Appl. Numer. Math. 206 (2024) 340–357

Contents lists available at ScienceDirect

Applied Numerical Mathematics

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

Research Paper

Efficient simulation of complex Ginzburg–Landau equations using high-order exponential-type methods

Marco Caliari *, Fabio Cassini

Department of Computer Science, University of Verona, Strada Le Grazie, 15, 37134, Verona, Italy

ARTICLE INFO

Keywords:
Complex Ginzburg–Landau equation
Exponential integrators
Splitting methods
Lawson schemes
μ-mode product
Fast Fourier transform-based methods

ABSTRACT

In this paper, we consider the task of efficiently computing the numerical solution of evolutionary complex Ginzburg–Landau equations on Cartesian product domains with homogeneous Dirichlet/Neumann or periodic boundary conditions. To this aim, we employ for the time integration high-order exponential methods of splitting and Lawson type with constant time step size. These schemes enjoy favorable stability properties and, in particular, do not show restrictions on the time step size due to the underlying stiffness of the models. The needed actions of matrix exponentials are efficiently realized by using a tensor-oriented approach that suitably employs the so-called μ-mode product (when the semidiscretization in space is performed with finite differences) or with pointwise operations in Fourier space (when the model is considered with periodic boundary conditions). The overall effectiveness of the approach is demonstrated by running simulations on a variety of two- and three-dimensional (systems of) complex Ginzburg–Landau equations with cubic or cubic-quintic nonlinearities, which are widely considered in literature to model relevant physical phenomena. In fact, we show that high-order exponential-type schemes may outperform standard techniques to integrate in time the models under consideration, i.e., the well-known second-order split-step method and the explicit fourth-order Runge–Kutta integrator, for stringent accuracies.

1. Introduction

The evolutionary Complex Ginzburg–Landau (CGL) equation is an ubiquitous model for many physical scenarios. For instance, such a Partial Differential Equation (PDE) has applications in nonlinear fiber optics, fluids, and Bose–Einstein condensation, especially for the study of the dynamics of the so-called dissipative solitons [1] (see also the comprehensive reviews [2,3] and references therein). In this work, we consider the CGL equation in dimensionless units written in the form

$$\partial_t u = (\alpha_1 + \mathrm{i}\beta_1)\Delta u + \alpha_2 u + (\alpha_3 + \mathrm{i}\beta_3)|u|^2 u + (\alpha_4 + \mathrm{i}\beta_4)|u|^4 u, \tag{1}$$

where $u\colon [0,T]\times\Omega\to\mathbb{C}$ represents the complex-valued unknown function. The physical meaning of u depends on the specific example under consideration. The model is completed with an appropriate initial solution and boundary conditions of homogeneous Dirichlet/Neumann or periodic type. Results of existence and uniqueness of solution for equation (1) can be found, for instance, in References [4–6]. The spatial domain Ω, possibly after a truncation, is assumed to be the Cartesian product of one-dimensional finite

* Corresponding author.
E-mail addresses: marco.caliari@univr.it (M. Caliari), fabio.cassini@univr.it (F. Cassini).

https://doi.org/10.1016/j.apnum.2024.08.009
Received 5 March 2024; Received in revised form 18 June 2024; Accepted 7 August 2024

Available online 10 August 2024

intervals, that is $\Omega = (a_1, b_1) \times \cdots \times (a_d, b_d)$. The parameters α_j, β_j, with $j = 1, \ldots, 4$, are real numbers. In the following, we assume that α_1 is positive and that β_1, α_2, α_3, and β_3 are different from zero. If $\alpha_4 = \beta_4 = 0$, the equation is usually referred to as *cubic complex Ginzburg–Landau* equation, and it was first introduced in Reference [7] in the context of linearized perturbations. Whenever α_4 and/or β_4 are different from zero, the equation is known as *cubic-quintic complex Ginzburg–Landau*. This type of nonlinearity was first considered in Reference [8] for the study of certain soliton dynamics in chemical reactions. Also, notice that model (1) is in fact a generalization of the well-known nonlinear Schrödinger equation that we do not consider in this manuscript.

A suitable semidiscretization in space of equation (1) by the method of lines on a spatial grid of size $n_1 \times \cdots \times n_d$ gives a system of Ordinary Differential Equations (ODEs) in the form

$$\boldsymbol{u}'(t) = K\boldsymbol{u}(t) + \boldsymbol{g}(\boldsymbol{u}(t)). \tag{2a}$$

Here K is a complex matrix of size $N \times N$, with $N = n_1 \cdots n_d$, that discretizes the linear operator $(\alpha_1 + \mathrm{i}\beta_1)\Delta + \alpha_2$, while $\boldsymbol{g}$ represents the discretization of the nonlinear part of the PDE. We also assume that the matrix K can be written as a Kronecker sum, that is

$$K = A_d \oplus A_{d-1} \oplus \cdots \oplus A_1 = \sum_{\mu=1}^{d} A_{\otimes\mu}, \tag{2b}$$

where

$$A_{\otimes\mu} = I_d \otimes \cdots \otimes I_{\mu+1} \otimes A_\mu \otimes I_{\mu-1} \otimes \cdots \otimes I_1. \tag{2c}$$

The matrices I_μ and A_μ, for $\mu = 1, \ldots, d$, are of size $n_\mu \times n_\mu$, and represent the identity and the discretization of the one-dimensional linear differential operator along the μth direction, that is

$$A_\mu \approx (\alpha_1 + \mathrm{i}\beta_1)\partial_{x_\mu x_\mu} + \frac{\alpha_2}{d},$$

respectively. The matrices A_μ usually incorporate boundary conditions and they can be sparse or dense, depending on the specific discretization technique. Typical examples of discretizations that lead to such a Kronecker structure are finite differences and spectral methods. The former, in particular, is widely employed in the literature when dealing with non-periodic boundary conditions (see, for instance, References [1,9,10]). On the other hand, if equation (1) is coupled with periodic boundary conditions, the most used space discretization is the Fourier pseudospectral method, which gives rise to a system of ODEs in the form (2) for the Fourier coefficients of the expansion (we refer to References [11–13], among the others, for more details). In any case, remark that the resulting system of ODEs is *stiff*, due to the presence of the second-order spatial differential operator in equation (1).

Concerning the time discretization, fully explicit methods (usually of Runge–Kutta type) are widely employed in the literature, even if they must take into account a time step size restriction to guarantee stability (see, for instance, Reference [9]). However, probably the most popular techniques for the time marching are based on splitting schemes, giving rise to Time Splitting Fourier Pseudospectral (TSFP) methods or Time Splitting Finite Difference (TSFD) schemes (see Reference [10], and Reference [14] for an exhaustive treatment in the field of Bose–Einstein condensates). In this context, the flow of the original equation is split into a linear subflow and a nonlinear one. The linear flow can be approximated by suitable implicit methods when dealing with finite difference discretizations (see again References [10,14]) or solved exactly in Fourier space (by computing a diagonal matrix exponential). The nonlinear flow can typically be solved analytically, otherwise an explicit Runge–Kutta method may be employed. The two subflows are then suitably combined to produce an approximation of the exact solution of second order in time (this is the so-called Strang splitting, also known as split-step method). While such an approach is computationally attractive, it is of low accuracy when compared, for instance, to the standard explicit fourth-order Runge–Kutta method. An obvious remedy is then to look for higher-order splitting methods, or alternative schemes that perform equally well. In theory, it is indeed possible to construct splitting schemes of arbitrary order by suitably composing and combining the subflows, and we refer to Reference [15] for a comparison of high-order splitting methods with real coefficients for nonlinear Schrödinger equations, and to Reference [16] for the construction of high-order schemes with complex coefficients for parabolic problems. In our context, however, the presence of the complex term $\alpha_1 + \mathrm{i}\beta_1$ in front of the Laplacian operator brings additional troubles. In fact, in the general case, both high-order splitting schemes with real and complex coefficients would introduce time steps with negative real parts, and thus a potential unstable behavior since the CGL equation is irreversible. A possible solution is the use of additive-type splitting methods, which employ only positive time steps (see References [17,18]). In the very recent paper [19], these schemes have been tested (also) on the one-dimensional cubic-quintic CGL equation and proved to be superior to classical high-order schemes.

An alternative solution is to rely on different time approximation techniques, that still enjoy favorable properties in terms of stability. In this sense, exponential integrators of Lawson type [20], also known as Integrating Factor (IF) methods, provide a valuable alternative, since they can be designed of arbitrary high order without incurring into negative time steps. These kinds of integrators require the computation of the action of the matrix exponential and are employed in many instances in the literature, see for example References [21–23]. However, to the best of our knowledge, they are of little usage in the community of researchers interested in simulating CGL equations (see preprint [24] in which some Lawson methods are compared with other exponential integrators and splitting schemes on one- or two-dimensional examples with periodic boundary conditions).

In this paper, we aim at proposing efficient multidimensional implementations of exponential-type schemes (i.e., of splitting and Lawson type) for CGL equations, and to apply the methods to physically relevant examples such as the evolution of two- and three-dimensional autooscillating fields in the presence of walls and corners, evolution of necklace-ring patterns, and dynamics of

dissipative solitons. To accomplish the result, we will in particular employ the μ-mode approach (see References [25,26]) or the Fast Fourier Transform (FFT), depending on the specific example under consideration, for the efficient evaluation of the actions of matrix exponentials needed in the linear subflow or in the stages of the time marching methods. Other numerical integrators for the CGL equations which do not suffer from a time step size restriction exist in the literature. They involve implicit or exponential methods in time. Among them, we mention IMplicit EXplicit (IMEX) methods, which require the solution of linear systems (see the recent Reference [27]), and exponential Runge–Kutta integrators, which need the approximation of matrix functions related to the exponential (the so-called φ-functions, see References [24,28]). We do not consider them in this manuscript since, in general, it is not possible to directly apply the μ-mode approach to the solution of linear systems and to the approximation of matrix φ-functions arising from the discretization of multidimensional CGL equations.

The remaining part of the manuscript is organized as follows. In Section 2 we present the schemes that will be employed for the time evolution of the CGL equations. We proceed in Section 3 by describing in detail how to efficiently compute the action of the linear operator and of the matrix exponential in the case of a finite difference spatial discretization or a Fourier pseudospectral one. We then perform in Section 4 extensive numerical experiments on two- and three-dimensional (coupled) CGL equations with cubic and cubic-quintic nonlinearities, showing the effectiveness of the proposed exponential-type time marching methods. We finally draw the conclusions in Section 5.

2. Time discretization

As mentioned in the introduction, although explicit methods may suffer from a time step size restriction when employed in stiff ODEs systems of type (2), they are usually very simple to implement and can be quite efficient, especially if the number of discretization points N, and thus the norm of the matrix K, is not very large (see, for instance, References [9,29]). In our context, we consider popular Runge–Kutta methods of order two and four as terms of comparison to the exponential-type schemes which we will propose. The employed integrators are

$$\begin{aligned} \boldsymbol{f}_{n1} &= K\boldsymbol{u}_n + \boldsymbol{g}(\boldsymbol{u}_n), \\ \boldsymbol{f}_{n2} &= K\left(\boldsymbol{u}_n + \tau \boldsymbol{f}_{n1}\right) + \boldsymbol{g}\left(\boldsymbol{u}_n + \tau \boldsymbol{f}_{n1}\right), \\ \boldsymbol{u}_{n+1} &= \boldsymbol{u}_n + \frac{\tau}{2}(\boldsymbol{f}_{n1} + \boldsymbol{f}_{n2}), \end{aligned} \tag{3}$$

known as Heun's method or explicit trapezoidal method, and

$$\begin{aligned} \boldsymbol{f}_{n1} &= K\boldsymbol{u}_n + \boldsymbol{g}(\boldsymbol{u}_n), \\ \boldsymbol{f}_{n2} &= K\left(\boldsymbol{u}_n + \frac{\tau}{2}\boldsymbol{f}_{n1}\right) + \boldsymbol{g}\left(\boldsymbol{u}_n + \frac{\tau}{2}\boldsymbol{f}_{n1}\right), \\ \boldsymbol{f}_{n3} &= K\left(\boldsymbol{u}_n + \frac{\tau}{2}\boldsymbol{f}_{n2}\right) + \boldsymbol{g}\left(\boldsymbol{u}_n + \frac{\tau}{2}\boldsymbol{f}_{n2}\right), \\ \boldsymbol{f}_{n4} &= K(\boldsymbol{u}_n + \tau \boldsymbol{f}_{n3}) + \boldsymbol{g}(\boldsymbol{u}_n + \tau \boldsymbol{f}_{n3}), \\ \boldsymbol{u}_{n+1} &= \boldsymbol{u}_n + \frac{\tau}{6}\left(\boldsymbol{f}_{n1} + 2\boldsymbol{f}_{n2} + 2\boldsymbol{f}_{n3} + \boldsymbol{f}_{n4}\right). \end{aligned} \tag{4}$$

For simplicity of presentation, we consider a constant time step size $\tau = t_{n+1} - t_n$, while $\boldsymbol{u}_n$ and $\boldsymbol{u}_{n+1}$ denote the approximations of $\boldsymbol{u}(t_n)$ and $\boldsymbol{u}(t_{n+1})$, respectively.

Concerning the exponential-type methods, we will consider two families: the splitting and the Lawson ones.

2.1. Splitting schemes

The most popular time integrator for systems of ODEs (2) arising from the spatial discretization of CGL equations is probably the Strang splitting scheme (see, for instance, References [11–13,30]). If we denote by

$$\check{\boldsymbol{u}}(t) = \phi_t(\check{\boldsymbol{u}}_0)$$

the flow of $\check{\boldsymbol{u}}'(t) = \boldsymbol{g}(\check{\boldsymbol{u}}(t))$ starting from $\check{\boldsymbol{u}}_0$, then the Strang splitting scheme for system (2) is defined by

$$\boldsymbol{u}_{n+1} = S_\tau(\boldsymbol{u}_n) = \phi_{\tau/2}\left(\mathrm{e}^{\tau K}\phi_{\tau/2}(\boldsymbol{u}_n)\right). \tag{5}$$

It is a method of second order with respect to the time step size τ. The linear flow is the action of the matrix exponential $\mathrm{e}^{\tau K}$. The nonlinear flow $\phi_{\tau/2}$ can be computed exactly (e.g., for the cubic or the quintic CGL equations, see Section 4) or conveniently approximated by explicit methods (the Runge–Kutta schemes presented above, for instance), since the stiffness is already handled by the linear flow. A related method is presented in Reference [10], where the linear flow is treated in a different way. In fact, it is first observed that the flow corresponding to

$$\partial_t \hat{u} = (\alpha_1 + \mathrm{i}\beta_1)\Delta\hat{u}$$

can be *exactly* split into the partial flows of

$$\partial_t \hat{u}_\mu = (\alpha_1 + \mathrm{i}\beta_1)\partial_{x_\mu x_\mu}\hat{u}_\mu, \quad \mu = 1, \dots, d,$$

where $\partial_{x_\mu x_\mu}$ is discretized by finite differences. Then, each of the discrete partial linear flows (i.e., the action of the matrix exponential) is approximated by the Crank–Nicolson scheme. Similarly, in Reference [31] the Crank–Nicolson method is applied to a Chebyshev tau spectral discretization in space. Instead of the second-order approximation in time of the discretized flow given by the underlying trapezoidal rule, we will see in Section 3 how it is possible to evaluate *efficiently and exactly* the action of the discretized flow $\mathrm{e}^{\tau K}$ in equation (5).

Standard composition methods involving the flows $\phi_{a_j\tau}$ and $\mathrm{e}^{b_j\tau K}$, with $a_j, b_j \in \mathbb{C}$, can generate higher-order methods. However, as mentioned previously in the introduction, they would require some of the coefficients a_j and b_j to have real negative part or complex conjugate imaginary parts with large arguments, making them ill-suited for irreversible problems like the CGL equations (see, for instance, References [16,18,32]). Therefore, for our purposes we will consider the following fourth-order splitting scheme

$$\boldsymbol{u}_{n+1} = \frac{4}{3}S_{\tau/2}(S_{\tau/2}(\boldsymbol{u}_n)) - \frac{1}{3}S_\tau(\boldsymbol{u}_n), \tag{6}$$

which can be explicitly written as

$$\begin{aligned} \boldsymbol{u}_{n2} &= \phi_{\tau/2}\left(\mathrm{e}^{\tau K}\phi_{\tau/2}(\boldsymbol{u}_n)\right), \\ \boldsymbol{u}_{n+1} &= \frac{4}{3}\phi_{\tau/4}\left(\mathrm{e}^{\tau K/2}\phi_{\tau/2}\left(\mathrm{e}^{\tau K/2}\phi_{\tau/4}(\boldsymbol{u}_n)\right)\right) - \frac{1}{3}\boldsymbol{u}_{n2}. \end{aligned} \tag{7}$$

This scheme can be seen as the Richardson extrapolation of the Strang splitting method, and was developed and analyzed in Reference [17]. It is an additive scheme similar to the so-called affine splitting schemes introduced in Reference [18], but compared to those methods it requires less flow evaluations. Another splitting scheme of order four with positive steps in the linear flow $\mathrm{e}^{\tau K}$ is presented in the conclusions of Reference [32], but in fact it requires nine flow evaluations. For this reason, in this work we limit ourselves to scheme (6) among high-order splitting methods.

2.2. Lawson schemes

Lawson methods were presented in Reference [20] as generalized Runge–Kutta methods for systems with large Lipschitz constants, hence well-suited for many stiff problems. They can be derived by formally rewriting system (2) in terms of the new unknown $\boldsymbol{w}(t) = \mathrm{e}^{-tK}\boldsymbol{u}(t)$, where e^{-tK} acts as an integrating factor. Applying an explicit Runge–Kutta method to the new system $\boldsymbol{w}'(t) = \mathrm{e}^{-tK}\boldsymbol{g}(\mathrm{e}^{tK}\boldsymbol{w}(t))$, and rewriting the scheme in terms of the original unknown, the desired integrator is obtained. For instance, if we apply Heun's method (3) to the differential equation for $\boldsymbol{w}(t)$, we obtain the Lawson method of order two known in the literature as Lawson2b

$$\begin{aligned} \boldsymbol{u}_{n2} &= \mathrm{e}^{\tau K}(\boldsymbol{u}_n + \tau\boldsymbol{g}(\boldsymbol{u}_n)), \\ \boldsymbol{u}_{n+1} &= \mathrm{e}^{\tau K}\left(\boldsymbol{u}_n + \frac{\tau}{2}\boldsymbol{g}(\boldsymbol{u}_n)\right) + \frac{\tau}{2}\boldsymbol{g}(\boldsymbol{u}_{n2}). \end{aligned} \tag{8}$$

If we apply instead the Runge–Kutta method of order four (4) to the system of ODEs for $\boldsymbol{w}(t)$, we obtain the Lawson scheme of order four

$$\begin{aligned} \boldsymbol{u}_{n2} &= \mathrm{e}^{\tau K/2}\left(\boldsymbol{u}_n + \frac{\tau}{2}\boldsymbol{g}(\boldsymbol{u}_n)\right), \\ \boldsymbol{u}_{n3} &= \mathrm{e}^{\tau K/2}\boldsymbol{u}_n + \frac{\tau}{2}\boldsymbol{g}(\boldsymbol{u}_{n2}), \\ \boldsymbol{u}_{n4} &= \mathrm{e}^{\tau K}\boldsymbol{u}_n + \tau\mathrm{e}^{\tau K/2}\boldsymbol{g}(\boldsymbol{u}_{n3}), \\ \boldsymbol{u}_{n+1} &= \mathrm{e}^{\tau K}\left(\boldsymbol{u}_n + \frac{\tau}{6}\boldsymbol{g}(\boldsymbol{u}_n)\right) + \frac{\tau}{3}\mathrm{e}^{\tau K/2}(\boldsymbol{g}(\boldsymbol{u}_{n2}) + \boldsymbol{g}(\boldsymbol{u}_{n3})) + \frac{\tau}{6}\boldsymbol{g}(\boldsymbol{u}_{n4}), \end{aligned} \tag{9}$$

which is typically referred to as Lawson4. Similarly to the splitting schemes, they require only the evaluation of the action of the matrix exponential, and in particular not of the φ-functions needed by other classes of exponential methods [33]. Notice finally that, by construction, these methods do not have negative coefficients in front of the time step size in the matrix exponentials.

3. Action of the linear operator and of its matrix exponential

3.1. Finite difference spatial discretization

When system (2) is the result of a finite difference approximation in space with homogeneous Dirichlet or Neumann boundary conditions, the matrices A_μ are sparse with a certain number of non-zero diagonals adjacent to the main one (the specific number depends on the chosen approximation). The time marching schemes described in the previous section require, in particular, the action of the operator K and of its (scaled) matrix exponential, whose efficient computation is crucial for the effectiveness of the overall approach. To this aim, it is possible to suitably exploit the Kronecker sum structure of the linear operator to provide equivalent tensor formulations of the tasks. A reader not familiar with the following tensor formalism is invited to check Reference [26] for a detailed explanation. In fact, we can compute the action of the matrix $K \in \mathbb{C}^{N\times N}$ on a vector $\boldsymbol{u} \in \mathbb{C}^N$ by exploiting the equivalence

$$K\boldsymbol{u} = \mathrm{vec}\left(\sum_{\mu=1}^{d} \boldsymbol{U} \times_\mu A_\mu\right), \tag{10}$$

without assembling the matrix K itself. Here, $\boldsymbol{U} \in \mathbb{C}^{n_1 \times \cdots \times n_d}$ is the order-d tensor such that $\mathrm{vec}(\boldsymbol{U}) = \boldsymbol{u}$, where vec is the operator which stacks the columns of the input tensor into a suitable column vector. Moreover, $\times_\mu$ denotes the so-called μ-mode product. This operation multiplies the matrix A_μ onto the μ-fibers (i.e., the generalizations of rows and columns to tensors) of the tensor $\boldsymbol{U}$. The action of the matrix exponential $\mathrm{e}^{\tau K}$ can also be computed using μ-mode products, again without assembling the matrix K. Indeed, it holds

$$\mathrm{e}^{\tau K}\boldsymbol{u} = \mathrm{vec}\left(\boldsymbol{U} \times_1 \mathrm{e}^{\tau A_1} \times_2 \cdots \times_d \mathrm{e}^{\tau A_d}\right). \tag{11}$$

The concatenation of μ-mode products is usually referred to as a Tucker operator, and costs $\mathcal{O}((n_1 + \cdots + n_d)N)$ floating-point operations. Moreover, after computing the relevant matrix exponentials, it can be performed using the high performance level 3 BLAS, since the matrices $\mathrm{e}^{\tau A_\mu}$ and the tensor $\boldsymbol{U}$ are *dense*. This approach leads to a huge computational saving, see Reference [25]. Notice that formula (11) is peculiar to the exponential and does not hold, for instance, for the inverse function (needed by implicit methods) or the φ-functions (needed by other exponential integrators). Also, we remark that each matrix exponential of small size $n_\mu \times n_\mu$ can be computed by using standard Padé (see Reference [34]) or Taylor approximations (see References [35–37]), with a cost $\mathcal{O}(n_\mu^3)$. These computations can be performed once and for all before the time integration starts if the time step size is constant. Otherwise, the matrix exponentials must be recomputed at each time step. We finally highlight that the sparsity pattern of the matrices A_μ is *not* important. In fact, they are of small size, and the resulting matrix exponentials are full. Hence, there is no considerable advantage in performing computations with sparse matrices. The limit case would be, for instance, the usage of dense spectral differentiation matrices, see in particular References [38,39]. Also, the substitution of the Laplace operator in equation (1) with its fractional counterpart (i.e., considering a space-fractional CGL equation) would lead to full discretization matrices (see, for example, References [40,41]).

3.2. Pseudospectral Fourier spatial discretization

When dealing with periodic boundary conditions, it is common to employ a pseudospectral Fourier decomposition in space. We consider without loss of generality the spatial domain $\Omega = (0, 2\pi)^d$. The generalization to arbitrary domains $(a_1, b_1) \times \cdots \times (a_d, b_d)$ is straightforward. Then, we can approximate the solution of the CGL equation (1) by

$$u(t, \boldsymbol{x}) \approx \sum_{j_d=1}^{n_d} \cdots \sum_{j_1=1}^{n_1} u_{j_1 \dots j_d}(t)\phi_{j_1}(x_1) \cdots \phi_{j_d}(x_d),$$

where

$$\phi_{j_\mu}(x_\mu) = \mathrm{e}^{\mathrm{i}(j_\mu - 1)x_\mu}$$

and $\boldsymbol{U}(t) = (u_{j_1 \dots j_d}(t))$ are the discrete Fourier coefficients of $u(t, \boldsymbol{x})$. By performing standard calculations, we then end up with a system of ODEs for the spectral coefficients in the form (2), where $\boldsymbol{u}(t) = \mathrm{vec}(\boldsymbol{U}(t))$ and the matrices A_μ are diagonal and defined by

$$A_\mu = \mathrm{diag}\left(\frac{\alpha_2}{d}, (\alpha_1 + \mathrm{i}\beta_1)(-1) + \frac{\alpha_2}{d}, \dots, (\alpha_1 + \mathrm{i}\beta_1)(-(n_\mu - 1)^2) + \frac{\alpha_2}{d}\right).$$

The nonlinear function $\boldsymbol{g}(\boldsymbol{u}(t))$ is in this case

$$\mathcal{F}(g(\mathcal{F}^{-1}(\boldsymbol{u}(t)))),$$

where $\mathcal{F}$ denotes the d-dimensional Fourier transform (which maps a function sampled in Ω at a Cartesian grid of size $n_1 \times \cdots \times n_d$ to its discrete Fourier coefficients). It can be realized by the Fast Fourier Transform (FFT) algorithm at a cost $\mathcal{O}((\log n_1 + \cdots + \log n_d)N)$. In this setting, the action of K on a vector $\boldsymbol{u}$ can be easily computed as the pointwise multiplication between the tensors

$$\boldsymbol{K} = (k_{j_1 \dots j_d}) = \left((\alpha_1 + \mathrm{i}\beta_1)\left(-\sum_{\mu=1}^{d}(j_\mu - 1)^2\right) + \alpha_2\right)$$

and $\boldsymbol{U}$, where $\mathrm{vec}(\boldsymbol{U}) = \boldsymbol{u}$. Similarly, since $\mathrm{e}^{\tau A_\mu}$ are diagonal matrices, the action of the matrix exponential $\mathrm{e}^{\tau K}$ on the vector $\boldsymbol{u}$ can be computed as the pointwise multiplication between the tensors

$$\boldsymbol{E} = (e_{j_1 \dots j_d}) = \left(\mathrm{e}^{\tau((\alpha_1 + \mathrm{i}\beta_1)(-\sum_{\mu=1}^{d}(j_\mu - 1)^2) + \alpha_2)}\right)$$

and $\boldsymbol{U}$. Forming the tensor $\boldsymbol{E}$ costs $\mathcal{O}(N)$ and, similarly to the finite difference case, can be done once and for all before the time integration starts if the time step size τ is constant.

We finally remark here that other pseudospectral decompositions could profit from the underlying tensor structure of the problem. For instance, another classical pseudospectral decomposition employed for Schrödinger or CGL equations is based on Hermite functions (see References [19,25]), which naturally vanish for $\boldsymbol{x} \to \infty$. In this case, the resulting system of ODEs for the spectral

coefficients is not diagonal, but still possesses Kronecker sum structure (the discretization matrices A_μ are full with nonzero even diagonals). Hence, we could still efficiently compute the action of the linear operator K and its matrix exponential by means of the μ-mode approach described in Section 3.1.

4. Numerical experiments

In this section, we perform some numerical experiments with a variety of CGL equations defined on Cartesian product domains to assess the efficiency of high-order exponential-type methods over other popular techniques.

In all the examples, we first compare the performances of the methods in reaching a range of accuracies (measured in the infinity norm relatively to a reference solution). To this aim, we choose different sequences of time steps and measure the overall needed wall-clock time of the integration. It turns out that, for each numerical experiment, the same number of time steps can be employed for second-order and fourth-order exponential-type methods, respectively. In addition, the selected sequences allow to observe the expected orders of convergence. Concerning the Runge–Kutta methods, we choose the number of time steps also to clearly show the need of a time step size restriction.

Then, in a second experiment we show the dynamics of the equation up to a large final time. In this case, the time marching is performed with a single time step size employing the integrator that performed best in the previous experiment.

All the numerical examples are done on a standard laptop equipped with an Intel® Core™ i7-10750H chip (six physical cores) and 16 GB of RAM. As programming language we employ MATLAB (interpreted by the software MathWorks MATLAB® R2022a). The source code to reproduce all the experiments, fully compatible with GNU Octave, can be found in a GitHub repository.[1]

In all cases, we employ the tensor formulation of the problem using the built-in MATLAB N-D array type as storage for the tensors (i.e., no external class is needed). In particular, when performing experiments with finite differences, both the actions of the linear operator K and of the matrix exponentials are performed, in a μ-mode fashion, by using the right hand sides of formulas (10) and (11), without vectorizing the results. In practice, the tensor-matrix operations are done with level 3 BLAS using the functions `kronsumv` and `tucker` of KronPACK[2] (see Reference [26]). The needed small sized matrix exponentials are computed using a rational Padé approach with scaling and squaring [34] (implemented in the MATLAB function `expm`). When performing experiments with periodic boundary conditions, we exploit the high performance FFTW library (through the MATLAB function `fftn`) to realize a Fourier pseudospectral discretization, and all the relevant operations are performed pointwise (see Section 3.2). We work in a constant time step size scenario. Hence, the computation of the matrix exponentials is performed once and for all before the time integration starts, as mentioned in the previous section. This phase has negligible computational cost compared to that of the time integration, see also the discussions in References [25,42,43] for more details. The investigation of reliable and efficient variable time step size methods is beyond the scope of the present manuscript.

Finally, for the sake of compact notation, we set some labels for the time marching methods introduced in Section 2. More in detail, the Strang splitting scheme (5) is denoted as STRANG, the Lawson2b integrator (8) as IF2, the second-order Runge–Kutta method (3) as RK2, the fourth-order splitting method (7) as SPLIT4, the Lawson4 scheme (9) as IF4, and the standard explicit fourth-order Runge–Kutta integrator (4) as RK4.

4.1. 2D cubic CGL equation

In the first numerical example, we consider the cubic CGL equation

$$\partial_t u = (\alpha_1 + \mathrm{i}\beta_1)\Delta u + \alpha_2 u + (\alpha_3 + \mathrm{i}\beta_3)|u|^2 u, \tag{12}$$

in the two-dimensional domain $\Omega = (0, 100)^2$ with the parameters $\alpha_1 = 1$, $\beta_1 = 2$, $\alpha_2 = 1$, $\alpha_3 = -1$, and $\beta_3 = 0.2$. The number of discretization points is equal to $n = 256$ in both directions (total number of degrees of freedom $N = 256^2$). The values of the initial datum are random of *small amplitude*, that is taken from a standard normal distribution divided by 5000. In practice, this is realized in MATLAB by the command `randn(n,n)/5000` with the random number generator specified by `rng('default')`. For this example, we will consider two different types of boundary conditions (i.e., homogeneous Dirichlet and periodic), which lead to substantially different dynamics. The parameters, the initial datum, and the boundary conditions are taken from Reference [9].

Concerning the time integration methods of splitting type (i.e., STRANG and SPLIT4), we notice that the nonlinear part has exact solution. Indeed, the analytical solution of

$$\check{\boldsymbol{u}}'(t) = (\alpha_3 + \mathrm{i}\beta_3)|\check{\boldsymbol{u}}(t)|^2 \check{\boldsymbol{u}}(t)$$

starting from the initial datum $\check{\boldsymbol{u}}_0$ is

$$\check{\boldsymbol{u}}(t) = \phi_t^C(\check{\boldsymbol{u}}_0) = \mathrm{e}^{-\frac{\alpha_3+\mathrm{i}\beta_3}{2\alpha_3}\log|1-2\alpha_3|\check{\boldsymbol{u}}_0|^2 t|}\check{\boldsymbol{u}}_0. \tag{13}$$

[1] Available at https://github.com/caliarim/expCGL commit `c57bbbf`.

[2] Available at https://github.com/caliarim/KronPACK commit `562a9da`.

Table 1
Number of time steps, wall-clock time (in seconds), relative error at the final time $T = 6$, and observed numerical order of convergence for the solution of the 2D cubic CGL equation (12) with homogeneous Dirichlet boundary conditions and different integrators. The total number of degrees of freedom is $N = 256^2$. In the table, the cross symbol × means NaN/Inf value returned. See also Fig. 1 for a graphical representation.

IF2				IF4			
steps	time (s)	error	order	steps	time (s)	error	order
25	2.245e−01	4.359e−07	—	100	1.669e+00	5.858e−10	—
75	3.386e−01	4.984e−08	1.974	150	2.543e+00	2.378e−10	2.223
125	5.389e−01	1.842e−08	1.948	200	5.884e+00	1.033e−10	2.899
175	7.183e−01	9.538e−09	1.957	250	7.579e+00	4.800e−11	3.433
225	9.734e−01	5.782e−09	1.992	300	1.132e+01	2.375e−11	3.860

STRANG				SPLIT4			
steps	time (s)	error	order	steps	time (s)	error	order
25	1.161e−01	4.359e−07	—	100	9.419e−01	5.858e−10	—
75	2.362e−01	4.984e−08	1.974	150	1.399e+00	2.378e−10	2.223
125	3.015e−01	1.842e−08	1.948	200	3.036e+00	1.033e−10	2.899
175	3.575e−01	9.538e−09	1.957	250	3.752e+00	4.797e−11	3.437
225	5.012e−01	5.782e−09	1.992	300	5.429e+00	2.368e−11	3.872

RK2				RK4			
steps	time (s)	error	order	steps	time (s)	error	order
355	×	×	—	255	×	×	—
455	×	×	×	355	3.382e+00	3.241e−04	×
555	2.179e+00	1.088e−04	×	455	4.086e+00	1.467e−09	49.58
655	2.599e+00	7.821e−05	1.993	555	4.608e+00	6.594e−10	4.024
755	2.956e+00	5.891e−05	1.994	655	5.587e+00	3.396e−10	4.006

4.1.1. *Homogeneous Dirichlet boundary conditions*

We first consider the example with homogeneous Dirichlet boundary conditions. We then approximate the Laplace operator by fourth-order finite differences using internal points. The resulting unidirectional discretization matrix is

$$\frac{1}{12h^2}\begin{bmatrix} -15 & -4 & 14 & -6 & 1 & 0 & \dots & \dots & 0 \\ 16 & -30 & 16 & -1 & 0 & \dots & \dots & \dots & 0 \\ -1 & 16 & -30 & 16 & -1 & 0 & \dots & \dots & 0 \\ 0 & -1 & 16 & -30 & 16 & -1 & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & -1 & 16 & -30 & 16 & -1 & 0 \\ 0 & \dots & \dots & 0 & -1 & 16 & -30 & 16 & -1 \\ 0 & \dots & \dots & \dots & 0 & -1 & 16 & -30 & 16 \\ 0 & \dots & \dots & 0 & 1 & -6 & 14 & -4 & -15 \end{bmatrix}$$

for both directions, where $h = 100/(n+1)$ denotes the spatial step size. Notice that in the first and in the last rows (which correspond to the first and last internal point of the discretized domain, respectively) we employed non-centered finite differences to overcome the lack of enough surrounding points for the usual centered stencil.

For the first experiment we set the final simulation time to $T = 6$. The detailed outcome, together with the numbers of time steps and the obtained numerical order of convergence for the different integrators, is reported in Table 1. The results are also graphically depicted in a precision-work diagram in Fig. 1. First of all, for sufficiently small time step sizes all the integrators show the expected order of convergence. We notice that the second-order exponential-type methods seem to achieve the same accuracy. This is true also for the fourth-order exponential-type schemes. However, the methods do not exactly compute the same numerical solution. Indeed, if we perform a simulation with $T = 30$ and 50 time steps, the relative difference in the infinity norm between the second-order methods and the fourth-order ones is 5.834e−01 and 1.372e−01, respectively. The explicit Runge–Kutta methods RK2 and RK4 are the only schemes which suffer from a restriction on the time step size. Indeed, a stability constraint is required due to the underlying stiffness of the problem. In fact, RK2 is the least efficient scheme. The RK4 integrator needs more than five times the number of time steps employed by the fourth-order exponential-type methods to reach comparable accuracies, resulting then less efficient in the precision-work diagram. Notice also that, for each integrator, the average cost per time step is essentially constant. This is expected, since all the methods under consideration are direct (see also the discussion in Reference [42]). Splitting methods (both of second and fourth order) appear to be more efficient than the corresponding Lawson ones. Moreover, if we extend the lines in Fig. 1 to cover a wide range of accuracies, we observe that the fourth-order methods are slightly better than the second-order schemes. Overall, SPLIT4 allows to reach stringent accuracies with the smallest amount of computational time.

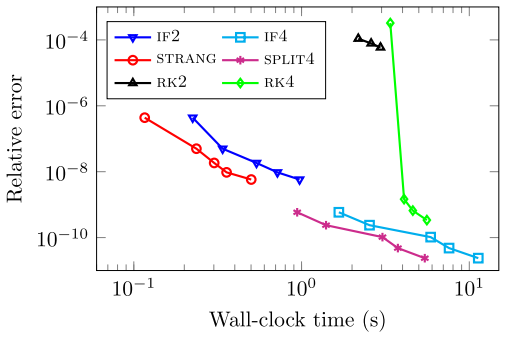

Fig. 1. Results for the simulation of the 2D cubic CGL equation (12) with homogeneous Dirichlet boundary conditions and $N = 256^2$ spatial discretization points. The number of time steps for each integrator is reported in Table 1. The final simulation time is $T = 6$. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

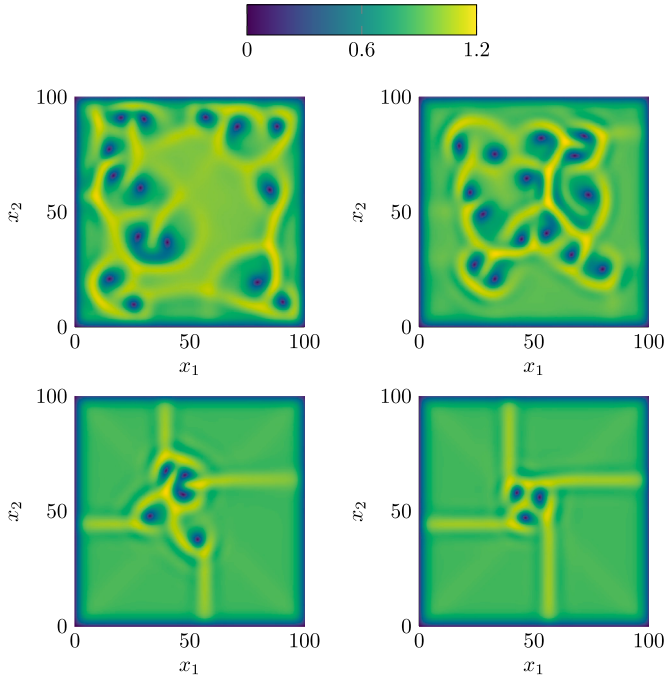

Fig. 2. Dynamics of $|u|$ for the 2D cubic CGL equation (12) with homogeneous Dirichlet boundary conditions and $N = 256^2$ spatial discretization points. The time integrator is SPLIT4. The final times are $T = 30$ (top left), $T = 60$ (top right), $T = 120$ (bottom left), and $T = 240$ (bottom right) with a number of time steps equal to $m = 50$, $m = 100$, $m = 200$, and $m = 400$, respectively. The wall-clock times of the simulations are 0.54, 1.04, 1.76, and 3.62 seconds, respectively.

Finally, in Fig. 2 we show the evolution of the dynamics of the cubic CGL equation under consideration. To this aim, we simulate with SPLIT4 up to the final times $T = 30$, $T = 60$, $T = 120$, and $T = 240$ (with $m = 50$, $m = 100$, $m = 200$, and $m = 400$ time steps, respectively). The latter simulation took less than 4 seconds. The results match satisfactorily with the theoretical predictions and with what already reported in the literature (see in particular Reference [9]). Indeed, starting from a random initial condition, defects are formed and they move around the domain while being pushed to the center from the waves emitted by the boundaries. Finally, after $T = 240$, the system reaches a so-called *frozen state*, and the modulus of the solution does not change anymore.

4.1.2. *Periodic boundary conditions*

We now consider the cubic CGL equation (12) completed with periodic boundary conditions. In this case, the space discretization is performed with a Fourier pseudospectral method.

Again, for the first experiment we set the final simulation time to $T = 6$. The outcome is reported in Table 2 and graphically presented in Fig. 3. Similar considerations as for the previous example can be drawn. The explicit Runge–Kutta integrators perform poorly. The second-order exponential-type methods reach comparable accuracies using the same number of time steps, and similarly for the fourth-order ones. Splitting methods are slightly more efficient in terms of computational time. Moreover, the advantage of using fourth-order exponential-type methods for stringent accuracies is even more evident and, for a wide range of tolerances, the best integrator is again SPLIT4.

Table 2
Number of time steps, wall-clock time (in seconds), relative error at the final time $T = 6$, and observed numerical order of convergence for the solution of the 2D cubic CGL equation (12) with periodic boundary conditions and different integrators. The total number of degrees of freedom is $N = 256^2$. In the table, the cross symbol × means NaN/Inf value returned. See also Fig. 3 for a graphical representation.

IF2				IF4			
steps	time (s)	error	order	steps	time (s)	error	order
25	6.806e−02	4.322e−07	—	200	1.038e+00	1.846e−10	—
75	1.742e−01	5.055e−08	1.953	250	1.244e+00	1.012e−10	2.694
125	2.811e−01	1.877e−08	1.940	300	1.481e+00	5.742e−11	3.110
175	3.881e−01	9.809e−09	1.928	350	1.807e+00	3.378e−11	3.441
225	5.305e−01	6.014e−09	1.946	400	1.949e+00	2.060e−11	3.706

STRANG				SPLIT4			
steps	time (s)	error	order	steps	time (s)	error	order
25	4.930e−02	4.322e−07	—	200	8.717e−01	1.846e−10	—
75	9.104e−02	5.055e−08	1.953	250	9.997e−01	1.012e−10	2.695
125	1.461e−01	1.877e−08	1.940	300	1.231e+00	5.742e−10	3.108
175	2.027e−01	9.809e−09	1.928	350	1.425e+00	3.378e−11	3.440
225	2.643e−01	6.014e−09	1.946	400	1.588e+00	2.060e−11	3.705

RK2				RK4			
steps	time (s)	error	order	steps	time (s)	error	order
755	×	×	—	555	×	×	—
855	×	×	×	655	2.745e+00	6.015e−08	×
955	1.976e+00	3.732e−05	×	755	3.104e+00	2.205e−10	39.47
1055	2.217e+00	3.060e−05	1.996	855	3.937e+00	1.327e−10	4.089
1155	2.434e+00	2.554e−05	1.996	955	4.329e+00	8.491e−11	4.034

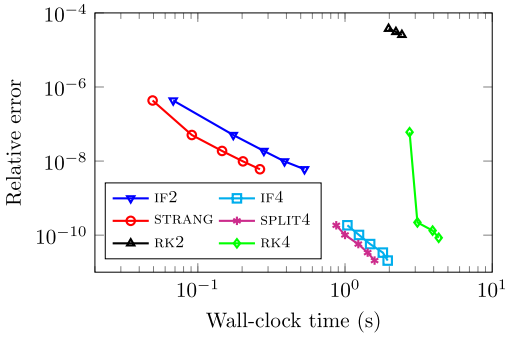

Fig. 3. Results for the simulation of the 2D cubic CGL equation (12) with periodic boundary conditions and $N = 256^2$ spatial discretization points. The number of time steps for each integrator is reported in Table 2. The final simulation time is $T = 6$.

For the second experiment, we show in Fig. 4 the dynamics of $|u|$ at the final times $T = 30$, $T = 60$, $T = 120$, and $T = 240$. The time integrator is SPLIT4 with a number of time steps equal to $m = 50$, $m = 100$, $m = 200$, and $m = 400$, respectively. Also in this case, the behavior matches with the theoretical expectations. In particular, the dynamics is deeply different from the case of homogeneous Dirichlet boundary conditions. Indeed, the initially formed defects move and merge but now cross the boundaries in a periodic fashion, and no frozen state is observed. In terms of wall-clock time, the simulation with the largest number of times steps executed in about two seconds.

4.2. 3D cubic CGL equation with homogeneous Dirichlet–Neumann boundary conditions

We now consider the cubic CGL equation

$$\partial_t u = (\alpha_1 + \mathrm{i}\beta_1)\Delta u + \alpha_2 u + (\alpha_3 + \mathrm{i}\beta_3)|u|^2 u, \tag{14}$$

in the three-dimensional domain $\Omega = (0, 100)^3$ with parameters $\alpha_1 = 1$, $\beta_1 = 2$, $\alpha_2 = 1$, $\alpha_3 = -1$, and $\beta_3 = 0.2$. The number of discretization points is equal to $n = 128$ in all the three directions (total number of degrees of freedom $N = 128^3$). As for the two-dimensional examples of Section 4.1, the initial datum is a standard normally distributed random obtained as `randn(n,n,n)/5000` with

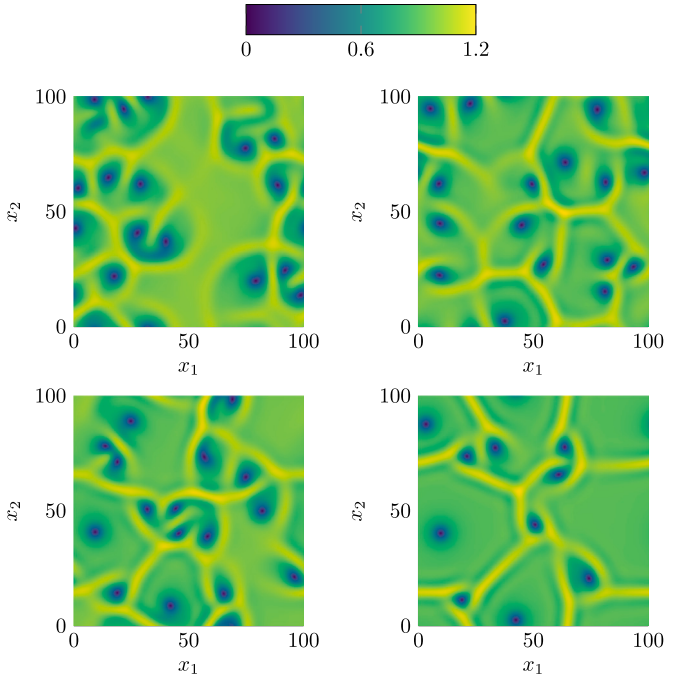

Fig. 4. Dynamics of $|u|$ for the 2D cubic CGL equation (12) with periodic boundary conditions and $N = 256^2$ spatial discretization points. The time integrator is SPLIT4. The final times are $T = 30$ (top left), $T = 60$ (top right), $T = 120$ (bottom left), and $T = 240$ (bottom right) with a number of time steps equal to $m = 50$, $m = 100$, $m = 200$, and $m = 400$, respectively. The wall-clock times of the simulations are 0.33, 0.59, 1.15, and 2.20 seconds, respectively.

`rng('default')`. Also, for the splitting methods the nonlinear flow can be again analytically computed as in formula (13). The equation is completed with homogeneous Dirichlet boundary conditions in the portion of boundary given by $\{(x_1, x_2, x_3) : x_1 x_2 x_3 = 0\}$, and homogeneous Neumann elsewhere (see Reference [9] for a similar example in two spatial dimensions). The discretization in space is then performed with fourth-order finite differences using semi-internal nodes in each direction. The unidirectional discretization matrix is in this case

$$\frac{1}{12h^2}\begin{bmatrix} -15 & -4 & 14 & -6 & 1 & 0 & \dots & \dots & 0 \\ 16 & -30 & 16 & -1 & 0 & \dots & \dots & \dots & 0 \\ -1 & 16 & -30 & 16 & -1 & 0 & \dots & \dots & 0 \\ 0 & -1 & 16 & -30 & 16 & -1 & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & -1 & 16 & -30 & 16 & -1 & 0 \\ 0 & \dots & \dots & 0 & -1 & 16 & -30 & 16 & -1 \\ 0 & \dots & 0 & 1 & -6 & 14 & -4 & -15 & 10 \\ 0 & \dots & \dots & 0 & 1 & -\frac{8}{3} & -6 & 56 & -\frac{145}{3} \end{bmatrix},$$

in all directions, where $h = 100/n$ is the spatial step size. The first row (which corresponds to the first internal point of the discretized domain) is the same as for the one in Section 4.1.1, while the last two (which corresponds to the last two points of the discretized domain) have been suitably modified to take into account a fourth-order discretization of the homogeneous Neumann boundary conditions and the lack of enough surrounding points for the usual centered stencil.

For the first experiment we set the final time to $T = 10$ and test the performances of the different integrators. The results are summarized in Table 3 and depicted in Fig. 5. The conclusions are equivalent to the ones drawn for the numerical examples in Section 4.1. In particular, the explicit Runge–Kutta methods suffer from a time step size restriction due to the stiffness, and a slight advantage for the splitting methods compared to the Lawson schemes is observed.

Concerning the second experiment, we investigate the dynamics of $|u|$ for the three-dimensional equation (14) using SPLIT4 as time integrator. To this aim we perform simulations up to the final times $T = 100$, $T = 200$, $T = 300$, and $T = 400$ with a number of time steps equal to $m = 1000$, $m = 2000$, $m = 3000$, and $m = 4000$, respectively. We show in Fig. 6 the corresponding isosurfaces at level 0.3. As we expect, in the three-dimensional scenario the defects have the shape of vortex filaments (see, for instance, References [29,44]), which move and evolve as the time proceeds. More specifically, they are pushed away from the faces of the cube in which homogeneous Dirichlet boundary conditions are prescribed, and they are ejected from the opposite faces (in which homogeneous Neumann boundary conditions are set). In fact, if we let evolve the system after time $T = 400$, no more defects are present in the considered domain. This is in line with what reported for the two-dimensional case in the literature (see, for instance, Reference [9]).

Table 3
Number of time steps, wall-clock time (in seconds), relative error at the final time $T = 10$, and observed numerical order of convergence for the solution of the 3D cubic CGL equation (14) with homogeneous Dirichlet–Neumann boundary conditions and different integrators. The total number of degrees of freedom is $N = 128^3$. In the table, the cross symbol × means NaN/Inf value returned. See also Fig. 5 for a graphical representation.

IF2				IF4			
steps	time (s)	error	order	steps	time (s)	error	order
25	3.964e+00	1.235e−03	—	50	2.247e+01	1.889e−07	—
75	1.175e+01	1.349e−04	2.015	100	4.506e+01	1.143e−08	4.047
125	1.956e+01	4.834e−05	2.009	150	6.879e+01	2.194e−09	4.071
175	2.812e+01	2.461e−05	2.007	200	9.175e+01	6.868e−10	4.036
225	3.683e+01	1.4869e−05	2.005	250	1.206e+02	2.803e−10	4.017

STRANG				SPLIT4			
steps	time (s)	error	order	steps	time (s)	error	order
25	2.342e+00	1.190e−03	—	50	1.852e+01	1.848e−07	—
75	6.855e+00	1.335e−04	1.991	100	3.237e+01	1.114e−08	4.052
125	1.110e+01	4.809e−05	1.999	150	4.714e+01	2.135e−09	4.075
175	1.567e+01	2.454e−05	1.999	200	6.261e+01	6.679e−10	4.039
225	1.993e+01	1.485e−05	2.000	250	7.815e+01	2.724e−10	4.019

RK2				RK4			
steps	time (s)	error	order	steps	time (s)	error	order
115	×	×	—	115	×	×	—
215	×	×	×	215	8.885e+01	8.108e−01	×
315	6.137e+01	1.292e−03	×	315	1.311e+02	5.817e−08	43.07
415	8.163e+01	7.484e−04	1.979	415	1.700e+02	1.942e−08	3.980
515	1.001e+02	4.876e−04	1.984	515	2.108e+02	8.214e−09	3.984

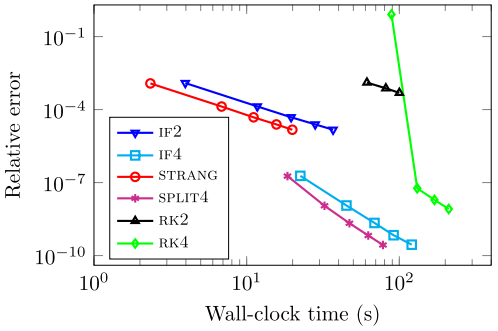

Fig. 5. Results for the simulation of the 3D cubic CGL equation (14) with homogeneous Dirichlet–Neumann boundary conditions and $N = 128^3$ spatial discretization points. The number of time steps for each integrator is reported in Table 3. The final simulation time is $T = 10$.

4.3. 3D cubic-quintic CGL equation with periodic boundary conditions

We now turn our attention to the following cubic-quintic CGL equation

$$\partial_t u = (\alpha_1 + \mathrm{i}\beta_1)\Delta u + \alpha_2 u + (\alpha_3 + \mathrm{i}\beta_3)|u|^2 u + (\alpha_4 + \mathrm{i}\beta_4)|u|^4 u, \tag{15}$$

in the cube $\Omega = (-12, 12)^3$ completed with periodic boundary conditions (see References [45,46]). The parameters are set to $\alpha_1 = 1/2$, $\beta_1 = 1/2$, $\alpha_2 = -1/2$, $\alpha_3 = 2.52$, $\beta_3 = 1$, $\alpha_4 = -1$, and $\beta_4 = -0.11$. Let us define $\rho = \sqrt{x_1^2 + x_2^2}$ and $\theta = \mathrm{atan2}(x_2, x_1)$. Then, we take as initial datum

$$u_0 = \delta \operatorname{sech}\left(\frac{\sqrt{(\rho - \rho_0)^2 + x_3^2}}{\omega}\right) \cos(\eta\theta) \exp(\mathrm{i}\kappa\theta),$$

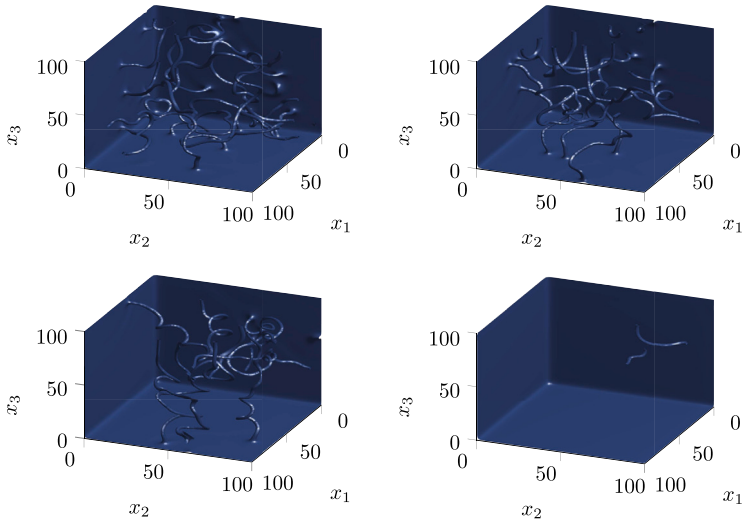

Fig. 6. Dynamics of $|u|$ for the 3D cubic CGL equation (14) with homogeneous Dirichlet–Neumann boundary conditions and $N = 128^3$ spatial discretization points (isosurface level 0.3). The time integrator is SPLIT4. The final times are $T = 100$ (top left), $T = 200$ (top right), $T = 300$ (bottom left), and $T = 400$ (bottom right) with a number of time steps equal to $m = 1000$, $m = 2000$, $m = 3000$, and $m = 4000$, respectively. The wall-clock times of the simulations are 306.82, 630.23, 917.55, and 1272.67 seconds, respectively.

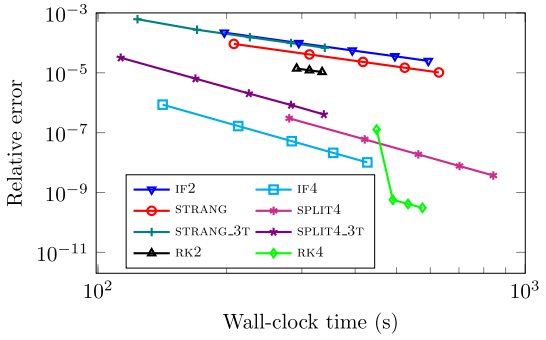

Fig. 7. Results for the simulation of the 3D cubic-quintic CGL equation (15) with periodic boundary conditions and $N = 128^3$ spatial discretization points. The number of time steps for each integrator is reported in Table 4. The final simulation time is $T = 5$.

with $\delta = 1.2$, $\rho_0 = 6$, $\omega = 2.5$, $\eta = 5$, and $\kappa = 3$. This initial condition corresponds to a *necklace-ring* with 10 *beads*, see the top left plot in Fig. 8 for a graphical representation. As space discretization we employ a Fourier pseudospectral approach with $n = 128$ modes along each direction, obtaining a total number of degrees of freedom equal to $N = 128^3$.

For the time integration in the first experiment, we consider all the marching schemes with the final time set to $T = 5$. Concerning specifically the splitting methods, notice that in contrast to the cubic nonlinearity case considered in Sections 4.1 and 4.2, here there is no exact solution for the nonlinear flow (see also the discussion in Reference [10]). In fact, in the time marching we numerically compute an approximation of it with a single step of RK4, since there is no source of stiffness and it overall gave the best results (the linear part is obviously integrated exactly). An alternative approach is to separate the nonlinearity into a cubic part and a quintic one, of which analytical solutions are known. Indeed, for the former we have the exact solution given by formula (13), while for the latter if we consider

$$\check{\boldsymbol{u}}'(t) = (\alpha_4 + \mathrm{i}\beta_4)|\check{\boldsymbol{u}}(t)|^4\check{\boldsymbol{u}}(t)$$

with initial datum $\check{\boldsymbol{u}}_0$ we have the exact solution

$$\check{\boldsymbol{u}}(t) = \phi_t^{\mathrm{q}}(\check{\boldsymbol{u}}_0) = \mathrm{e}^{-\frac{\alpha_4+\mathrm{i}\beta_4}{4\alpha_4}\log|1-4\alpha_4|\check{\boldsymbol{u}}_0|^4 t|}\check{\boldsymbol{u}}_0.$$

Embedding this procedure in a three-term splitting approach gives rise in principle to valuable alternatives for the time marching. In particular, for the second-order Strang splitting we consider

$$\boldsymbol{u}^{n+1} = S_\tau^3(\boldsymbol{u}_n) = \phi^{\mathrm{q}}_{\tau/2}(\phi^{\mathrm{c}}_{\tau/2}(\mathrm{e}^{\tau K}\phi^{\mathrm{c}}_{\tau/2}(\phi^{\mathrm{q}}_{\tau/2}(\boldsymbol{u}_n)))),$$

Table 4
Number of time steps, wall-clock time (in seconds), relative error at the final time $T = 5$, and observed numerical order of convergence for the solution of the 3D cubic-quintic CGL equation (15) with periodic boundary conditions and different integrators. The total number of degrees of freedom is $N = 128^3$. In the table, the cross symbol × means NaN/Inf value returned. See also Fig. 7 for a graphical representation.

IF2				IF4			
steps	time (s)	error	order	steps	time (s)	error	order
1000	1.976e+02	2.197e−04	—	300	1.417e+02	8.617e−07	—
1500	2.952e+02	9.859e−05	1.977	450	2.130e+02	1.659e−07	4.064
2000	3.940e+02	5.572e−05	1.984	600	2.842e+02	5.186e−08	4.042
2500	4.956e+02	3.576e−05	1.987	750	3.554e+02	2.109e−08	4.031
3000	5.935e+02	2.488e−05	1.990	900	4.272e+02	1.013e−08	4.025

STRANG				SPLIT4			
steps	time (s)	error	order	steps	time (s)	error	order
1000	2.080e+02	9.212e−05	—	300	2.804e+02	3.005e−07	—
1500	3.126e+02	4.094e−05	2.000	450	4.213e+02	5.942e−08	3.998
2000	4.173e+02	2.303e−05	2.000	600	5.623e+02	1.878e−08	4.004
2500	5.226e+02	1.474e−05	2.000	750	7.020e+02	7.672e−09	4.011
3000	6.284e+02	1.023e−05	2.000	900	8.426e+02	3.685e−09	4.023

STRANG_3T				SPLIT4_3T			
steps	time(s)	error	order	steps	time(s)	error	order
1000	1.237e+02	6.161e−04	—	300	1.130e+02	3.189e−05	—
1500	1.706e+02	2.738e−04	2.000	450	1.691e+02	6.378e−06	3.969
2000	2.268e+02	1.540e−04	2.000	600	2.255e+02	2.034e−06	3.973
2500	2.830e+02	9.858e−05	2.000	750	2.830e+02	8.375e−07	3.977
3000	3.396e+02	6.846e−05	2.000	900	3.379e+02	4.054e−07	3.980

RK2				RK4			
steps	time (s)	error	order	steps	time (s)	error	order
1200	×	×	—	1000	×	×	—
1300	×	×	×	1100	4.492e+02	1.273e−07	×
1400	2.918e+02	1.386e−05	×	1200	4.901e+02	5.698e−10	62.17
1500	3.127e+02	1.208e−05	2.002	1300	5.320e+02	4.133e−10	4.012
1600	3.347e+02	1.061e−05	2.002	1400	5.753e+02	3.077e−10	3.982

where ϕ^{c} and ϕ^{q} denote the exact flows for the cubic and the quintic parts, respectively. The scheme is labeled as STRANG_3T. Then, similarly to formula (6), if we perform the Richardson extrapolation we obtain the fourth-order method

$$\boldsymbol{u}_{n+1} = \frac{4}{3} S^3_{\tau/2}(S^3_{\tau/2}(\boldsymbol{u}_n)) - \frac{1}{3} S^3_{\tau}(\boldsymbol{u}_n),$$

that we label SPLIT4_3T. The outcome of the experiments is summarized in Table 4 and plotted in Fig. 7. As for all the previous examples, we notice that all the time integrators show the expected order of convergence. Also, the explicit Runge–Kutta methods again show a severe time step size restriction, which makes them not of practical usage for fast simulations with a low number of time steps and large number of degrees of freedom. Concerning the second-order exponential-type methods, we observe from the table that the three-term splitting STRANG_3T is cheaper than the STRANG method with nonlinear flux computed numerically, as expected. On the other hand, the relative error obtained is almost seven times larger. This makes the STRANG scheme the preferred one in the precision-work diagram, since the IF2 method performs similarly to STRANG_3T. For the exponential-type methods of fourth order, we observe that the three-term splitting approach SPLIT4_3T does not pay off compared to SPLIT4, since the lower computational cost is surpassed by a considerable higher error obtained. As a matter of fact, the IF4 is more efficient than all the other exponential-type methods. We then conclude that for the example under consideration the integrator which performs best is IF4.

In the second experiment we present the dynamics of the model for different final times. To this aim, we select as time integrator IF4, and we consider the final simulation times $T = 20$, $T = 40$, and $T = 80$ with a number of time steps equal to $m = 80$, $m = 160$, and $m = 320$, respectively. The results, depicted in Fig. 8, are coherent with what already presented in the literature, see References [45, 46]. In particular, the magnitude of the coefficients in the equation transforms the initial necklace-ring to a torus, which shrinks as time evolves. After time $T = 120$ the isosurface does not show qualitatively modifications, suggesting that the stationary state has been reached. As already observed previously in the numerical experiments, we notice that also here the simulation wall-clock time of a single step is the same for each simulation, since the underlying marching method is a direct algorithm. The simulation with the largest number of time steps takes about 160 seconds.

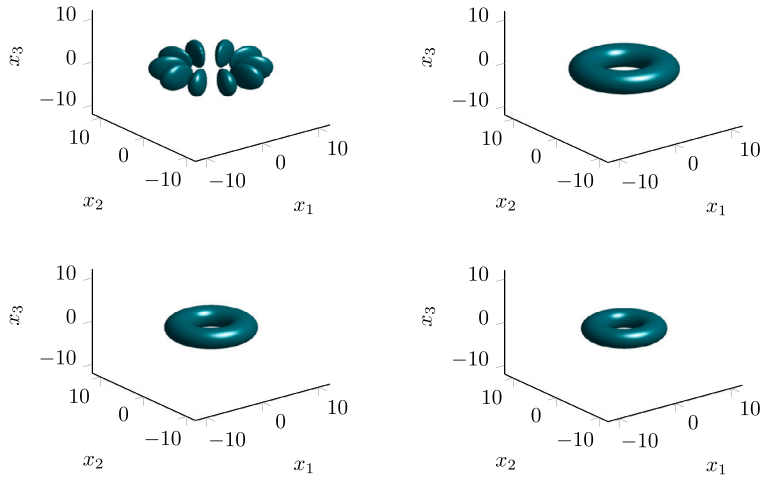

Fig. 8. Dynamics of $|u|^2$ for the 3D cubic-quintic CGL equation (15) with periodic boundary conditions and $N = 128^3$ spatial discretization points (isosurface level 0.5). The time integrator is IF4. We depict the initial condition (top left plot), and the isosurfaces at the final times $T = 20$ (top right plot), $T = 40$ (bottom left plot), and $T = 80$ (bottom right plot). The number of time steps is equal to $m = 80$, $m = 160$, $m = 320$, respectively. The wall-clock times of the simulations are 38.93, 81.38, and 159.65 seconds, respectively.

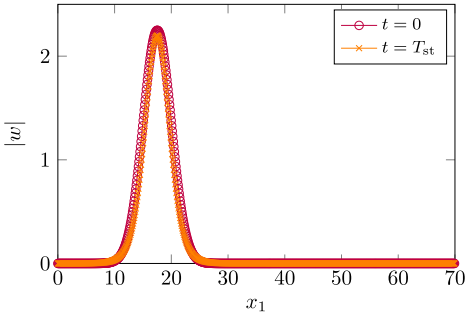

Fig. 9. Plot of $|w|$ obtained by simulating the one-dimensional cubic-quintic CGL equation (17) with periodic boundary conditions up to the final time $T_{st} = 25$. The time integrator is IF4 with $m = 10000$ time steps. The number of spatial discretization points is $n_1 = 700$. The wall-clock time of this simulation is 2.62 seconds. The obtained state is then employed to build the initial condition for the 2D coupled model (16).

4.4. Coupled 2D cubic-quintic CGL equation with periodic boundary conditions

We consider as last example the interaction of two quasi-1D dissipative solitons modeled by a system of two coupled cubic-quintic CGL equations (see Reference [47])

$$\begin{cases} \partial_t u = \alpha_0 \partial_{x_1} u + (\alpha_1 + \mathrm{i}\beta_1)\Delta u + \alpha_2 u + (\alpha_3 + \mathrm{i}\beta_3)|u|^2 u + (\alpha_4 + \mathrm{i}\beta_4)|u|^4 u + \alpha_5 |v|^2 u, \\ \partial_t v = -\alpha_0 \partial_{x_1} v + (\alpha_1 + \mathrm{i}\beta_1)\Delta v + \alpha_2 v + (\alpha_3 + \mathrm{i}\beta_3)|v|^2 v + (\alpha_4 + \mathrm{i}\beta_4)|v|^4 v + \alpha_5 |u|^2 v. \end{cases} \tag{16}$$

The spatial domain is set to $\Omega = (0, 70) \times (0, 35)$, and the boundary conditions are periodic. The parameter α_0 represents the approach velocity of the dissipative solitons (and is set to -0.4), while α_5 is the cubic cross-coupling parameter (taken equal to 0.5). The remaining parameters are set to $\alpha_1 = 0.125$, $\beta_1 = 0.5$, $\alpha_2 = -0.9$, $\alpha_3 = 1$, $\beta_3 = 0.8$, $\alpha_4 = -0.1$, and $\beta_4 = -0.6$. The semidiscretization in space is performed with a Fourier pseudospectral approach with $n_1 = 700$ modes along the first direction and $n_2 = 350$ along the second one (the total number of degrees of freedom is then $N = 700 \cdot 350$).

Following the description in Reference [47] (see also the discussion in References [3,12]), the initial conditions u_0 and v_0 are obtained starting from stationary solitons as follows. First, we solved the one-dimensional cubic-quintic CGL equation

$$\begin{cases} \partial_t w = (\alpha_1 + \mathrm{i}\beta_1)\partial_{x_1 x_1} w + \alpha_2 w + (\alpha_3 + \mathrm{i}\beta_3)|w|^2 w + (\alpha_4 + \mathrm{i}\beta_4)|w|^4 w, \\ w_0(x_1) = \delta \exp\left(-\dfrac{(x_1 - \chi)^2}{2\sigma^2}\right), \end{cases} \tag{17}$$

Table 5
Number of time steps, wall-clock time (in seconds), relative error at the final time $T = 3$, and observed numerical order of convergence for the solution of the 2D coupled cubic-quintic CGL equations (16) with periodic boundary conditions and different integrators. The total number of degrees of freedom is $N = 700 \cdot 350$. In the table, the cross symbol × means NaN/Inf value returned. See also Fig. 10 for a graphical representation.

IF2				IF4			
steps	time (s)	error	order	steps	time (s)	error	order
5000	1.428e+02	1.255e−04	—	500	3.344e+01	2.079e−05	—
7000	1.982e+02	6.431e−05	1.987	1000	6.712e+01	1.305e−06	3.994
9000	2.514e+02	3.900e−05	1.990	1500	1.003e+02	2.576e−07	4.001
11000	3.159e+02	2.614e−05	1.992	2000	1.340e+02	8.130e−08	4.009
13000	3.759e+02	1.874e−05	1.994	2500	1.677e+02	3.311e−08	4.026

STRANG				SPLIT4			
steps	time (s)	error	order	steps	time (s)	error	order
5000	1.800e+02	1.008e−04	—	500	8.550e+01	4.100e−06	—
7000	2.536e+02	5.142e−05	1.999	1000	1.712e+02	2.560e−07	4.001
9000	3.243e+02	3.111e−05	1.999	1500	2.498e+02	5.029e−08	4.013
11000	3.967e+02	2.083e−05	1.999	2000	3.330e+02	1.569e−08	4.049
13000	4.678e+02	1.491e−05	2.000	2500	4.158e+02	6.237e−09	4.134

RK2				RK4			
steps				steps	time (s)	error	order
1062	×	×	—	562	×	×	—
1562	×	×	×	1062	6.674e+01	2.574e−03	×
2062	5.780e+01	2.106e−03	×	1562	9.886e+01	5.288e−07	22.01
2562	7.232e+01	1.341e−03	2.080	2062	1.303e+02	1.752e−07	3.978
3062	8.601e+01	9.295e−04	2.055	2562	1.577e+02	7.393e−08	3.974

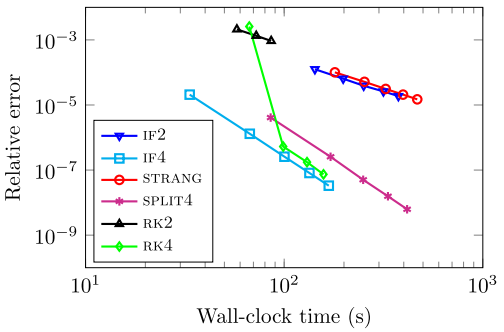

Fig. 10. Results for the simulation of the 2D coupled cubic-quintic CGL equations (16) with periodic boundary conditions and $N = 700 \cdot 350$ spatial discretization points. The number of time steps for each integrator is reported in Table 5. The final simulation time is $T = 3$.

completed with periodic boundary conditions, in the domain $(0, 70)$ up to $T_{\text{st}} = 25$. The discretization in space is again performed with a Fourier pseudospectral method using $n_1 = 700$ modes. As time integrator we employ the IF4 method with $m = 10000$ time steps. The parameters of the initial condition w_0 are set to $\delta = 2.25$, $\chi = 17.5$, and $\sigma = 2.5$. We selected $T_{\text{st}} = 25$ as the final simulation time since, after that time, we did not observe qualitatively variations of $|w(t, x_1)|$ anymore (see Fig. 9 for the plot of the initial and final states). Notice that performing such a simulation is inexpensive in terms of wall clock time (in fact, it took less than three seconds to obtain the results). Then, we set the initial states of the coupled equations (16) to $u_0(x_1, x_2) = w(T_{\text{st}}, x_1)$ and $v_0(x_1, x_2) = w(T_{\text{st}}, 70 - x_1)$. The discretization in space of the coupled PDEs (16) yields a system of ODEs (2) with

$$K = \begin{bmatrix} K^{\boldsymbol{u}} & 0 \\ 0 & K^{\boldsymbol{v}} \end{bmatrix},$$

where $K^{\boldsymbol{u}}$ and $K^{\boldsymbol{v}}$ are matrices in Kronecker form which discretize the linear operators of the equation for the unknown u and the unknown v, respectively. Due to the block diagonal structure of K, the computation of the action of the operator and its matrix exponential can be independently performed on the components $\boldsymbol{u}$ and $\boldsymbol{v}$. This is true also for similar systems with more than two coupled equations (see, for instance, Reference [48]).

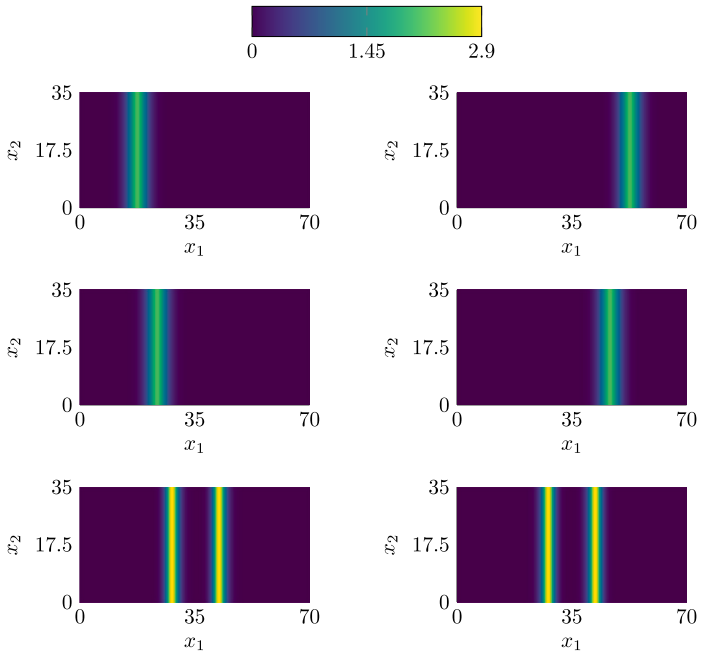

Fig. 11. Dynamics of $|u|$ (left) and $|v|$ (right) for the 2D coupled cubic-quintic CGL equations (16) with periodic boundary conditions and $N = 700 \cdot 350$ spatial discretization points. The time integrator is IF4. The initial condition is depicted (top), while the simulations are performed with the final times $T = 15$ (center) and $T = 30$ (bottom). The number of time steps is equal to $m = 1000$ and $m = 2000$, respectively. The wall-clock times of the simulations are 60.89 and 125.49 seconds, respectively.

As did for the numerical examples in the previous sections, we first integrate the system of cubic-quintic CGL equations (16) using different time marching schemes and varying number of time steps. The nonlinear fluxes in the splitting methods are approximated by a single step of the RK4 scheme. For the simulations, the final time is set to $T = 3$. The results, together with the specific number of time steps employed for each method, are reported in Table 5 and graphically depicted in Fig. 10. As we can observe, the most efficient method is also in this case the fourth-order Lawson scheme IF4. In fact, it does not show a stability restriction due to the stiffness (as the explicit RK4 and RK2 methods do) and overall surpasses all the methods under consideration (in particular SPLIT4, which requires more than twice the wall-clock time needed by IF4 to reach about the same level of accuracy).

Finally, we perform a last experiment on the coupled model by employing IF4 to show the dynamics of $|u|$ and $|v|$ (see Fig. 11). To this aim, we consider the final times $T = 15$ and $T = 30$ with a number of time steps equal to $m = 1000$ and $m = 2000$, respectively. As expected, the advection terms in the equations move the solitons in opposite directions. When they approach, they collide and start interacting. As thoroughly explained in Reference [47], the choice of the parameters in the equations should bring to a stationary state constituted by two solitons in the same position for both the u and the v components. This is indeed verified by the numerical simulation performed, which after $T = 30$ shows qualitatively the same figure.

5. Conclusions

We have highlighted, by conducting extensive numerical experiments in a variety of situations, that complex Ginzburg–Landau equations defined on Cartesian product domains can be efficiently integrated in time by using high-order exponential-type integrators with constant time step size. We have shown that the proposed methods are more efficient (in terms of computational time) than the explicit fourth-order Runge–Kutta scheme and the Strang splitting procedure. Computational efficiency is achieved by suitably exploiting the underlying Kronecker sum structure of the problem or a Fourier pseudospectral decomposition in space (when the boundary conditions admit), both of which allow for an effective computation of the involved matrix functions. We have overall observed that, among the exponential-type schemes, the more the nonlinearity becomes involved (i.e., in the cubic-quintic and in the coupled cases), the better Lawson methods perform compared to splitting-based integrators. As possible interesting future investigations, we mention the realization of variable step size integrators, the simulation of further models (the cubic-quintic-septic model proposed in Reference [13], for instance), and the employment of the proposed techniques in the context of fractional models (see References [19,40]).

CRediT authorship contribution statement

Marco Caliari: Conceptualization, Formal analysis, Investigation, Methodology, Project administration, Software, Supervision, Validation, Writing – original draft, Writing – review & editing. **Fabio Cassini:** Conceptualization, Formal analysis, Funding acquisition, Investigation, Methodology, Software, Validation, Visualization, Writing – original draft, Writing – review & editing.

Declaration of competing interest

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

Acknowledgements

The authors are members of the Gruppo Nazionale Calcolo Scientifico-Istituto Nazionale di Alta Matematica (GNCS-INdAM). Fabio Cassini received financial support from the Italian Ministry of University and Research (MUR) with the PRIN Project 2022 No. 2022N9BM3N “Efficient numerical schemes and optimal control methods for time-dependent partial differential equations” and holds a fellowship funded by INdAM.

References

[1] D. Mihalache, D. Mazilu, F. Lederer, H. Leblond, B.A. Malomed, Collisions between coaxial vortex solitons in the three-dimensional cubic-quintic complex Ginzburg-Landau equation, Phys. Rev. A 77 (2008) 033817, https://doi.org/10.1103/PhysRevA.77.033817.
[2] I.S. Aranson, L. Kramer, The world of the complex Ginzburg-Landau equation, Rev. Mod. Phys. 74 (2002) 99–143, https://doi.org/10.1103/RevModPhys.74.99.
[3] B.A. Malomed, Multidimensional dissipative solitons and solitary vortices, Chaos Solitons Fractals 163 (2022) 112526, https://doi.org/10.1016/j.chaos.2022.112526.
[4] Q. Du, Global existence and uniqueness of solutions of the time-dependent Ginzburg-Landau model for superconductivity, Appl. Anal. 53 (1994) 1–17, https://doi.org/10.1080/00036819408840240.
[5] S. Descombes, M. Moussaoui, Global existence and regularity of solutions for complex Ginzburg-Landau equations, Boll. UMI 3-B (2000) 193–211.
[6] D. Shimotsuma, T. Yokota, K. Yoshii, Existence and decay estimates of solutions to complex Ginzburg–Landau type equations, J. Differ. Equ. 260 (2016) 3119–3149, https://doi.org/10.1016/j.jde.2015.10.030.
[7] K. Stewartson, J.T. Stuart, A non-linear instability theory for a wave system in plane Poiseuille flow, J. Fluid Mech. 48 (3) (1971) 529–545, https://doi.org/10.1017/S0022112071001733.
[8] V.I. Petviashvili, A.M. Sergeev, Spiral solitons in active media with an excitation threshold, Dokl. Akad. Nauk SSSR 276 (1984) 1380–1384, in Russian.
[9] V.M. Eguíluz, E. Hernández-García, O. Piro, Complex Ginzburg-Landau equation in the presence of walls and corners, Phys. Rev. E 64 (2001) 036205, https://doi.org/10.1103/PhysRevE.64.036205.
[10] S. Wang, L. Zhang, An efficient split-step compact finite difference method for cubic–quintic complex Ginzburg–Landau equations, Comput. Phys. Commun. 184 (2013) 1511–1521, https://doi.org/10.1016/j.cpc.2013.01.019.
[11] Y.J. He, H.H. Fan, J.W. Dong, H.Z. Wang, Self-trapped spatiotemporal necklace-ring solitons in the Ginzburg-Landau equation, Phys. Rev. E 74 (2006) 016611, https://doi.org/10.1103/PhysRevE.74.016611.
[12] G.-A. Zakeri, E. Yomba, Modulational instability regions for coupled Ginzburg-Landau equations with higher order of nonlinearities, Phys. Rev. E 91 (2015) 062904, https://doi.org/10.1103/PhysRevE.91.062904.
[13] M. Djoko, T.C. Kofane, The cubic–quintic–septic complex Ginzburg–Landau equation formulation of optical pulse propagation in 3D doped Kerr media with higher-order dispersions, Opt. Commun. 416 (2018) 190–201, https://doi.org/10.1016/j.optcom.2018.02.027.
[14] W. Bao, Y. Cai, Mathematical theory and numerical methods for Bose-Einstein condensation, Kinet. Relat. Models 6 (1) (2013) 1–135, https://doi.org/10.3934/krm.2013.6.1.
[15] M. Thalhammer, M. Caliari, C. Neuhauser, High-order time-splitting Hermite and Fourier spectral methods, J. Comput. Phys. 228 (3) (2009) 822–832, https://doi.org/10.1016/j.jcp.2008.10.008.
[16] E. Hansen, A. Ostermann, High order splitting methods for analytic semigroups exist, BIT Numer. Math. 49 (2009) 527–542, https://doi.org/10.1007/s10543-009-0236-x.
[17] S. Descombes, Convergence of a splitting method of high order for reaction-diffusion systems, Math. Comput. 70 (236) (2001) 1481–1501, https://doi.org/10.1090/S0025-5718-00-01277-1.
[18] M. De Leo, D. Rial, C. Sánchez de la Vega, High-order time-splitting methods for irreversible equations, IMA J. Numer. Anal. 36 (2016) 1842–1866, https://doi.org/10.1093/imanum/drv058.
[19] L.A. Raviola, M.F. De Leo, Performance of affine-splitting pseudo-spectral methods for fractional complex Ginzburg-Landau equations, Appl. Math. Comput. 466 (2024) 128428, https://doi.org/10.1016/j.amc.2023.128428.
[20] J.D. Lawson, Generalized Runge-Kutta processes for stable systems with large Lipschitz constants, SIAM J. Numer. Anal. 4 (3) (1967) 372–380, https://doi.org/10.1137/0704033.
[21] S. Balac, A. Fernandez, F. Mahé, F. Méhats, R. Texier-Picard, The interaction picture method for solving the generalized nonlinear Schrödinger equation in optics, ESAIM: Math. Model. Numer. Anal. 50 (2016) 945–964, https://doi.org/10.1051/m2an/2015060.
[22] C. Besse, G. Dujardin, I. Lacroix-Violet, High order exponential integrators for nonlinear Schrödinger equations with application to rotating Bose–Einstein condensates, SIAM J. Numer. Anal. 55 (3) (2017) 1387–1411, https://doi.org/10.1137/15M1029047.
[23] N. Crouseilles, L. Einkemmer, J. Massot, Exponential methods for solving hyperbolic problems with application to collisionless kinetic equations, J. Comput. Phys. 420 (2020) 109688, https://doi.org/10.1016/j.jcp.2020.109688.
[24] X. Ding, S.H. Kang, Stepsize-adaptive integrators for dissipative solitons in cubic-quintic complex Ginzburg-Landau equations, arXiv preprint, arXiv:1703.09622, 2017, https://doi.org/10.48550/arXiv.1703.09622.
[25] M. Caliari, F. Cassini, L. Einkemmer, A. Ostermann, F. Zivcovich, A μ-mode integrator for solving evolution equations in Kronecker form, J. Comput. Phys. 455 (2022) 110989, https://doi.org/10.1016/j.jcp.2022.110989.
[26] M. Caliari, F. Cassini, F. Zivcovich, A μ-mode BLAS approach for multidimensional tensor-structured problems, Numer. Algorithms 92 (4) (2023) 2483–2508, https://doi.org/10.1007/s11075-022-01399-4.
[27] D. Wang, M. Li, Y. Lu, Unconditionally convergent and superconvergent analysis of second-order weighted IMEX FEMs for nonlinear Ginzburg-Landau equation, Comput. Math. Appl. 146 (2023) 84–105, https://doi.org/10.1016/j.camwa.2023.06.033.
[28] H. Montanelli, N. Bootland, Solving periodic semilinear stiff PDEs in 1D, 2D and 3D with exponential integrators, Math. Comput. Simul. 178 (2020) 307–327, https://doi.org/10.1016/j.matcom.2020.06.008.
[29] G. Rousseau, H. Chaté, R. Kapral, Twisted vortex filaments in the three-dimensional complex Ginzburg–Landau equation, Chaos 18 (2008) 026103, https://doi.org/10.1063/1.2940439.
[30] Y. Zhang, W. Bao, Q. Du, Numerical simulation of vortex dynamics in Ginzburg-Landau-Schrödinger equation, Eur. J. Appl. Math. 18 (2007) 607–630, https://doi.org/10.1017/S0956792507007140.

[31] H. Wang, An efficient Chebyshev–Tau spectral method for Ginzburg–Landau–Schrödinger equations, Comput. Phys. Commun. 181 (2010) 325–340, https://doi.org/10.1016/j.cpc.2009.10.007.
[32] F. Castella, P. Chartier, S. Descombes, G. Vilmart, Splitting methods with complex times for parabolic equations, BIT Numer. Math. 49 (2009) 487–508, https://doi.org/10.1007/s10543-009-0235-y.
[33] M. Hochbruck, A. Ostermann, Exponential integrators, Acta Numer. 19 (2010) 209–286, https://doi.org/10.1017/S0962492910000048.
[34] A.H. Al-Mohy, N.J. Higham, A new scaling and squaring algorithm for the matrix exponential, SIAM J. Matrix Anal. Appl. 31 (3) (2010) 970–989, https://doi.org/10.1137/09074721X.
[35] M. Caliari, F. Zivcovich, On-the-fly backward error estimate for matrix exponential approximation by Taylor algorithm, J. Comput. Appl. Math. 346 (2019) 532–548, https://doi.org/10.1016/j.cam.2018.07.042.
[36] J. Sastre, J. Ibáñez, E. Defez, Boosting the computation of the matrix exponential, Appl. Math. Comput. 340 (2019) 206–220, https://doi.org/10.1016/j.amc.2018.08.017.
[37] J.M. Alonso, J. Ibáñez, E. Defez, P. Alonso-Jordá, Euler polynomials for the matrix exponential approximation, J. Comput. Appl. Math. 425 (2023) 115074, https://doi.org/10.1016/j.cam.2023.115074.
[38] J.P. Boyd, Chebyshev and Fourier Spectral Methods, 2nd edition, DOVER Publications, Inc., Mineola, New York, 2000.
[39] M. Caliari, F. Cassini, L. Einkemmer, A. Ostermann, Accelerating exponential integrators to efficiently solve semilinear advection-diffusion-reaction equations, SIAM J. Sci. Comput. 46 (2) (2024) A906–A928, https://doi.org/10.1137/23M1562056.
[40] Y.-L. Zhao, A. Ostermann, X.-M. Gu, A low-rank Lie-Trotter splitting approach for nonlinear fractional complex Ginzburg-Landau equations, J. Comput. Phys. 446 (2021) 110652, https://doi.org/10.1016/j.jcp.2021.110652.
[41] L. Zhang, Q. Zhang, H.-W. Sun, Preconditioned fourth-order exponential integrator for two-dimensional nonlinear fractional Ginzburg-Landau equation, Comput. Math. Appl. 150 (2023) 211–228, https://doi.org/10.1016/j.camwa.2023.09.029.
[42] M. Caliari, F. Cassini, A second order directional split exponential integrator for systems of advection–diffusion–reaction equations, J. Comput. Phys. 498 (2024) 112640, https://doi.org/10.1016/j.jcp.2023.112640.
[43] F. Cassini, Efficient third order tensor-oriented directional splitting for exponential integrators, arXiv preprint, arXiv:2310.07551, 2024, https://doi.org/10.48550/arXiv.2310.07551.
[44] I.S. Aranson, A.R. Bishop, L. Kramer, Dynamics of vortex lines in the three-dimensional complex Ginzburg-Landau equation: instability, stretching, entanglement, and helices, Phys. Rev. E 57 (5) (1998) 5276–5286, https://doi.org/10.1103/PhysRevE.57.5276.
[45] B. Liu, Y.-J. He, Z.-R. Qiu, H.-Z. Wang, Annularly and radially phase-modulated spatiotemporal necklace-ring patterns in the Ginzburg–Landau and Swift–Hohenberg equations, Opt. Express 17 (2009) 12203–12209, https://doi.org/10.1364/OE.17.012203.
[46] J. Zhang, G. Yan, M. Wang, Lattice Boltzmann simulations for the vortex tori pattern in the three-dimensional cubic-quintic complex Ginzburg–Landau equation, J. Comput. Phys. 306 (2016) 311–319, https://doi.org/10.1016/j.jcp.2015.11.039.
[47] O. Descalzi, H.R. Brand, Class of compound dissipative solitons as a result of collisions in one and two spatial dimensions, Phys. Rev. E 90 (2014) 020901(R), https://doi.org/10.1103/PhysRevE.90.020901.
[48] A. Sigler, B.A. Malomed, D.V. Skryabin, Localized states in a triangular set of linearly coupled complex Ginzburg-Landau equations, Phys. Rev. E 74 (2006) 066604, https://doi.org/10.1103/PhysRevE.74.066604.