159.
[8] Meng XN, Laursen TA. Energy consistent algorithms for dynamic finite deformation plasticity. Comput Methods Appl Mech Eng 2002;191:1639–75.
[9] Romero I. Algorithms for coupled problems that preserve symmetries and the laws of thermodynamics: Part I: Monolithic integrators and their application to finite strain thermoelasticity. Comput Methods Appl Mech Eng 2010;199:1841–58.
[10] Romero I. Algorithms for coupled problems that preserve symmetries and the laws of thermodynamics: Part II: Fractional step methods. Comput Methods Appl Mech Eng 2010;199:2235–48.
[11] Harten A. On the symmetric form of systems of conservation laws with entropy. J Comput Phys 1983;49:151–64.
[12] Hughes TJR, Franca LP, Mallet M. A new finite element formulation for computational fluid dynamics: I. Symmetric forms of the compressible Euler and Navier–Stokes equations and the second law of thermodynamics. Comput Methods Appl Mech Eng 1986;54:223–34.
[13] Shakib F, Hughes TJR, Johan Z. A new finite element formulation for computational fluid-dynamics. 10. The compressible Euler and Navier–Stokes equations. Comput Methods Appl Mech Eng 1991;89:141–219.
[14] Tadmor E. Skew-selfadjoint form for systems of conservation laws. J Math Anal Appl 1984;103:428–42.
[15] Du Q, Nicolaides RA. Numerical analysis of a continuum model of phase transition. SIAM J Numer Anal 1991;28:1310–22.
[16] D.J. Eyre, An unconditionally stable one-step scheme for gradient systems, unpublished. <www.math.utah.edu/~eyre/research/methods/stable.ps>.

[17] Furihata D. A stable and conservative finite difference scheme for the Cahn–Hilliard equation. Numer Math 2001;87:675–99.
[18] Gomez H, Hughes TJR. Provably unconditionally stable, second-order time-accurate, mixed variational methods for phase-field models. J Comput Phys 2011;230:5310–27.
[19] He L, Liu Y. A class of stable spectral methods for the Cahn–Hilliard equation. J Comput Phys 2009;228:5101–10.
[20] Hu Z, Wise SM, Wang C, Lowengrub JS. Stable and efficient finite-difference nonlinear multigrid schemes for the phase field crystal equation. J Comput Phys 2009;228:5323–39.
[21] Wang C, Wang X, Wise SM. Unconditionally stable schemes for equations of thin film epitaxy. Discrete Contin Dynam Syst Series A 2010;28:405–23.
[22] Wang C, Wise SM. An energy stable and convergent finite difference scheme for the modified phase field crystal equation. SIAM J Numer Anal 2011;49:945–69.
[23] Wise SM. Unconditionally stable finite difference, nonlinear multigrid simulation of the Cahn–Hilliard–Hele–Shaw system of equations. J Sci Comput 2010;44:38–68.
[24] Wise SM, Wang C, Lowengrub JS. An energy-stable and convergent finite-difference scheme for the phase field crystal equation. SIAM J Numer Anal 2009;47(3):2269–88.
[25] Christov CI, Pontes J, Walgraef D, Velarde MG. Implicit time splitting for fourth-order parabolic equations. Comput Methods Appl Mech Eng 1997;148:209–24.
[26] Elder KR, Viñals J, Grant M. Ordering dynamics in the two-dimensional stochastic Swift–Hohenberg equation. Phys Rev Lett 1992;68(20):3024–7.
[27] Lloyd DJB, Sandstede B, Avitabile D, Champneys AR. Localized hexagon patterns of the planar Swift–Hohenberg equation. SIAM J Appl Dynam Syst 2008;7:1049–100.
[28] Viñals J, Hernandez-Garcia E, San Miguel M, Toral R. Numerical study of the dynamical aspects of pattern selection in the stochastic Swift–Hohenberg equation in one dimension. Phys Rev A 1991;44(2):1123–33.
[29] Xi H, Gunton JD, Viñals J. Spiral-pattern formation in Rayleigh–Bénard convection. Phys Rev E 1993;47(5):R2987–90.
[30] Xi H, Viñals J, Gunton JD. Numerical solution of the Swift–Hohenberg equation in two dimensions. Physica 1991;177:356–65.
[31] Staliunas K, Sánchez-Morcillo VJ. Dynamics of phase domains in the Swift–Hohenberg equation. Phys Lett A 1998;241:28–34.
[32] Christov CI, Pontes J. Numerical scheme for Swift–Hohenberg equation with strict implementation of Lyapunov functional. Math Comput Model 2002;35:87–99.
[33] Cottrell JA, Hughes TJR, Bazilevs Y. Isogeometric analysis: toward integration of CAD and FEA. Wiley; 2009.
[34] Hughes TJR, Cottrell JA, Bazilevs Y. Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement. Comput Methods Appl Mech Eng 2005;194:4135–95.
[35] Gomez H, Calo VM, Bazilevs Y, Hughes TJR. Isogeometric analysis of the Cahn–Hilliard phase-field model. Comput Methods Appl Mech Eng 2008;197:4333–52.
[36] Chandrasekhar S. Hydrodynamic and hydromagnetic stability. Dover Publications; 1974.
[37] Drazin PG, Reid WH. Hydrodynamic stability. Cambridge University Press; 2004.
[38] Getling AV. Rayleigh-Bénard convection, structures and dynamics. World Scientific Publishing; 1998.
[39] Koschmieder EL. Beńard cells and Taylor vortices. Cambridge University Press; 1993.
[40] Gurtin ME. Generalized Ginzburg–Landau and Cahn–Hilliard equations based on a microforce balance. Physica D 1996;92:178–92.
[41] Hughes TJR. The finite element method: linear static and dynamic finite element analysis. Mineola, NY: Dover Publications; 2000.
[42] Akkerman I, Bazilevs Y, Calo VM, Hughes TJR, Hulshoff S. The role of continuity in residual-based variational multiscale modeling of turbulence. Comput Mech 2007;41:371–8.
[43] Auricchic F, Beirao da Veiga L, Hughes TJR, Reali A, Sangalli G. Isogeometric collocation methods. Math Models Methods Appl Sci 2010;20(11):2075–107.
[44] Bazilevs Y, Calo VM, Cottrell JA, Evans JA, Hughes TJR, Lipton S, Scott MA, Sederberg TW. Isogeometric analysis using T-splines. Comput Methods Appl Mech Eng 2010;199:229–63.
[45] Bazilevs Y, Calo VM, Cottrell JA, Hughes TJR, Reali A, Scovazzi G. Variational multiscale residual-based turbulence modeling for large eddy simulation of incompressible flows. Comput Methods Appl Mech Eng 2007;197:173–201.
[46] Bazilevs Y, Hughes TJR. NURBS-based isogeometric analysis for the computation of flows about rotating components. Comput Mech 2008;43:143–50.
[47] Buffa A, Sangalli G, Vázquez R. Isogeometric analysis in electromagnetics: B-splines approximation. Comput Methods Appl Mech Eng 2010;199:1143–52.
[48] Cottrell JA, Hughes TJR, Reali A. Studies of refinement and continuity in isogeometric structural analysis. Comput Methods Appl Mech Eng 2007;196:4160–83.
[49] Elguedj T, Bazilevs Y, Calo VM, Hughes TJR. $\overline{B}$ and $\overline{F}$ projection methods for nearly incompressible linear and non-linear elasticity and plasticity using higher-order NURBS elements. Comput Methods Appl Mech Eng 2008;197:2732–62.
[50] Evans JA, Bazilevs Y, Babuška I, Hughes TJR. *n*-widths, sup infs, and optimality ratios for the *k*-version of the isogeometric finite element method. Comput Methods Appl Mech Eng 2009;198:1726–41.
[51] Hughes TJR, Reali A, Sangalli G. Duality and unified analysis of discrete approximations in structural dynamics and wave propagation: comparison of p-method finite elements with k-method NURBS. Comput Methods Appl Mech Eng 2008;197:4104–24.
[52] Lipton S, Evans JA, Bazilevs Y, Elguedj T, Hughes TJR. Robustness of isogeometric structural discretizations under severe mesh distortion. Comput Methods Appl Mech Eng 2010;199:357–73.
[53] Piegl L, Tiller W. The NURBS book. New York: Springer–Verlag; 1997.
[54] Rogers DF. An introduction to NURBS: with historical perspective. Morgan Kaufmann; 2001.
[55] Gomez H, Hughes TJR, Nogueira X, Calo VM. Isogeometric analysis of the isothermal Navier–Stokes–Korteweg equations. Comput Methods Appl Mech Eng 2010;199:1828–40.
[56] Gomez H, París J. Numerical simulation of asymptotic states of the damped Kuramoto–Sivashinsky equation. Phys Rev E 2011;83:046702.
[57] Cox SM, Matthews PC. Exponential time differencing for stiff systems. J Comput Phys 2002;176:430–55.
[58] Cheng M, Warren JA. An efficient algorithm for solving the phase field crystal model. J Comput Phys 2008;227:6241–8.
[59] Dahlquist G. A spectral stability problem for linear multistep methods. BIT 1963;3:27–43.
[60] Vuong A-V, Heinrich Ch, Simenon B. ISOGAT: a 2D tutorial MATLAB code for isogeometric analysis. Comput Aided Geometric Design 2010;27:644–55.