Computer Physics Communications 306 (2025) 109378

Contents lists available at ScienceDirect

Computer Physics Communications

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

Computer Programs in Physics

Parallel finite-element codes for the Bogoliubov-de Gennes stability analysis of Bose-Einstein condensates

Georges Sadaka [a], Pierre Jolivet [b], Efstathios G. Charalampidis [a,c,d], Ionut Danaila [a,*]

[a] Univ Rouen Normandie, CNRS, Laboratoire de Mathématiques Raphaël Salem, UMR 6085, F-76000 Rouen, France
[b] Sorbonne Université - CNRS, LIP6, 4 place Jussieu, 75252 Paris, France
[c] Mathematics Department, California Polytechnic State University, San Luis Obispo, CA 93407-0403, USA
[d] Department of Mathematics and Statistics and Computational Sciences Research Center, San Diego State University, San Diego, CA 92182-7720, USA

ARTICLE INFO

Keywords:
Bose-Einstein condensate
Gross-Pitaevskii equation
Bogoliubov-de Gennes equation
Finite elements
FreeFEM
PETSc
SLEPc

ABSTRACT

We present and distribute a parallel finite-element toolbox written in the free software `FreeFEM` for computing the Bogoliubov-de Gennes (BdG) spectrum of stationary solutions to one- and two-component Gross-Pitaevskii (GP) equations, in two or three spatial dimensions. The parallelization of the toolbox relies exclusively upon the recent interfacing of `FreeFEM` with the `PETSc` library. The latter contains itself a wide palette of state-of-the-art linear algebra libraries, graph partitioners, mesh generation and domain decomposition tools, as well as a suite of eigenvalue solvers that are embodied in the `SLEPc` library. Within the present toolbox, stationary states of the GP equations are computed by a Newton method. Branches of solutions are constructed using an adaptive step-size continuation algorithm. The combination of mesh adaptivity tools from `FreeFEM` with the parallelization features from `PETSc` makes the toolbox efficient and reliable for the computation of stationary states. Their BdG spectrum is computed using the `SLEPc` eigenvalue solver. We perform extensive tests and validate our programs by comparing the toolbox's results with known theoretical and numerical findings that have been reported in the literature.

Program summary

Program Title: FFEM_BdG_ddm_toolbox.zip
CPC Library link to program files: https://doi.org/10.17632/w9hg964wpb.1
Licensing provisions: GPLv3
Programming language: `FreeFEM` (v 4.12) free software (www.freefem.org)
Nature of problem: Among the plethora of configurations that may exist in Gross-Pitaevskii (GP) equations modeling one or two-component Bose-Einstein condensates, only the ones that are deemed spectrally stable (or even, in some cases, weakly unstable) have high probability to be observed in realistic ultracold atoms experiments. To investigate the spectral stability of solutions requires the numerical study of the linearization of GP equations, the latter commonly known as the Bogoliubov-de Gennes (BdG) spectral problem. The present software offers an efficient and reliable tool for the computation of eigenvalues (or modes) of the BdG problem for a given two- or three-dimensional GP configuration. Then, the spectral stability (or instability) can be inferred from its spectrum, thus predicting (or not) its observability in experiments.
Solution method: The present toolbox in `FreeFEM` consists of the following steps. At first, the GP equations in two (2D) and three (3D) spatial dimensions are discretized by using P2 (piece-wise quadratic) Galerkin triangular (in 2D) or tetrahedral (in 3D) finite elements. For a given configuration of interest, mesh adaptivity in `FreeFEM` is deployed in order to reduce the size of the problem, thus reducing the toolbox's execution time. Then, stationary states of the GP equations are obtained by a Newton method whose backbone involves the use of a reliable and efficient linear solver judiciously selected from the `PETSc`[1] library. Upon identifying stationary configurations, to trace branches of such solutions a parameter continuation method over the chemical potential in the GP equations (effectively controlling the number of atoms in a BEC) is employed with step-size adaptivity of the continuation parameter. Finally, the computation of the stability of branches of solutions (*i.e.* the BdG spectrum), is carried

☆ The review of this paper was arranged by Prof. Andrew Hazel.

* Corresponding author.
E-mail addresses: georges.sadaka@univ-rouen.fr (G. Sadaka), pierre@joliv.et (P. Jolivet), echaralampidis@sdsu.edu (E.G. Charalampidis), ionut.danaila@univ-rouen.fr (I. Danaila).

https://doi.org/10.1016/j.cpc.2024.109378
Received 17 March 2024; Received in revised form 10 August 2024; Accepted 10 September 2024
Available online 13 September 2024

out by accurately solving, at each point in the parameter space, the underlying eigenvalue problem by using the SLEPc[2] library. Three-dimensional computations are made affordable in the present toolbox by using the domain decomposition method (DDM). In the course of the computation, the toolbox stores not only the solutions but also the eigenvalues and respective eigenvectors emanating from the solution to the BdG problem. We offer examples for computing stationary configurations and their BdG spectrum in one- and two-component GP equations.
Running time: From minutes to hours depending on the mesh resolution and space dimension.

1. Introduction

The study of Bose-Einstein condensates (BECs) has admittedly enjoyed a substantial interest for more than two decades since their first observation in ultracold atoms experiments [1,2]. Both theoretical and experimental developments have been summarized in [3,4]. These studies revealed the emergence of interesting wave configurations including vortices and vortex structures [5–8], and the quest for experimentally creating and studying new states has been an exciting and active area of research. Indeed, a plethora of experimental techniques have been developed including imprinting techniques [9–11], stirring the condensate above a certain critical angular speed [12,13], counterflow techniques [14], the use of anisotropic potentials [15], nonlinear interference between different condensate fragments [16] as well as the so-called Kibble-Zurek mechanism [17], among many others. The variety of configurations that have emerged through these studies is vast, and has sparked theoretical and computational investigations over the years. Some basic examples of such structures are dark solitons, single vortex lines (with I-, U- or S-shaped ones in rotated BECs) [18], as well as vortex rings [19] (see also the review [20] and references therein). More complex states, such as multiple vortex lines and rings, vortex stars, and hopfions have also been reported in the literature (see for example [21–23]), together with recent computational techniques for vortex identification [24]. Alongside prototypical bound modes in multi-component BECs that can be identified (*e.g.* dark-bright [25], vortex-bright [26,27] and dark-antidark [28], as well as vortex-ring-bright and vortex-line-bright solitons [29]), more exotic configurations have been reported, including skyrmions [30,31], monopoles and Alice rings [32,33]. Even more, with the introduction of state-of-the-art bifurcation techniques for partial differential equations (PDEs), more and more multi-component solutions were identified [34,35].

The principal model for the above theoretical and computational studies has been the Gross-Pitaevskii (GP) equation [4] (and variants, including multi-component settings), which is a PDE known to describe the properties of a BEC in the mean-field approximation. Note that the GP model is a nonlinear Schrödinger (NLS) equation that incorporates an external potential to confine the atoms in the condensate [4]. One of the key steps in these studies, however, is concerned with the response of the pertinent waveforms under the presence of a perturbation induced, *e.g.* by imperfections in the initial state preparation in the BEC. This crucial step involves the study of the spectral stability [36] of the solution to the GP equation at the theoretical/computational level, and it is a two-fold process. At first, a stationary solution to the GP equation is identified by means of (spatial) discretization methods and root-finding, *i.e.* fixed-point techniques. Then, the GP equation is linearized about this (stationary) solution, thus resulting into a spectral (eigenvalue) problem, known as the Bogoliubov-de Gennes (BdG) problem [37,38]. The numerical solution of the BdG problem provides important information about the spectral characteristics of waveforms that may have high probability to be observed numerically if they are deemed stable (or even weakly unstable, depending on the growth rates of the unstable eigenvalues).

Up until now, there has been a wide variety of publicly available programs (written in `C`, `Fortran`, `MATLAB`, and `FreeFEM`) dedicated exclusively to the computation of stationary states to the GP equation that employ spectral methods [39–41], finite elements [42,43] (see also [44]), and finite differences [45–49]. In almost all of these studies, stationary solutions are computed by solving the GP equation under the constraint of the conservation of the number of atoms; a large variety of numerical algorithms can be used for the constrained problem, among which the celebrated *normalized gradient method* [50] is one of the most popular. Alternative approaches for the computation of solutions to the GP equation involve the re-formulation of the problem as a bifurcation one, where the chemical potential (controlling the number of atoms) is varied by using numerical continuation [51] coupled with Newton's method [52]. This approach has been adopted in a series of studies that employ finite-element [28,34,53,54], finite-difference [22,23,34], as well as spectral (spatial) discretization methods [23]. However, to the best of our knowledge, a limited number of publicly available codes for studying the BdG spectrum of configurations to the GP equation exist. One such a code (written in `Fortran`) is the FACt toolbox [55] which computes thermal fluctuations in BECs by solving the associated BdG equations. Recent efforts in that same vein involve the publicly available toolbox in `FreeFEM` that was developed by a subset of the present authors [56]. It utilizes mesh adaptation techniques (that are built-in in `FreeFEM`) and employs the ARPACK eigenvalue solver [57] (which is interfaced with `FreeFEM`) for solving the BdG equations, although the calculations therein are carried out in sequential mode.

Building upon the recent work in [56], we present and distribute herein a *parallel* finite-element toolbox written in `FreeFEM` for computing the Bogoliubov-de Gennes (BdG) spectrum of stationary solutions to one- and two-component Gross-Pitaevskii (GP) equations in 2D or 3D. The parallelization of the toolbox relies exclusively on the recent interfacing of `FreeFEM` with the `PETSc` library [58] (see also [59]). The combination of mesh adaptivity and the simplification in the use of parallel linear solvers in `FreeFEM` (such as distributed direct solvers and domain decomposition methods [60,61]) renders the present toolbox an ideal framework for computing configurations in one- and two-component BECs in 2D or 3D. This further paves the path for the efficient and reliable computation of the BdG spectrum by using the `SLEPc` [62] library.

Our ultimate goal with the present toolbox is to offer a versatile and reliable tool to the BEC community which can perform parallel computations for exploring the BdG spectrum of 2D and 3D (one- or two-component) configurations of interest within reasonable computational time. Finally, the advantage of `FreeFEM` in hiding all technicalities of the finite-element method and using a syntax close to the mathematical formulation of the problem allows the user to focus on the mathematical and physical aspects of the problem and easily make changes in the codes to simulate new configurations.

The structure of the paper is as follows. In Sec. 2, we introduce the one- and two-component GP equations together with the associated BdG models. In Secs. 3 and 4, we describe the numerical methods for computing stationary states to the GP equations and their respective BdG spectra. We illustrate the validation of our programs in Secs. 5 and 6, whereas the architecture of the programs and the description of parameter and output files is discussed in Sec. 7. Finally, the main features of the toolbox are summarized in Sec. 8, where we additionally offer some of its potential extensions.

[1] https://petsc.org/.
[2] https://slepc.upv.es.

2. The Gross-Pitaevskii model and Bogoliubov-de Gennes equations

In this section, we present the theoretical setup of the toolbox. We introduce the one- and two-component Gross-Pitaevskii (GP) and Bogoliubov-de Gennes equations in Secs. 2.1 and 2.2, respectively. We would like to stress out that the model equations below are expressed in nondimensional form, and further details about the physical units of the model equations together with their scaling can be found in our recent contribution [56] (and references therein). For the user's convenience, we include in this toolbox example scripts (see files `phys_to_adim_1comp.edp` and `phys_to_adim_2comp.edp` in the `Tools_scaling` subdirectory) that compute non-dimensional parameters from physical values corresponding to several experimental studies published in the literature. These programs could guide the user in linking parameters of existing experiments with non-dimensional parameters used in this contribution (and, more generally, in theoretical studies).

2.1. The one-component case: Gross-Pitaevskii and Bogoliubov-de Gennes equations

The spatio-temporal behavior of a Bose-Einstein condensate (BEC) is described by the Gross-Pitaevskii (GP) equation [4]:

$$\mathrm{i}\frac{\partial\psi}{\partial t} = -\frac{1}{2}\nabla^2\psi + C_{\mathrm{trap}}\psi + \beta|\psi|^2\psi, \tag{1}$$

where $\psi(\boldsymbol{x},t) : \mathcal{D}\times\mathbb{R}^+ \to \mathbb{C}$ is the macroscopic complex-valued wave function, defined over the domain $\mathcal{D}\in\mathbb{R}^d$, with d the spatial dimension, *i.e.* $d = 1, 2, 3$. In Eq. (1), β is the nonlinearity strength corresponding to repulsive ($\beta > 0$) or attractive ($\beta < 0$) interactions. The external potential $C_{\mathrm{trap}}(\boldsymbol{x})$ confining the atoms in the condensate is:

$$C_{\mathrm{trap}}(x,y,z) = \frac{1}{2}\left(\omega_x^2x^2 + \omega_y^2y^2 + \omega_z^2z^2\right), \tag{2}$$

where $\omega_x, \omega_y, \omega_z$ are the trapping frequencies. Using in Eq. (1) the standing wave ansatz $\psi(\boldsymbol{x},t) = \phi(\boldsymbol{x})e^{-\mathrm{i}\mu t}$, with μ the chemical potential, we obtain the stationary GP equation:

$$-\frac{1}{2}\nabla^2\phi + C_{\mathrm{trap}}\phi + \beta|\phi|^2 = \mu\phi. \tag{3}$$

In this work, we compute stationary solutions to Eq. (3) with homogeneous Dirichlet boundary conditions (*i.e.* $\phi = 0$ on $\partial\mathcal{D}$) for fixed μ. Branches of such solutions are obtained by performing numerical continuation [51] over μ, which corresponds here to a bifurcation parameter. At each step in the continuation process, we monitor the energy

$$\mathcal{E}(\phi) = \int_{\mathcal{D}}\left(\frac{1}{2}|\nabla\phi(\boldsymbol{x})|^2 + C_{\mathrm{trap}}(\boldsymbol{x})|\phi(\boldsymbol{x})|^2 + \frac{\beta}{2}|\phi(\boldsymbol{x})|^4\right)d\boldsymbol{x}, \tag{4}$$

and the total number of atoms

$$N(\phi) = \int_{\mathcal{D}}\phi\overline{\phi}\,d\boldsymbol{x} = \int_{\mathcal{D}}|\phi|^2 d\boldsymbol{x}. \tag{5}$$

The Bogoliubov-de Gennes (BdG) problem we aim to solve is obtained by using in Eq. (1) the ansatz

$$\psi(\boldsymbol{x},t) = \left[\phi(\boldsymbol{x}) + \delta\left(A(\boldsymbol{x})e^{-\mathrm{i}\omega t} + \overline{B}(\boldsymbol{x})e^{\mathrm{i}\overline{\omega}t}\right)\right]e^{-\mathrm{i}\mu t}, \quad \delta \ll 1, \tag{6}$$

where $\phi(\boldsymbol{x})$ is a stationary state, A and B are complex-valued functions, and ω is a complex number. We obtain the linear eigenvalue problem called the BdG equation:

$$\begin{pmatrix} \mathcal{H} - \mu + 2\beta|\phi|^2 & \beta\phi^2 \\ -\beta\overline{\phi}^2 & -(\mathcal{H} - \mu + 2\beta|\phi|^2) \end{pmatrix}\begin{pmatrix} A \\ B \end{pmatrix} = \omega\begin{pmatrix} A \\ B \end{pmatrix}, \tag{7}$$

where

$$\mathcal{H} \equiv -\frac{1}{2}\nabla^2 + C_{\mathrm{trap}}. \tag{8}$$

The present toolbox computes the eigenvalue-eigenvector pair (ω, A, B) for a given stationary solution ϕ. Note that we consider the most general case, with $\phi\in\mathbb{C}$ and $\omega\in\mathbb{C}$. The particular case of real solutions ($\phi\in\mathbb{R}$) and real eigenvalues ($\omega\in\mathbb{R}$), corresponding to elementary excitations, was recently studied in [63]. We direct the reader to [56] for a detailed discussion on the properties of the general BdG problem.

2.2. The two-component case: Gross-Pitaevskii and Bogoliubov-de Gennes equations

A mixture of two BECs (*e.g.* different hyperfine states of the same species) is described by a coupled system of two GP equations [3,4,19]:

$$\begin{cases} \mathrm{i}\dfrac{\partial\psi_1}{\partial t} = \left(-\dfrac{1}{2}\nabla^2 + C_{\mathrm{trap}} + \beta_{11}|\psi_1|^2 + \beta_{12}|\psi_2|^2\right)\psi_1, \\ \mathrm{i}\dfrac{\partial\psi_2}{\partial t} = \left(-\dfrac{1}{2}\nabla^2 + C_{\mathrm{trap}} + \beta_{21}|\psi_1|^2 + \beta_{22}|\psi_2|^2\right)\psi_2. \end{cases} \tag{9}$$

Coefficients β_{11} and β_{22} in (9) represent the interaction strengths between atoms of same species, whereas the β_{12} and β_{21} represent the ones between different species. For mathematical models involving more than two GP equations (*e.g.* for spinor BEC), we refer to [64]. Similarly to the one-component case, using in Eqs. (9) the Ansätze $\psi_j(\boldsymbol{x},t) = \phi_j(\boldsymbol{x})e^{-\mathrm{i}\mu_j t}$, $j = 1, 2$, with chemical potentials μ_1 and μ_2, we obtain the following coupled system of stationary GP equations:

$$\begin{cases} \mu_1\phi_1 = \left(-\dfrac{1}{2}\nabla^2 + C_{\mathrm{trap}} + \beta_{11}|\phi_1|^2 + \beta_{12}|\phi_2|^2\right)\phi_1, \\ \mu_2\phi_2 = \left(-\dfrac{1}{2}\nabla^2 + C_{\mathrm{trap}} + \beta_{21}|\phi_1|^2 + \beta_{22}|\phi_2|^2\right)\phi_2, \end{cases} \tag{10}$$

The system (10), with homogeneous Dirichlet boundary conditions ($\phi_j = 0$ on $\partial\mathcal{D}$ for $j = 1, 2$), is solved for fixed values of μ_1 and μ_2. The characterization of a stationary solution is based on the total energy

$$\mathcal{E}(\phi_1,\phi_2) = \int_{\mathcal{D}}\sum_{i=1}^{2}\left(\frac{1}{2}|\nabla\phi_i|^2 + C_{\mathrm{trap}}|\phi_i|^2 + \frac{1}{2}\sum_{j=1}^{2}\beta_{ij}|\phi_i|^2|\phi_j|^2\right)d\boldsymbol{x}, \tag{11}$$

as well as the total number of atoms $N(\phi_1,\phi_2) = N(\phi_1) + N(\phi_2)$, where $N(\cdot)$ is given by Eq. (5). To study the spectral stability of stationary solutions we consider the Ansätze

$$\psi_1(\boldsymbol{x},t) = \left[\phi_1(\boldsymbol{x}) + \delta\left(A(\boldsymbol{x})e^{-\mathrm{i}\omega t} + \overline{B}(\boldsymbol{x})e^{\mathrm{i}\overline{\omega}t}\right)\right]e^{-\mathrm{i}\mu_1 t}, \tag{12a}$$

$$\psi_2(\boldsymbol{x},t) = \left[\phi_2(\boldsymbol{x}) + \delta\left(C(\boldsymbol{x})e^{-\mathrm{i}\omega t} + \overline{D}(\boldsymbol{x})e^{\mathrm{i}\overline{\omega}t}\right)\right]e^{-\mathrm{i}\mu_2 t}, \tag{12b}$$

with $A, B, C, D, \omega\in\mathbb{C}$, and obtain the BdG equations for the two-component case:

$$M\begin{pmatrix} A \\ B \\ C \\ D \end{pmatrix} = \omega\begin{pmatrix} A \\ B \\ C \\ D \end{pmatrix},$$

$$M = \begin{pmatrix} M_{11} & \beta_{11}\phi_1^2 & \beta_{12}\phi_1\overline{\phi_2} & \beta_{12}\phi_1\phi_2 \\ -\beta_{11}\overline{\phi_1}^2 & M_{22} & -\beta_{12}\overline{\phi_1}\phi_2 & -\beta_{12}\overline{\phi_1}\phi_2 \\ \beta_{21}\overline{\phi_1}\phi_2 & \beta_{21}\phi_1\phi_2 & M_{33} & \beta_{22}\phi_2^2 \\ -\beta_{21}\overline{\phi_1}\phi_2 & -\beta_{21}\phi_1\overline{\phi_2} & -\beta_{22}\overline{\phi_2}^2 & M_{44} \end{pmatrix}, \tag{13}$$

with matrix elements

$$\begin{cases} M_{11} &= \mathcal{H} - \mu_1 + 2\beta_{11}|\phi_1|^2 + \beta_{12}|\phi_2|^2, \\ M_{22} &= -M_{11}, \\ M_{33} &= \mathcal{H} - \mu_2 + \beta_{21}|\phi_1|^2 + 2\beta_{22}|\phi_2|^2, \\ M_{44} &= -M_{33}, \end{cases} \tag{14}$$

and $\mathcal{H}$ given by Eq. (8).

3. The computation of stationary solutions to the GP equations

3.1. Newton's method for a single-component BEC

For the computation of stationary solutions to Eq. (3), we use Newton's method [52]. We first split the complex-valued wave function ϕ into real and imaginary parts via $\phi = \phi_r + \mathrm{i}\,\phi_i$ and obtain from (3) the following coupled system of nonlinear equations

$$\begin{cases} -\frac{1}{2}\nabla^2\phi_r + C_{\mathrm{trap}}\phi_r + \beta f(\phi_r,\phi_i)\phi_r - \mu\phi_r = 0, \\ -\frac{1}{2}\nabla^2\phi_i + C_{\mathrm{trap}}\phi_i + \beta f(\phi_r,\phi_i)\phi_i - \mu\phi_i = 0. \end{cases} \tag{15}$$

We introduced in (15) the (scalar) function $f(\phi_r,\phi_i) = |\phi|^2 = \phi_r^2 + \phi_i^2$ that corresponds to the cubic nonlinearity in the GP equation. Note that the expression of f has been programmed in the toolbox in a general way; other types of expressions (corresponding to the GP equation with different nonlinearity than cubic) can be used and easily implemented in the toolbox.

The homogeneous Dirichlet conditions for the complex-valued wave function ϕ translate into imposing $\phi_r = \phi_i = 0$ on ∂D. After setting the classical Sobolev spaces [65] $V = H_0^1(D)$ for ϕ_r and ϕ_i, we define the weak formulation (mandatory for the finite-element implementation) of Eq. (15) as: find $(\phi_r,\phi_i) \in V \times V = V^2$, such that for all test functions $(v_r, v_i) \in V^2$

$$\begin{cases} \mathcal{F}_r(\phi_r,\phi_i,v_r) = \displaystyle\int_D (C_{\mathrm{trap}} - \mu)\phi_r v_r + \int_D \frac{1}{2}\nabla\phi_r\cdot\nabla v_r \\ \qquad\qquad + \displaystyle\int_D \beta f(\phi_r,\phi_i)\phi_r v_r = 0, \\ \mathcal{F}_i(\phi_r,\phi_i,v_i) = \displaystyle\int_D (C_{\mathrm{trap}} - \mu)\phi_i v_i + \int_D \frac{1}{2}\nabla\phi_i\cdot\nabla v_i \\ \qquad\qquad + \displaystyle\int_D \beta f(\phi_r,\phi_i)\phi_i v_i = 0. \end{cases} \tag{16}$$

The above coupled system of nonlinear equations is discretized using finite elements in `FreeFEM` (see Sec. 3.3), and solved by means of Newton's method which requires a sufficiently good initial guess. For a given value of μ and an initial guess (ϕ_r^0,ϕ_i^0), Newton's method computes corrections to the solution components (ϕ_r,ϕ_i) iteratively via

$$q = \phi_r^k - \phi_r^{k+1}, \quad s = \phi_i^k - \phi_i^{k+1}, \quad k \geq 0, \tag{17}$$

where q and s are solutions of the linearized equations

$$\begin{pmatrix} \left(\frac{\partial\mathcal{F}_r}{\partial\phi_r}\right)_{\phi_r=\phi_r^k,\phi_i=\phi_i^k} & \left(\frac{\partial\mathcal{F}_r}{\partial\phi_i}\right)_{\phi_r=\phi_r^k,\phi_i=\phi_i^k} \\ \left(\frac{\partial\mathcal{F}_i}{\partial\phi_r}\right)_{\phi_r=\phi_r^k,\phi_i=\phi_i^k} & \left(\frac{\partial\mathcal{F}_i}{\partial\phi_i}\right)_{\phi_r=\phi_r^k,\phi_i=\phi_i^k} \end{pmatrix} \begin{pmatrix} q \\ s \end{pmatrix} = \begin{pmatrix} \mathcal{F}_r(\phi_r^k,\phi_i^k,v_r) \\ \mathcal{F}_i(\phi_r^k,\phi_i^k,v_i) \end{pmatrix}, \tag{18}$$

with the corresponding weak formulation

$$\begin{cases} \displaystyle\int_D (C_{\mathrm{trap}} - \mu)qv_r + \int_D \frac{1}{2}\nabla q\cdot\nabla v_r \\ \qquad + \displaystyle\int_D \beta\left(\frac{\partial f}{\partial\phi_r}(\phi_r^k,\phi_i^k)\,\phi_r^k q + \frac{\partial f}{\partial\phi_i}(\phi_r^k,\phi_i^k)\,\phi_r^k s + f(\phi_r^k,\phi_i^k)q\right)v_r \\ \qquad = \displaystyle\int_D (C_{\mathrm{trap}} - \mu)\phi_r^k v_r + \int_D \frac{1}{2}\nabla\phi_r^k\cdot\nabla v_r + \int_D \beta f(\phi_r^k,\phi_i^k)\phi_r^k v_r, \\ \displaystyle\int_D (C_{\mathrm{trap}} - \mu)sv_i + \int_D \frac{1}{2}\nabla s\cdot\nabla v_i \\ \qquad + \displaystyle\int_D \beta\left(\frac{\partial f}{\partial\phi_r}(\phi_r^k,\phi_i^k)\,\phi_i^k q + \frac{\partial f}{\partial\phi_i}(\phi_r^k,\phi_i^k)\,\phi_i^k s + f(\phi_r^k,\phi_i^k)s\right)v_i \\ \qquad = \displaystyle\int_D (C_{\mathrm{trap}} - \mu)\phi_i^k v_i + \int_D \frac{1}{2}\nabla\phi_i^k\cdot\nabla v_i + \int_D \beta f(\phi_r^k,\phi_i^k)\phi_i^k v_i. \end{cases} \tag{19}$$

Note that the implementation of Eqs. (19) in `FreeFEM` takes a form very similar to the mathematical formulation of the problem due to its versatile metalanguage used therein. This is an advantage for the user who can thus build bug-free numerical codes when cumbersome mathematical expressions are coded.

3.2. Newton's method for the two-component BEC

The two-component GP system (10) is solved similarly by means of Newton's method, after splitting ϕ_1 and ϕ_2 into real and imaginary parts via $\phi_1 = \phi_{1r} + \mathrm{i}\,\phi_{1i}$ and $\phi_2 = \phi_{2r} + \mathrm{i}\,\phi_{2i}$. Equations (10) are thus converted into a system consisting of four real-valued (nonlinear) equations:

$$\begin{cases} -\frac{1}{2}\nabla^2\phi_{1r} + (C_{\mathrm{trap}} - \mu_1)\phi_{1r} + \beta_{11}f(\phi_{1r},\phi_{1i})\phi_{1r} + \beta_{12}f(\phi_{2r},\phi_{2i})\phi_{1r} = 0, \\ -\frac{1}{2}\nabla^2\phi_{1i} + (C_{\mathrm{trap}} - \mu_1)\phi_{1i} + \beta_{11}f(\phi_{1r},\phi_{1i})\phi_{1i} + \beta_{12}f(\phi_{2r},\phi_{2i})\phi_{1i} = 0, \\ -\frac{1}{2}\nabla^2\phi_{2r} + (C_{\mathrm{trap}} - \mu_2)\phi_{2r} + \beta_{21}f(\phi_{1r},\phi_{1i})\phi_{2r} + \beta_{22}f(\phi_{2r},\phi_{2i})\phi_{2r} = 0, \\ -\frac{1}{2}\nabla^2\phi_{2i} + (C_{\mathrm{trap}} - \mu_2)\phi_{2i} + \beta_{21}f(\phi_{1r},\phi_{1i})\phi_{2i} + \beta_{22}f(\phi_{2r},\phi_{2i})\phi_{2i} = 0. \end{cases} \tag{20}$$

Again, homogeneous Dirichlet boundary conditions on ϕ_1 and ϕ_2 are imposed: $\phi_{1r} = \phi_{1i} = \phi_{2r} = \phi_{2i} = 0$ on ∂D. The weak formulation of Eqs. (20) can be written as follows: find $(\phi_{1r},\phi_{1i},\phi_{2r},\phi_{2i}) \in V^4$, such that for all test functions $(v_{1r},v_{1i},v_{2r},v_{2i}) \in V^4$

$$\begin{cases} \mathcal{F}_{1r} = \displaystyle\int_D (C_{\mathrm{trap}} - \mu_1)\phi_{1r}v_{1r} + \int_D \frac{1}{2}\nabla\phi_{1r}\cdot\nabla v_{1r} + \int_D (\beta_{11}f(\phi_{1r},\phi_{1i}) \\ \qquad + \beta_{12}f(\phi_{2r},\phi_{2i}))\phi_{1r}v_{1r} = 0, \\ \mathcal{F}_{1i} = \displaystyle\int_D (C_{\mathrm{trap}} - \mu_1)\phi_{1i}v_{1i} + \int_D \frac{1}{2}\nabla\phi_{1i}\cdot\nabla v_{1i} + \int_D (\beta_{11}f(\phi_{1r},\phi_{1i}) \\ \qquad + \beta_{12}f(\phi_{2r},\phi_{2i}))\phi_{1i}v_{1i} = 0, \\ \mathcal{F}_{2r} = \displaystyle\int_D (C_{\mathrm{trap}} - \mu_2)\phi_{2r}v_{2r} + \int_D \frac{1}{2}\nabla\phi_{2r}\cdot\nabla v_{2r} + \int_D (\beta_{21}f(\phi_{1r},\phi_{1i}) \\ \qquad + \beta_{22}f(\phi_{2r},\phi_{2i}))\phi_{2r}v_{2r} = 0, \\ \mathcal{F}_{2i} = \displaystyle\int_D (C_{\mathrm{trap}} - \mu_2)\phi_{2i}v_{2i} + \int_D \frac{1}{2}\nabla\phi_{2i}\cdot\nabla v_{2i} + \int_D (\beta_{21}f(\phi_{1r},\phi_{1i}) \\ \qquad + \beta_{22}f(\phi_{2r},\phi_{2i}))\phi_{2i}v_{2i} = 0. \end{cases} \tag{21}$$

Newton's method computes, for fixed chemical potentials μ_1 and μ_2 and given initial guess $\left(\phi_{1r}^0,\phi_{1i}^0,\phi_{2r}^0,\phi_{2i}^0\right)$, the corrections

$$q_1 = \phi_{1r}^k - \phi_{1r}^{k+1}, \quad s_1 = \phi_{1i}^k - \phi_{1i}^{k+1}, \quad q_2 = \phi_{2r}^k - \phi_{2r}^{k+1}, \quad s_2 = \phi_{2i}^k - \phi_{2i}^{k+1}, \tag{22}$$

which are solutions to the following system of linear equations:

$$\begin{aligned}
&\int_{\mathcal{D}} (C_{\text{trap}} - \mu_1) q_1 v_{1r} + \int_{\mathcal{D}} \frac{1}{2} \nabla q_1 \cdot \nabla v_{1r} \\
&\quad + \int_{\mathcal{D}} (\beta_{11} f(\phi_{1r}^k, \phi_{1i}^k) + \beta_{12} f(\phi_{2r}^k, \phi_{2i}^k)) q_1 v_{1r} \\
&\quad + \int_{\mathcal{D}} \beta_{11} \left(\frac{\partial f}{\partial \phi_r}(\phi_{1r}^k, \phi_{1i}^k) \phi_{1r}^k q_1 + \frac{\partial f}{\partial \phi_i}(\phi_{1r}^k, \phi_{1i}^k) \phi_{1r}^k s_1 \right) v_{1r} \\
&\quad + \int_{\mathcal{D}} \beta_{12} \left(\frac{\partial f}{\partial \phi_r}(\phi_{2r}^k, \phi_{2i}^k) \phi_{1r}^k q_2 + \frac{\partial f}{\partial \phi_i}(\phi_{2r}^k, \phi_{2i}^k) \phi_{1r}^k s_2 \right) v_{1r} \\
&= \int_{\mathcal{D}} (C_{\text{trap}} - \mu_1) \phi_{1r}^k v_{1r} + \int_{\mathcal{D}} \frac{1}{2} \nabla \phi_{1r}^k \cdot \nabla v_{1r} \\
&\quad + \int_{\mathcal{D}} (\beta_{11} f(\phi_{1r}^k, \phi_{1i}^k) + \beta_{12} f(\phi_{2r}^k, \phi_{2i}^k)) \phi_{1r}^k v_{1r},
\end{aligned} \tag{23}$$

$$\begin{aligned}
&\int_{\mathcal{D}} (C_{\text{trap}} - \mu_1) s_1 v_{1i} + \int_{\mathcal{D}} \frac{1}{2} \nabla s_1 \cdot \nabla v_{1i} \\
&\quad + \int_{\mathcal{D}} (\beta_{11} f(\phi_{1r}^k, \phi_{1i}^k) + \beta_{12} f(\phi_{2r}^k, \phi_{2i}^k)) s_1 v_{1i} \\
&\quad + \int_{\mathcal{D}} \beta_{11} \left(\frac{\partial f}{\partial \phi_r}(\phi_{1r}^k, \phi_{1i}^k) \phi_{1i}^k q_1 + \frac{\partial f}{\partial \phi_i}(\phi_{1r}^k, \phi_{1i}^k) \phi_{1i}^k s_1 \right) v_{1i} \\
&\quad + \int_{\mathcal{D}} \beta_{12} \left(\frac{\partial f}{\partial \phi_r}(\phi_{2r}^k, \phi_{2i}^k) \phi_{1i}^k q_2 + \frac{\partial f}{\partial \phi_i}(\phi_{2r}^k, \phi_{2i}^k) \phi_{1i}^k s_2 \right) v_{1i} \\
&= \int_{\mathcal{D}} (C_{\text{trap}} - \mu_1) \phi_{1i}^k v_{1i} + \int_{\mathcal{D}} \frac{1}{2} \nabla \phi_{1i}^k \cdot \nabla v_{1i} \\
&\quad + \int_{\mathcal{D}} (\beta_{11} f(\phi_{1r}^k, \phi_{1i}^k) + \beta_{12} f(\phi_{2r}^k, \phi_{2i}^k)) \phi_{1i}^k v_{1i},
\end{aligned} \tag{24}$$

$$\begin{aligned}
&\int_{\mathcal{D}} (C_{\text{trap}} - \mu_2) q_2 v_{2r} + \int_{\mathcal{D}} \frac{1}{2} \nabla q_2 \cdot \nabla v_{2r} \\
&\quad + \int_{\mathcal{D}} (\beta_{22} f(\phi_{2r}^k, \phi_{2i}^k) + \beta_{21} f(\phi_{1r}^k, \phi_{1i}^k)) q_2 v_{2r} \\
&\quad + \int_{\mathcal{D}} \beta_{21} \left(\frac{\partial f}{\partial \phi_r}(\phi_{1r}^k, \phi_{1i}^k) \phi_{2r}^k q_1 + \frac{\partial f}{\partial \phi_i}(\phi_{1r}^k, \phi_{1i}^k) \phi_{2r}^k s_1 \right) v_{2r} \\
&\quad + \int_{\mathcal{D}} \beta_{22} \left(\frac{\partial f}{\partial \phi_r}(\phi_{2r}^k, \phi_{2i}^k) \phi_{2r}^k q_2 + \frac{\partial f}{\partial \phi_i}(\phi_{2r}^k, \phi_{2i}^k) \phi_{2r}^k s_2 \right) v_{2r} \\
&= \int_{\mathcal{D}} (C_{\text{trap}} - \mu_2) \phi_{2r}^k v_{2r} + \int_{\mathcal{D}} \frac{1}{2} \nabla \phi_{2r}^k \cdot \nabla v_{2r} \\
&\quad + \int_{\mathcal{D}} (\beta_{21} f(\phi_{1r}^k, \phi_{1i}^k) + \beta_{22} f(\phi_{2r}^k, \phi_{2i}^k)) \phi_{2r}^k v_{2r},
\end{aligned} \tag{25}$$

$$\begin{aligned}
&\int_{\mathcal{D}} (C_{\text{trap}} - \mu_2) s_2 v_{2i} + \int_{\mathcal{D}} \frac{1}{2} \nabla s_2 \cdot \nabla v_{2i} \\
&\quad + \int_{\mathcal{D}} (\beta_{22} f(\phi_{2r}^k, \phi_{2i}^k) + \beta_{21} f(\phi_{1r}^k, \phi_{1i}^k)) s_2 v_{2i} \\
&\quad + \int_{\mathcal{D}} \beta_{21} \left(\frac{\partial f}{\partial \phi_r}(\phi_{1r}^k, \phi_{1i}^k) \phi_{2i}^k q_1 + \frac{\partial f}{\partial \phi_i}(\phi_{1r}^k, \phi_{1i}^k) \phi_{2i}^k s_1 \right) v_{2i} \\
&\quad + \int_{\mathcal{D}} \beta_{22} \left(\frac{\partial f}{\partial \phi_r}(\phi_{2r}^k, \phi_{2i}^k) \phi_{2i}^k q_2 + \frac{\partial f}{\partial \phi_i}(\phi_{2r}^k, \phi_{2i}^k) \phi_{2i}^k s_2 \right) v_{2i} \\
&= \int_{\mathcal{D}} (C_{\text{trap}} - \mu_2) \phi_{2i}^k v_{2i} + \int_{\mathcal{D}} \frac{1}{2} \nabla \phi_{2i}^k \cdot \nabla v_{2i} \\
&\quad + \int_{\mathcal{D}} (\beta_{21} f(\phi_{1r}^k, \phi_{1i}^k) + \beta_{22} f(\phi_{2r}^k, \phi_{2i}^k)) \phi_{2i}^k v_{2i}.
\end{aligned} \tag{26}$$

Again, the implementation of Eqs. (23)-(26) with `FreeFEM` is very similar to the mathematical formulation.

3.3. Finite-element implementation with `FreeFEM`

We now present the finite-element implementation in the free software `FreeFEM` [57] of the weak formulations for the one- and two-component GP equations solved with Newton's method. Note that the main principles of programming and numerical settings presented herein are shared with the implementation of the BdG problem, see Sec. 4.

One of the main advantages while programming in `FreeFEM` is that cumbersome formulas are coded in a compact form, and close to their mathematical formulations. For example, the weak form of the system of linear equations (19) is conveniently implemented as a list of expressions embodied in a `Macro` (see `BdG_1comp_ddm/A_macro/Macro_problem.edp`) in which integral terms are easy to identify:

```
NewMacro problemGP
    macro f(ur,ui) (ur^2 + ui^2)//
    macro dfdur(ur,ui) (2.*ur)//
    macro dfdui(ur,ui) (2.*ui)//

    varf vGP([q,s],[vr,vi]) =
    intN(Th,qforder=ord)((Ctrap - mu)*q*vr + .5*grad
        (q)'*grad(vr)
    + (Ctrap - mu)*s*vi + .5*grad(s)'*grad(vi)
    + beta * (f(phir,phii)*q*vr + f(phir,phii)*s*vi)
    + beta * phir*vr*(dfdur(phir,phii)*q + dfdui(
        phir,phii)*s)
    + beta * phii*vi*(dfdur(phir,phii)*q + dfdui(
        phir,phii)*s))
    + intN(Th,qforder=ord)((Ctrap - mu)*phir*vr +
        .5*grad(phir)'*grad(vr)
    + (Ctrap - mu)*phii*vi + .5*grad(phii)'*grad(vi)
    + beta * f(phir,phii) * (phir*vr + phii*vi))
    BCGP;
EndMacro
```

Another advantage of this formulation in `FreeFEM` is that it can be invariantly used in any (spatial) dimension ($d = 2$ or $d = 3$), and for any available type of finite elements. This is accomplished by simply declaring respective values in the files defining the computational case. Indicatively, for the computation of the 2D ground state using a $P2$ finite-element space, the user can declare (see for example `BdG_1comp_ddm/INIT/2D_ground_state.inc`):

```
macro dimension 2//
macro FEchoice P2//
```

These choices are transmitted in the main programs, see, *e.g.* `FFEM_GP_1c_2D_3D_ddm.edp`:

```
func Pk = [FEchoice,FEchoice];
...
meshN Th; // Local mesh
meshN ThBackup; // Global mesh
fespace Wh(Th,FEchoice);
fespace Whk(Th,Pk);
```

```
fespace WhBackup(ThBackup,FEchoice);
fespace WhkBackup(ThBackup,Pk);
...
Wh<complex> phi, phitemp; // Wavefunction
Whk [q,s], [phir,phii];
WhBackup<complex> phiBackup, phitempBackup; //
    Wavefunction
```

Similarly, for the two-component case, the macro formulation for the linear system (23)-(24) can be found in the file `BdG_2comp_ddm/A_macro/Macro_problem.edp`, and reads

```
NewMacro problemGP
    macro f(ur,ui) (ur^2 + ui^2)//
    macro dfdur(ur,ui) (2.*ur)//
    macro dfdui(ur,ui) (2.*ui)//

    varf vGP([q1,s1,q2,s2],[v1r,v1i,v2r,v2i])=
    intN(Th,qforder=ord)(
    1./2.*grad(q1)'*grad(v1r) + (Ctrap - mu1)*q1*v1r
         + (beta11*f(phi1r,phi1i) + beta12*f(phi2r,
        phi2i))*q1*v1r
    + beta11*(dfdur(phi1r,phi1i)*phi1r*q1 + dfdui(
        phi1r,phi1i)*phi1r*s1)*v1r
    + beta12*(dfdur(phi2r,phi2i)*phi1r*q2 + dfdui(
        phi2r,phi2i)*phi1r*s2)*v1r
    +1./2.*grad(s1)'*grad(v1i) + (Ctrap - mu1)*s1*
        v1i + (beta11*f(phi1r,phi1i) + beta12*f(
        phi2r,phi2i))*s1*v1i
    + beta11*(dfdur(phi1r,phi1i)*phi1i*q1 + dfdui(
        phi1r,phi1i)*phi1i*s1)*v1i
    + beta12*(dfdur(phi2r,phi2i)*phi1i*q2 + dfdui(
        phi2r,phi2i)*phi1i*s2)*v1i
    +1./2.*grad(q2)'*grad(v2r) + (Ctrap - mu2)*q2*
        v2r + (beta22*f(phi2r,phi2i) + beta21*f(
        phi1r,phi1i))*q2*v2r
    + beta22*(dfdur(phi2r,phi2i)*phi2r*q2 + dfdui(
        phi2r,phi2i)*phi2r*s2)*v2r
    + beta21*(dfdur(phi1r,phi1i)*phi2r*q1 + dfdui(
        phi1r,phi1i)*phi2r*s1)*v2r
    +1./2.*grad(s2)'*grad(v2i) + (Ctrap - mu2)*s2*
        v2i + (beta22*f(phi2r,phi2i) + beta21*f(
        phi1r,phi1i))*s2*v2i
    + beta22*(dfdur(phi2r,phi2i)*phi2i*q2 + dfdui(
        phi2r,phi2i)*phi2i*s2)*v2i
    + beta21*(dfdur(phi1r,phi1i)*phi2i*q1 + dfdui(
        phi1r,phi1i)*phi2i*s1)*v2i
    )
    + intN(Th,qforder=ord)(
    1./2.*grad(phi1r)'*grad(v1r) + (Ctrap - mu1)*
        phi1r*v1r + (beta11*f(phi1r,phi1i) + beta12*
        f(phi2r,phi2i))*phi1r*v1r
    + 1./2.*grad(phi1i)'*grad(v1i) + (Ctrap - mu1)*
        phi1i*v1i + (beta11*f(phi1r,phi1i) + beta12*
        f(phi2r,phi2i))*phi1i*v1i
    + 1./2.*grad(phi2r)'*grad(v2r) + (Ctrap - mu2)*
        phi2r*v2r + (beta22*f(phi2r,phi2i) + beta21*
        f(phi1r,phi1i))*phi2r*v2r
    + 1./2.*grad(phi2i)'*grad(v2i) + (Ctrap - mu2)*
        phi2i*v2i + (beta22*f(phi2r,phi2i) + beta21*
        f(phi1r,phi1i))*phi2i*v2i
    )
    BCGP;
EndMacro
```

We highlight here that the user has the flexibility to consider different trapping potentials in the two-component case if necessary. This can be accomplished by modifying the `.inc` files located in the `INIT` subdirectories, and consider, for example, `Ctrap1` and `Ctrap2` for the first and second components, respectively.

The programs that we deliver with this toolbox consider $P2$ (piecewise quadratic) finite elements. The mesh in `FreeFEM` (generically identified as `Th`) is made of triangles in 2D and tetrahedra in 3D. A fast mesh generator with a simple syntax is built in `FreeFEM`. A striking feature of `FreeFEM` is the ability to perform adaptive mesh refinement: the grid is refined in regions of large gradients and coarsened in low-gradient ones. This is of paramount importance, especially for high-dimensional problems where a sufficiently good resolution of the solution is required. Using a very fine mesh (with no mesh adaptation) for the entire domain would lead to a large memory consumption and an excessively long computational time. With the implementation of adaptive mesh refinement in the present toolbox in `FreeFEM`, we maintain reasonable problem sizes, and thus computational time, while keeping a high degree of accuracy.

For 2D configurations, the mesh is adapted by using the built-in `adaptmesh` command in `FreeFEM`. In short (further details can be found in our recent contribution [56]), the underlying algorithm modifies the inner product in the mesh generator to evaluate distance and volume [66–68]. For 3D configurations, adaptive mesh refinement in `FreeFEM` is performed through the use of the libraries `mshmet` and `mmg` [69] where similar algorithms are employed. In the present implementation for computing stationary 2D and 3D configurations to the GP equations, we use adaptive mesh refinement based on the density of the solution as well as its real and imaginary parts. This approach has been considered in [70], and has been proven quite effective in computing complicated vortex solutions.

The underlying nonlinear equations are solved by means of Newton's method which is fed by an initial guess (with fixed chemical potential(s)), see Secs. 5 and 6 for example cases. Newton's iterations are stopped when one of the following criteria is satisfied:

$$\left\| \begin{pmatrix} q \\ s \end{pmatrix} \right\|_\infty < \epsilon_q, \quad \left\| \begin{pmatrix} \mathcal{F}_r \\ \mathcal{F}_i \end{pmatrix} \right\|_2 < \epsilon_F, \tag{27}$$

The former controls the convergence (in the infinity norm) in Newton's method whereas the latter checks the accuracy of the solution (the residual in the L2-norm). In practice, we use $\epsilon_q = 10^{-8}$ and $\epsilon_F = 10^{-16}$ but we found that both criteria are satisfied simultaneously in all the cases we have considered in this paper. Moreover, we note that convergence in Newton's method depends crucially on the choice of the linear solver we employ. Specifically, in 2D, we use an exact LU decomposition, as computed (within the `SLEPc` library) by the MUMPS solver with options:
`"-pc_type lu -ksp_type preonly"`
The computational cost in 2D is thus manageable. For 3D cases, we switch to a more economical preconditioner, and in particular, the algebraic multigrid method which is available in HYPRE with options:
`"-pc_type hypre -ksp_type gmres -ksp_atol 1e-12 -ksp_rtol 1e-6 -ksp_gmres_restart 50 -ksp_max_it 500 -ksp_pc_side right -sub_pc_type lu -sub_pc_factor_mat_solver_type mumps".`

The toolbox can trace branches of stationary configurations to the GP equations by performing numerical continuation [51] over the parameters of the model. For the one-component case, we consider the chemical potential μ as our principal continuation parameter. In particular, we start from a value of the chemical potential μ_0 for which the initial guess is sufficiently close to the stationary state of interest. Upon convergence in Newton's method (discussed above), we use the resulting converged state as an initial guess for the next step in the continuation process with chemical potential $\mu_0 + \delta\mu$. We highlight the fact that we include a simple adaptive strategy for the selection of the increment $\delta\mu$ in the toolbox. Initially, the $\delta\mu$ is fixed to 10^{-3} when μ_0 is close to the respective state's linear limit. It then gets doubled, *i.e.* $\delta\mu = 2\delta\mu$, at every 10 steps in the continuation process until it reaches $\delta\mu_{max} = 0.015$

whereupon it remains fixed, and the continuation stops when the final value μ_f specified by the user is reached.

Finally, for the two-component setting, we follow different continuation strategies that involve relevant principal continuation parameters in order to match the toolbox's results with ones that exist in the literature. For example, the 2D ring-antidark branch is traced by performing continuation over μ_1 and μ_2 first, and then over the inter-component interactions β_{12} and β_{21} (with fixed μ_1 and μ_2). The 2D vortex-antidark branch is traced by fixing μ_1 and μ_2 first, and continuation over the inter-component interactions β_{12} and β_{21} is performed afterwards. Ultimately, various continuation strategies can be conveniently designed and implemented in the toolbox by the user involving different principal continuation parameters.

4. Solving the BdG equations

We solve the BdG problems for the one- and two-component cases by using the `SLEPc` library [62] which is now interfaced with `FreeFEM`. First, we write the weak form of the BdG problems that will be supplied to the solver. Indicatively, and for the one-component case, the weak formulation of the BdG problem associated with Eq. (7) reads:

$$\begin{cases} \int_D \frac{1}{2}\nabla A\cdot\nabla v_1 + \int_D (C_{\text{trap}}-\mu)Av_1 \\ \qquad + \int_D 2\beta|\phi|^2 Av_1 + \int_D \beta\phi^2 Bv_1 = \omega\int_D Av_1, \\ -\int_D \frac{1}{2}\nabla B\cdot\nabla v_2 - \int_D (C_{\text{trap}}+\mu)Bv_2 \\ \qquad - \int_D 2\beta|\phi|^2 Bv_2 - \int_D \beta\overline{\phi}^2 Av_2 = \omega\int_D Bv_2. \end{cases} \tag{28}$$

The bilinear terms in the left hand side of this equation form the finite-element matrix M that is fed to `SLEPc` library. The implementation of the BdG problem of Eq. (28) can be straightforwardly made now in `FreeFEM`:

```
NewMacro problemBdG
    varf vBdGMat([A,B],[v1,v2]) =
    intN(Th,qforder=ord)(.5*grad(v1)'*grad(A) +(
        Ctrap-mu)*A*v1'
    + 2.*beta*abs(phi)^2*A*v1' + beta*phi^2*B*v1'
    - .5*grad(v2)'*grad(B) - (Ctrap-mu)*B*v2'
    - 2.*beta*abs(phi)^2*B*v2' - beta*(phi')^2*A*v2
        ')
    BCBdG;

    varf vBdGVec([A,B],[v1,v2]) = intN(Th,qforder=
        ord)(A*v1' + B*v2');
EndMacro
```

It is easy to see the correspondence between the weak formulation of Eq. (28) and its implementation in the above macro (see the file `BdG_1comp_ddm/A_macro/Macro_problem.edp`). For the computation of the BdG spectrum, we apply a small shift to slightly regularize the eigenproblem, *e.g.* $\sigma = 10^{-4}$ or $\sigma = 10^{-2}$ that is implemented in the `EPSSolve` function of `SLEPc` by using the parameters:
`"-st_type sinvert -eps_target sigma"`.

Upon computing the eigenvalues and eigenvectors of the BdG problem in `SLEPc`, we further check their accuracy by displaying the residual of Eq. (7):

$$\left\| M\begin{pmatrix} A \\ B \end{pmatrix} - \omega\begin{pmatrix} A \\ B \end{pmatrix} \right\|_\infty \tag{29}$$

by using the `SLEPc` parameters: `"-eps_error_backward ::ascii_info_detail"`.

Finally, we present the weak formulation in the two-component case emanating from Eqs. (13)-(14):

$$\begin{cases} \int_D \frac{1}{2}\nabla A\cdot\nabla v_1 + \int_D (C_{\text{trap}}-\mu_1)Av_1 + \int_D \left(2\beta_{11}|\phi_1|^2+\beta_{12}|\phi_2|^2\right)Av_1 \\ \qquad + \int_D \beta_{11}\phi_1^2 Bv_1 + \int_D \beta_{12}\phi_1\overline{\phi_2}Cv_1 + \int_D \beta_{12}\phi_1\phi_2 Dv_1 = \omega\int_D Av_1, \\ -\int_D \frac{1}{2}\nabla B\cdot\nabla v_2 - \int_D (C_{\text{trap}}-\mu)Bv_2 - \int_D \left(2\beta_{11}|\phi_1|^2+\beta_{12}|\phi_2|^2\right)Bv_2 \\ \qquad - \int_D \beta_{11}\overline{\phi_1}^2 Av_2 - \int_D \beta_{12}\overline{\phi_1\phi_2}Cv_2 - \int_D \beta_{12}\overline{\phi_1}\phi_2 Dv_2 = \omega\int_D Bv_2, \\ \int_D \frac{1}{2}\nabla C\cdot\nabla v_3 + \int_D (C_{\text{trap}}-\mu)Cv_3 + \int_D \left(2\beta_{22}|\phi_2|^2+\beta_{21}|\phi_1|^2\right)Cv_3 \\ \qquad + \int_D \beta_{21}\overline{\phi_1}\phi_2 Av_3 + \int_D \beta_{21}\phi_1\phi_2 Bv_3 + \int_D \beta_{22}\phi_2^2 Dv_3 = \omega\int_D Cv_3, \\ -\int_D \frac{1}{2}\nabla D\cdot\nabla v_4 - \int_D (C_{\text{trap}}-\mu)Dv_4 - \int_D \left(2\beta_{22}|\phi_2|^2+\beta_{21}|\phi_1|^2\right)Dv_4 \\ \qquad - \int_D \beta_{21}\overline{\phi_1\phi_2}Av_4 - \int_D \beta_{21}\phi_1\overline{\phi_2}Bv_4 - \int_D \beta_{22}\overline{\phi_2}^2 Cv_4 = \omega\int_D Dv_4. \end{cases} \tag{30}$$

Again, the implementation of the BdG problem of Eq. (30) is easy in `FreeFEM` (see `BdG_2comp_ddm/A_macro/Macro_problem.edp`):

```
NewMacro problemBdG
   varf vBdGMat([A,B,C,D],[v1,v2,v3,v4]) =
      intN(Th,qforder=ord)(.5*grad(v1)'*grad(A) + (
          Ctrap - mu1)*A*v1' + (2.*beta11*un2(phi1,
          phi1) + beta12*un2(phi2,phi2))*A*v1'
        + beta11*phi1*phi1*B*v1' + beta12*phi1*phi2
            '*C*v1' + beta12*phi1*phi2*D*v1'
        -.5*grad(v2)'*grad(B) - (Ctrap - mu1)*B*v2'
            - (2.*beta11*un2(phi1,phi1) + beta12*
            un2(phi2,phi2))*B*v2'
        - beta11*phi1'*phi1'*A*v2' - beta12*phi1'*
            phi2'*C*v2' - beta12*phi1'*phi2*D*v2'
        +.5*grad(v3)'*grad(C) + (Ctrap - mu2)*C*v3'
            + (2.*beta22*un2(phi2,phi2) + beta21*
            un2(phi1,phi1))*C*v3'
        + beta22*phi2*phi2*D*v3' + beta21*phi1'*
            phi2*A*v3' + beta21*phi1*phi2*B*v3'
        -.5*grad(v4)'*grad(D) - (Ctrap - mu2)*D*v4'
            - (2.*beta22*un2(phi2,phi2) + beta21*
            un2(phi1,phi1))*D*v4'
        - beta22*phi2'*phi2'*C*v4' - beta21*phi1'*
            phi2'*A*v4' - beta21*phi1*phi2'*B*v4'
      )
      BCBdG;

   varf vBdGVec([A,B,C,D],[v1,v2,v3,v4]) = intN(Th,
       qforder=ord)(A*v1' + B*v2' + C*v3' + D*v4');
EndMacro
```

5. Validation test cases for the one-component BEC

The first mandatory validation test consists in proving that the new parallel BdG toolbox using `PETSc` provides the correct results for the

cases that were affordable with the previously published sequential BdG toolbox using ARPACK [56]. These cases correspond to fundamental one-component BEC configurations well documented in the physical literature. We set $\beta = 1$ (repulsive interactions) and $C_{\text{trap}} = \frac{1}{2}\omega_\perp^2 r^2$ (isotropic trap, with $r^2 = x^2 + y^2 + z^2$) and compute the following configurations:

- The 2D ground state (parameter file `BdG_1comp_ddm/INIT/2D_ground_state.inc`). The distribution of the BdG modes for oscillations of the ground state in the Thomas-Fermi (TF) limit for repulsive BECs was found in [71], and is given by

$$\omega_{m,k}^{\text{TF}} = \omega_\perp \sqrt{m + 2k^2 + 2k(1+m)}, \tag{31}$$

  where $m, k \geq 0$ are integers. The toolbox computes the first 20 BdG modes for $\mu = 6$ and $\omega_\perp = 0.2$. An important test for this case is to accurately capture the zero mode: using 4 MPI processors, this eigenvalue is computed as $\omega_{0,0}^{\text{TF}} = 6 \cdot 10^{-7} + \mathrm{i}\, 8 \cdot 10^{-14}$. This gives an indication of the precision of calculations with the present toolbox. This case also serves to check that mesh adaptation provides the same results as computations with a refined fixed mesh.
- The 2D dark soliton (parameter file `BdG_1comp_ddm/INIT/2D_dark_soliton.inc`). This case is also known as the dark-soliton stripe (see [72] and references therein). At the linear limit, this state is constructed as

$$\phi_{DS} = \sqrt{\frac{\omega_\perp}{2\pi}} H_0(\sqrt{\omega_\perp}x) H_1(\sqrt{\omega_\perp}y) e^{-\frac{1}{2}\omega_\perp(x^2+y^2)}, \tag{32}$$

  where H_n are Hermite polynomials of degree n. Similarly as before, we set $\omega_\perp = 0.2$, and perform a numerical continuation over μ all the way up to $\mu = 3$ in order to trace the entire branch. This case is computationally interesting, since it exists a preferred direction along which the configuration will tend to align itself. When we adapt the mesh, this direction changes, and the wave function will then rotate. We overcome this issue by allowing the toolbox to perform mesh adaptation at each continuation step. Note that mesh adaptivity is performed at every step in Newton's method, especially when the norm of the correction is greater than 0.1. The results are identical to the results of [73]: we observe the emergence of a cascade of pitchfork (*i.e.* symmetry-breaking) bifurcations.
- The 3D ground state (parameter file `BdG_1comp_ddm/INIT/3D_ground_state.inc`). This is the only 3D case affordable with sequential computations. We compute the BdG spectrum for $\omega_\perp = 1$ without MPI (using the sequential toolbox published in [56]) and with 4 MPI processors (and the μ-adaptivity continuation strategy discussed previously). Both numerical results are in full agreement with the numerical results reported in [22].

These three cases were also considered in our previous contribution [56], where detailed illustrations of the corresponding GP stationary states and associated BdG spectra are provided. The case of a central vortex state (see [56]) is also provided (parameter file `BdG_1comp_ddm/INIT/2D_central_vortex.inc`). The user can obtain the graphs representing stationary states (contours/iso-surfaces of atomic density) and BdG spectra by running the programs of the present toolbox.

To illustrate the capability of the new toolbox to deal with complex 3D cases, we briefly present cases that have been recently considered in the physical literature:

- 3D dark soliton (parameter file `BdG_1comp_ddm/INIT/3D_dark_soliton.inc`). The existence of this state in 3D and its BdG analysis was considered in [74], where the azimuthal symmetry of the state was taken into account to reduce the 3D BdG problem to a 2D one. We perform a full 3D BdG analysis by using an isotropic potential with $\omega_\perp = 1$. The 3D dark soliton (or planar dark soliton) can be constructed in the linear limit by the Cartesian eigenstate $|0,0,1\rangle$ (bearing a zero cut in the z direction), and can be expressed in terms of Hermite polynomials [74] (see also [54]). This state emanates from the linear limit at $\mu = 5/2$, and it is degenerate; the eigenstates $|1,0,0\rangle$ and $|0,1,0\rangle$ produce the same solution although they now have zero cuts along the x and y directions, respectively. The script uses this eigenstate as a seed to Newton's method to perform a continuation over μ. Our 3D results match perfectly the ones obtained in [74] with the axisymmetry hypothesis.
- 3D vortex lines and beyond. We consider two extra cases with complex 3D structure that cannot be computed with the sequential toolbox: a single-charged vortex-line state [22] (parameter file `BdG_1comp_ddm/INIT/3D_vortex_line_single.inc`), and a vortex-ring configuration bearing two (oppositely charged) vortex lines handles [54] (parameter file `BdG_1comp_ddm/INIT/3D_vortex_ring_2vortex_lines.inc`). The former state bifurcates from the linear limit at $\mu = 2.5$ (*i.e.* 1st-excited state), and can be classified in terms of cylindrical coordinates as $|0,1,0\rangle_{\text{cyl}} = r^2 L_0^1(x^2+y^2)e^{i\theta}e^{-(x^2+y^2+z^2)/2}$ [54] (where L_0^1 stands for the associated Laguerre polynomial). Similarly, the vortex-ring with two handles bifurcates at $\mu = 3.5$ from the linear limit, and is constructed by the combination of Hermite polynomials (in Cartesian coordinates) $|2,0,0\rangle + |0,2,0\rangle + \mathrm{i}|1,0,1\rangle$. Our toolbox was capable of tracing branches of solutions for both cases and the respective results are shown in Figs. 1 and 2, respectively. In particular, panels a) in these figures depict the BdG spectra of the pertinent states that match with the findings in [22] and [54]. Panels b) show atomic density isosurfaces $|\phi|^2$ of the solutions for $\mu = 4.5$ and $\mu = 6$, respectively.

To conclude this first part of validation tests and warn the potential user about the necessary computational resources, we present in Table 1 a summary of the considered cases, together with typical computational times and mesh sizes (*i.e.* the number of elements). Table 2 contains the number of unknowns (ndof), the number of tetrahedra (nt), and the number of non-zero elements (nnz) of the matrix used for the computation of the BdG spectra. The toolbox initially builds a mesh by taking into account the topology of the solution. For example, a disk-shaped mesh with smaller triangles and minimum edge size $h_{min} = h_{max}/45$ in its center is used for studying a 2D vortex configuration. The mesh is refined at each iteration in Newton's method in regions of large gradients (*e.g.* around solitons or vortices) and de-refined otherwise (zones of constant density) when adaptive mesh refinement is chosen, see Sec. 3.3. For a given case, we draw comparisons in Table 1 between results that were obtained with 4 MPI processors and without MPI using adaptive mesh refinement. We draw also comparisons in Table 2 for more complex 3D cases. As a general recommendation, we suggest to use adaptive mesh refinement while exploring branches of solutions for which their topology is unknown.

6. Validation test cases for the two-component BEC

We now move on to the study of localized configurations in two-component GP equations (10). Admittedly, the study of their existence, and more crucially, their BdG spectrum (upon solving Eqs. (13)-(14)) places them in an one level harder category. Indeed, the size of the BdG problem for 2D and (even more) 3D configurations becomes quite large, especially when one wants to provide a detailed and accurate description of the spectral properties of such configurations. However, the combination of mesh adaptivity implemented in `FreeFEM` with parallelization tools provided by `PETSc` makes the present toolbox a great candidate to compute such challenging 3D cases. Table 3 summarizes the test cases we considered, and has the same format as Table 2.

Table 1
Results on test cases for the one-component GP and BdG problems with mesh adaptivity. Results are presented with 4 MPI processors and without MPI. The computational time, the mesh size (number of elements) and the number of continuation steps (niter) performed for each case are shown. When using mesh adaptivity, the size of the mesh for the last step of the continuation is depicted in the mesh size column. For 2D cases we compute 100 eigenvalues whereas for the 3D ground state, we compute 40 eigenvalues only. The user can compute more eigenvalues if more memory is available. The BdG spectrum is computed every other two (continuation) steps in μ. The computations were performed on a Macbook pro M1, 16GB of DDR4 2400 MHz RAM.

	Without MPI				4 MPI processors		
	niter	CPU time GP	CPU time BdG	mesh size	CPU time GP	CPU time BdG	mesh size
2D_ground_state	1	00:00:05	00:00:26	10,900	00:00:03	00:00:12	10,866
2D_dark_soliton	208	00:19:12	00:58:18	20,912	00:06:58	00:26:01	19,859
3D_ground_state	133	01:09:16	05:51:27	46,681	00:14:44	04:58:57	47,097

Table 2
Summary of results on 3D test cases for the one-component GP and BdG problems with mesh adaptivity. The number of processors, the mean CPU time per each continuation step, the total number of continuation steps (ncont) performed (for tracing the respective branches) are shown. Moreover, the table contains the number of times the BdG problem was solved (nBdG; we computed the eigenvalues at every 3 continuation steps in μ), the number of unknowns (ndof), the number of tetrahedra (nt), the number of non-zero elements (nnz) of the matrix used for the computation of the BdG spectra, the estimated memory used for each processor maxRSS. For all test cases, 80 eigenvalues were computed in the BdG problem only. Again, the user can compute more eigenvalues if more memory is available. The present computations were performed on the CRIANN Computing Center and MATRICS platform utilizing an Intel Broadwell E5-2680 v4 @ 2.40GHz (14 cores per socket) architecture with two sockets per node and 128 GB of DDR4 2400 MHz RAM. An Intel Omnipath 100Gb/s low latency network was used for communications.

GP test cases	Processors	CPU time	ncont	ndof	nt	maxRSS
3D_dark_soliton	28	00:01:01	168	205,822	76,455	0.78 Gb
3D_vortex_line_single	28	00:04:02	168	329,988	122,969	0.79 Gb
3D_vortex_ring_2vortex_lines	56	00:05:19	201	654,802	244,597	1.10 Gb

BdG test cases	Processors	CPU time	nBdG	ndof	nnz	maxRSS
3D_dark_soliton	28	00:01:30	56	103,116	11,782,505	3.07 Gb
3D_vortex_line_single	28	00:03:41	56	165,362	18,934,023	7.09 Gb
3D_vortex_ring_2vortex_lines	56	00:08:02	67	327,887	37,600,455	8.57 Gb

Table 3
Same as Table 2, but for the two-component GP and BdG problems (again, with mesh adaptivity). Note that 100 and 60 eigenvalues were computed for all 2D and 3D test cases, respectively. The continuations (and thus BdG computations) were performed over β_{12} for the 2D vortex-antidark and 2D ring-antidark states whereas for the rest of the cases, over μ_2.

GP test cases	Processes	CPU time	ncont	ndof	nt	maxRSS
2D_vortex-antidark	4	00:00:17	46	30,506	3,744	1,11 Gb
2D_ring-antidark	4	00:00:09	105	37,847	4,634	4,53 Gb
2D_dark-bright-soliton	4	00:00:12	81	27,775	3,399	0,34 Gb
2D_ground-state-soliton-necklace	4	00:00:29	81	58,081	7,175	2,22 Gb
2D_ground-state-multipole	4	00:00:28	81	41,717	5,131	1,42 Gb
3D_dark-bright-soliton-stripe	28	00:02:18	81	24,698	3,936	1,22 Gb
3D_vortex-ring-bright-state	28	00:03:52	81	24,578	3,935	1,16 Gb

BdG test cases	Processes	CPU time	nBdG	ndof	nnz	maxRSS
2D_vortex-antidark	4	00:00:31	46	27,787	5,125,301	0,78 Gb
2D_ring-antidark	4	00:00:41	91	37,791	6,946,743	1,02 Gb
2D_dark-bright-soliton	4	00:00:33	81	25,303	4,644,479	0,75 Gb
2D_ground-state-soliton-necklace	4	00:01:03	81	55,155	10,136,839	1,43 Gb
2D_ground-state-multipole	4	00:00:37	81	38,882	7,142,816	1,11 Gb
3D_dark-bright-soliton-stripe	28	00:03:56	81	72,202	32,963,342	6,70 Gb
3D_vortex-ring-bright-state	28	00:03:21	81	71,627	32,681,221	7,03 Gb

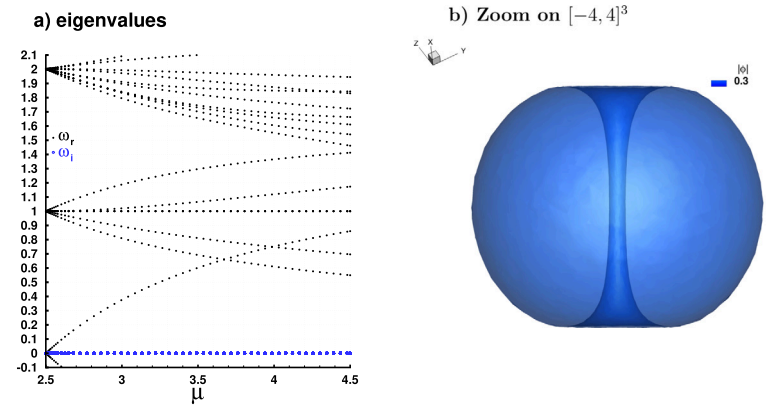

Fig. 1. 3D one-component BEC with single-charged vortex-line configuration. a) The BdG spectrum and b) density $|\phi|^2$ for $\mu = 4.5$. The computational domain is the cube $[-5.4, 5.4]^3$.

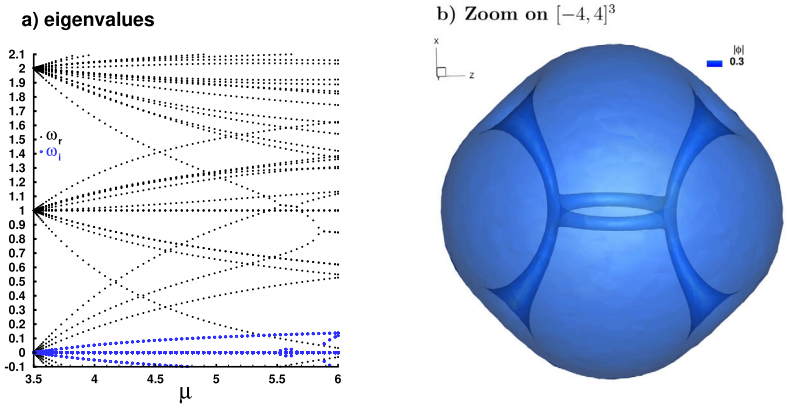

Fig. 2. 3D one-component BEC with vortex-ring with two (oppositely charged) vortex-line handles configuration. a) The BdG spectrum and b) density $|\phi|^2$ for $\mu = 6$. The computational domain is the cube $[-6.23, 6.23]^3$.

6.1. 2D two-component BEC test cases

The first two-component BEC validation cases consider two states taken from [28] and analyzed in detail in [56]: vortex-antidark and dark-antidark ring solutions. The corresponding parameter files are `BdG_2comp_ddm/INIT/2D_vortex-antidark.inc` and `BdG_2comp_ddm/INIT/2D_ring-antidark.inc`, respectively. Such bound modes emerge in two-component GP equations due to the inter-component interaction. Indeed, a dark soliton or a vortex (or a ring) in ϕ_1 will induce an effective potential through the inter-component nonlinearity which itself "traps" a localized mode in ϕ_2. As a consequence, atoms in ϕ_2 "fill-in" the density dip of ϕ_1 through this (effective) trapping process. The scripts compute solutions to the GP system (10) in the case of repulsive inter-component interactions with miscibility condition $0 \le \beta_{12} < \sqrt{\beta_{11}\beta_{22}}$. To simplify the case study, we set $\beta_{11} = \beta_{22} = \beta = 1$, $\beta_{12} = \beta_{21}$, and $0 < \beta_{12} < \beta$, since only the ratio between non-linear interaction coefficients matters. The pictures illustrating these cases (stationary solutions and BdG spectra) are not shown, since identical to those presented in [56].

The last 2D cases consider more exotic states that were recently studied in [34]. Since the construction of the initial condition is crucial in capturing these states, we briefly present the main technical details. At first, we construct the ground state of Eq. (3) by

$$\phi = \sqrt{\frac{\omega_\perp}{2\pi}} H_0(\sqrt{\omega_\perp}x) H_0(\sqrt{\omega_\perp}y) e^{-\frac{1}{2}\omega_\perp(x^2+y^2)}, \tag{33}$$

and use it to seed Newton's method with $\beta = 1.03$ and $\omega_\perp = 0.2$. The branch of the ground state is traced from $\mu \approx 0.202$, *i.e.* close to the linear limit where this state bifurcates from, until $\mu = 1$. For this computation, we use the program for the one-component BEC with parameter file `BdG_1comp_ddm/INIT/2D_Hermite_LL_phi1.inc`.

The terminal profile ϕ, now called ϕ_1, is extracted while setting $\mu_1 = \mu = 1$. Then, we focus on Eq. (10) with $\beta_{11} = 1.03$, $\beta_{22} = 0.97$, and $\beta_{12} = 1$ ($\omega_\perp = 0.2$ and $\mu_1 = 1$ are as before). Following the approach discussed in [25], we plug the (terminal) profile ϕ_1 into the equation for ϕ_2 [cf. Eq. (10)], and linearize it with respect to ϕ_2. This process results in the following eigenvalue problem for (μ_2, ϕ_2)

$$-\frac{1}{2}\nabla^2\phi_2 + C_{\text{eff}}\phi_2 = \mu_2\phi_2 \tag{34}$$

with $C_{\text{eff}} = C_{\text{trap}} + \beta_{21}|\phi_1|^2$ being the effective potential [25] which is responsible for "trapping" bound modes in the ϕ_2 component of Eq. (10). Equation (34) is solved numerically in the script `BdG_2comp_ddm/FFEM_LL_2c_2D_3D_ddm.edp` (using parameter `.inc` files specified below for each case) to obtain eigenvalue-eigenvector pairs (μ_2, ϕ_2), that together with (μ_1, ϕ_1) form the initial guess that we seed to Newton's method. We then trace branches of bound modes of the coupled system of Eq. (10), over the principal continuation parameter μ_2. Note that upon selecting a pair (μ_2, ϕ_2), we perform continuation from μ_2 until $\mu_2 + 0.4$ in all the cases. The toolbox computes the following states that are illustrated in Fig. 3 (atomic density of the state computed for the last value of μ_2) and Fig. 4 (BdG spectra):

- 2D state with a dark-soliton in one component and a bright-soliton in the other one (parameter file `BdG_2comp_ddm/INIT/2D_dark-bright-soliton.inc`). This solution (first column of Fig. 3) corresponds to the first eigenvalue of problem (34) and bifurcates from the linear limit at $\mu_2 \approx 1.05133$. This state is unstable

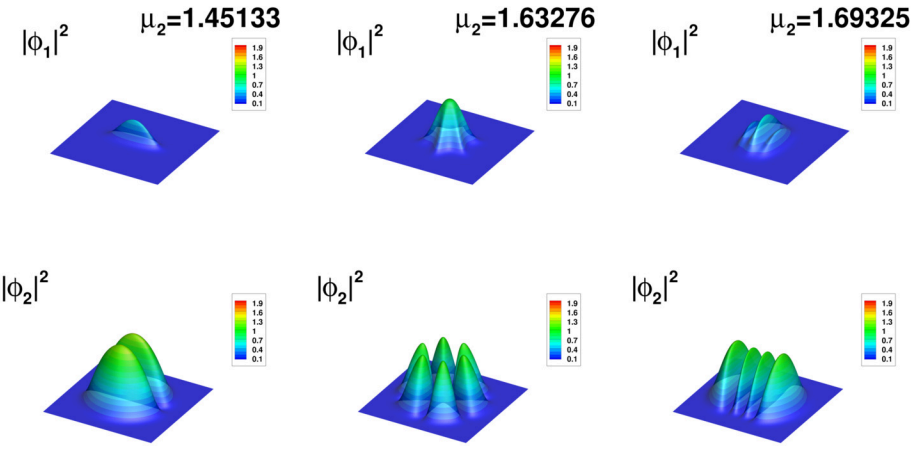

Fig. 3. 2D two-component BEC with soliton and necklace configurations. Density profiles of each of the components, *i.e.* $|\phi_1|^2$ (top row) and $|\phi_2|^2$ (bottom row), for different values of μ_2 (the last value considered in the continuation procedure). From left to right: the dark-bright soliton state, ground state and soliton necklace state, ground state and multipole state (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

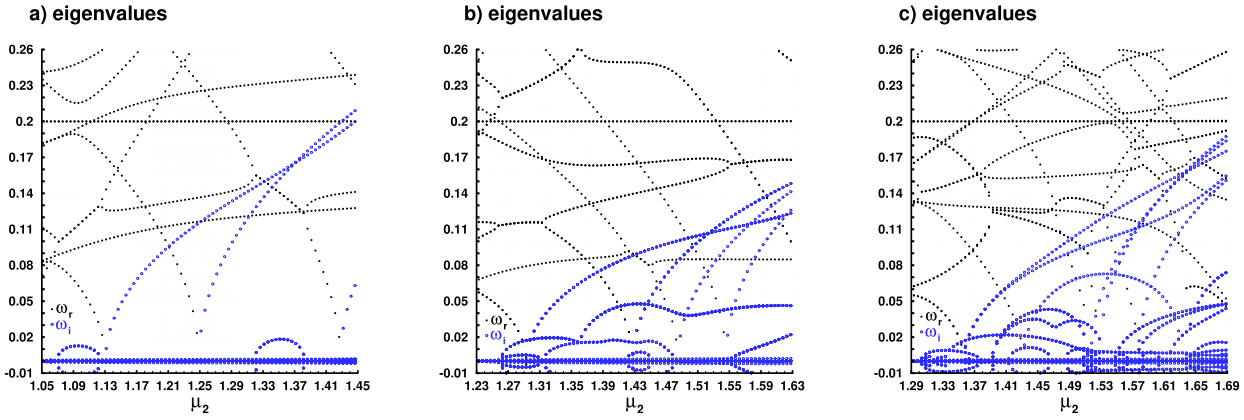

Fig. 4. 2D two-component BEC with soliton and necklace configurations. The BdG spectra as functions of μ_2 corresponding to a) the dark-bright soliton branch, b) ground state (ϕ_1) and soliton necklace (ϕ_2) branch, and c) the ground state and multipole branch. Each of these branches bifurcate from $(\mu_1, \mu_2) \approx (1, 1.05133)$, $(1, 1.23276)$, and $(1, 1.29325)$.

over μ_2 (see Fig. 4a) except from a very narrow window of stability close to the linear limit (see also Fig. 2a in [34], and references therein).

- 2D state with ground-state in one component and soliton necklace in the other one (parameter file `BdG_2comp_ddm/INIT/2D_ground-state-soliton-necklace.inc`). This branch (second column of Fig. 3) corresponds to the 7th eigenvalue of problem (34) and bifurcates from the previous solution at $\mu_2 \approx 1.23276$. It involves a soliton necklace in ϕ_2 (note its imprint on ϕ_1). Similar to the previous case, this state is unstable (see also Fig. 1a in [34]) but features a very narrow window of stability as shown in panel b) of Fig. 4.
- 2D state with ground-state in one component and a multipole in the other one (parameter file `BdG_2comp_ddm/INIT/2D_ground-state-multipole.inc`). This so-called multipole branch (third column of Fig. 3, see also Fig. 16c in [34]) corresponds to the 8th eigenvalue of problem (34) and bifurcates from the linear limit at $\mu_2 = 1.29325$. It can be described by the combination of Cartesian (*i.e.* Hermitian) eigenstates $|2,1\rangle + |0,3\rangle$.

All the cases that we discussed here match perfectly with the numerical results of [34].

6.2. 3D two-component BEC test cases

We present two 3D two-component BEC configurations that are very challenging when computing the BdG spectra. To capture branches of solutions, we employ the same "trapping" technique that was discussed for the 2D two-component cases. The wave function ϕ_1 carries the ground state, and is obtained by running the script for the one-component BEC with $\beta = 1.03$ and $\omega_\perp = 1$ and parameter file `BdG_1comp_ddm/INIT/3D_Hermite_LL_phi1.inc`. Continuation over μ from its linear limit, *i.e.* $\mu \approx 1.501$, is stopped at $\mu = 2$. Then, the eigenvalue problem (34) is solved by the script `BdG_2comp_ddm/FFEM_LL_2c_2D_3D_ddm.edp` (using parameter `.inc` files specified below for each case) to obtain the eigenvalue-eigenvector pairs (μ_2, ϕ_2). This way, and upon selecting an eigenvalue-eigenvector pair of our choice, we trace branches of 3D bound modes of Eq. (10) by performing continuation over μ_2 (while setting $\beta_{11} = 1.03$, $\beta_{22} = 0.97$, $\beta_{12} = 1$) for fixed $\omega_\perp = 1$ and $\mu_1 = 2$. We stop the continuation process when the continuation parameter reaches $\mu_2 + 0.4$, *i.e.* being 0.4 units far away from the respective linear limit of ϕ_2.

With this technique, we can obtain the following 3D states illustrated in Fig. 5 (BdG spectra in the first column and density for the last value of μ_2 in the second column):

- 3D state with ground state in ϕ_1 and the planar dark soliton in ϕ_2 (parameter file `BdG_2comp_ddm/INIT/3D_dark-bright-soliton-stripe.inc`.) This state corresponds to the first eigen-

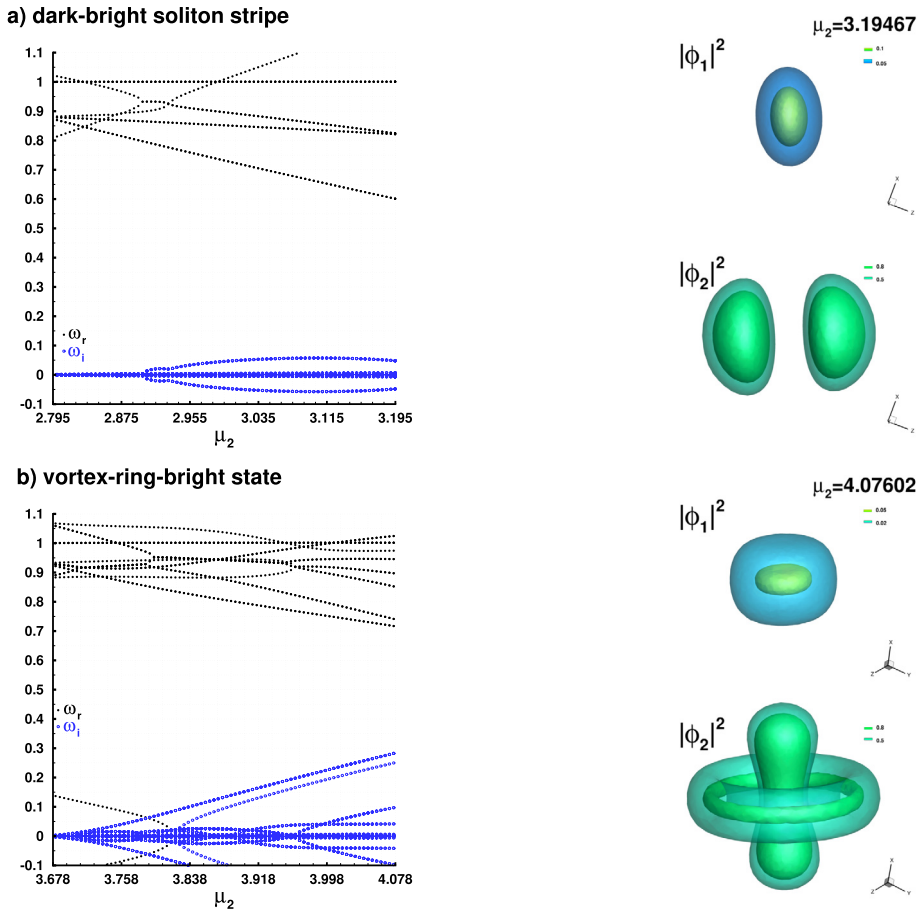

Fig. 5. 3D two-component BEC cases. BdG spectra over μ_2 and atomic density for the two components for the last value of μ_2 considered in the continuation procedure: a) the dark-bright soliton stripe (with the ground state in ϕ_1 and the planar dark soliton in ϕ_2), and b) vortex-ring-bright state (with the vortex-ring state in ϕ_2).

value of (34) and represents a dark-bright soliton stripe in 3D (see Fig. 5a and also [19]. It bifurcates from its linear limit at $\mu_2 \approx 2.79467$. It can be classified in terms of Cartesian eigenfunctions as $|0,1,0\rangle$. Our numerical results on its BdG spectrum show that the state is stable from its inception until $\mu_2 \approx 2.902$ when it becomes unstable.

- 3D state with ground state in ϕ_1 and a vortex-ring state in ϕ_2 (parameter file `BdG_2comp_ddm/INIT/3D_vortex-ring-bright-state.inc`.) This vortex-ring-bright state (see Fig. 5b) was obtained for $\mu_1 = 2$ and corresponds to the 6th eigenvalue of (34). It bifurcates from the linear limit of ϕ_2 at $\mu_2 = 3.67602$ and is generically unstable as shown by its BdG spectrum. Note that in the one-component setting, the vortex-ring state can be classified in terms of a combination of Cartesian eigenstates as $\frac{1}{\sqrt{2}}(|2,0,0\rangle + |0,2,0\rangle) + i|0,0,1\rangle$ [54].

7. Description of the programs

In this section, we first describe the architecture of the programs and the organization of the provided files. We then present the input parameters and the structure of the output files.

7.1. Program architecture

Codes and data files forming the BdG problem with the domain decomposition method (DDM) are stored in the `FFEM_BdG_ddm_tool-box` directory. The latter is organized in two main subdirectories: `BdG_1comp_ddm` and `BdG_2comp_ddm`, corresponding to the one- and two-component codes. Each subdirectory contains two main files: *FFEM_GP_$case_ddm.edp*, which is the main `FreeFEM` script file for the computation of the stationary state, and *FFEM_BdG_$case_ddm.edp* which is the main `FreeFEM` script file for the computation of the BdG eigenvalues ($case = 1c_2D_3D for the one-component case and $case = 2c_2D_3D for the two-component case).

To run these codes, first of all, the user must install `FreeFEM` with `PETSc` following the instructions in https://doc.freefem.org/introduction/installation.html. Then, the user can run the `FreeFEM` code for the computation of the GP stationary state by using either the command

```
mpirun -np 4 FreeFem++-mpi FFEM_GP_$case_ddm.edp
```

or

```
ff-mpirun -np 4 FFEM_GP_$case_ddm.edp.
```

The BdG eigenvalues can then be computed by typing (in terminal) either

```
mpirun -np 4 FreeFem++-mpi FFEM_BdG_$case_ddm.edp
```

or

```
ff-mpirun -np 4 FFEM_BdG_$case_ddm.edp.
```

Parameter files for the examples presented in this paper are stored in the `INIT` folder.

The obtained solutions are saved in the `dircase` directory. Depending on the output format selected by the user, data files are generated in specific folders for visualization with Tecplot,[3] ParaView,[4] and Gnu-

[3] https://tecplot.com.
[4] https://www.paraview.org.

plot.[5] We also provide ready-made layouts for visualization with Tecplot in the folder `Figures`. The user can thus obtain the figures from this paper using newly generated data. More details about the output structure are given in Sect. 7.4.

The complete architecture of the `BdG_1comp_ddm` and `BdG_2comp_ddm` directories is the following:

1. *FFEM_GP_$case_ddm.edp*: the main script for computing GP stationary states.
2. *FFEM_BdG_$case_ddm.edp*: the main script for computing the BdG spectrum.
3. *FFEM_LL_$case_ddm.edp*: the main script for solving the eigenvalue problem of Eq. (34) (*i.e.* for finding the eigenvalue-eigenvector pairs (μ_2, ϕ_2)). This script is used only for the two-component BEC cases.
4. *param_num_common.inc*: a parameter file containing main numerical parameters.
5. `INIT`: directory storing the parameter files for the examples presented in Sects. 5 and 6.
6. `Figures`: directory containing Tecplot layouts used to replot the figures shown in Sects. 5 and 6. The main code must be run with the associated example before opening the layout to replot the figure.
7. `A_macro`: directory containing macros used in the main scripts for GP and BdG problems.
8. `A_macro_LL`: directory containing macros used in the main scripts for the study of respective linear limits (LL).

7.2. Macros and functions

The different macros and functions used in the toolbox for the sequential code are stored in the `A_macro` folders:

- *Macro_BdGsolve.edp*: macro for computing the BdG eigenvalues associated with matrices of Eqs. (28) and (30).
- *Macro_createdir.edp*: macro for creating the file structure of the `dircase` folder.
- *Macro_globalpartition.edp*: macro for creating a partition of the global mesh, and sending the solution from the global mesh to the local one.
- *Macro_GPsolve.edp*: macro for computing the GP stationary state with Newton's method (see Eqs. (19) and (23)-(26)).
- *Macro_LLsolve.edp*: macro for computing the eigenvalues of Eq. (34).
- *Macro_meshAdapt.edp*: macro for adapting the mesh to the wave function.
- *Macro_onedomainsol.edp*: macro for sending the solution from the local domain to the global one.
- *Macro_operator.edp*: collection of useful macros and functions: gradients, energy (4), chemical potential, Hermite polynomials, etc. Also contains a macro that creates a spherical mesh for 3D problems.
- *Macro_output.edp*: macros used for saving data in Tecplot and ParaView formats.
- *Macro_plotEigenvector.edp*: macro for plotting the real and imaginary parts of a BdG eigenvector.
- *Macro_plotphi.edp*: macro for plotting the complex wave function. The user can press the "k" key to alternate between plots of the density, phase and real and imaginary parts of the wave function.
- *Macro_problem.edp*: definitions of the weak formulations for the GP [cf. Eqs. (19) or (30)], the BdG problems [cf. Eqs. (28) or (23)-(26)] and the linear limit problem [cf. Eq. (34)].
- *Macro_readmu.edp*: macro to read the μ from **dircase/Gnuplot/GP_results.dat**, and compute the corresponding BdG eigenvalues.
- *Macro_readmubeta.edp*: macro to read the values of μ or β from **GP_mucont_results.dat** or **GP_betacont_results.dat** that are contained in **dircase/Gnuplot/** in order to compute the corresponding BdG spectrum.
- *Macro_restart.edp*: macros used to save and load the wave function to or from `FreeFEM` files.
- *Macro_saveData.edp*: macro for saving the stationary wave function.
- *Macro_saveEigenvalues.edp*: macro for saving the BdG eigenvalues and eigenvectors.

7.3. Input parameters

Parameters are separated in two files. Numerical parameters used in all computations are specified in *param_num_common.inc*. Files in the `INIT` directory specify physical parameters associated with the state of interest, computation and numerical parameters specific to this problem. The files distributed with the toolbox provide a variety of examples that can be used as a starting point when selecting parameters for the study of new states.

(1) In the file `param_num_common.inc`, the parameters are:

- **displayplot**: controls the output information to plot. Possible values range from 0 (no plots), to 2 (plots data at all iterations of Newton's method, and all eigenvectors computed by the BdG code).
- **iwait**: Boolean indicating if the code must wait for user's input when a plot is shown (`true`) or it can continue (`false`) with the next plot.
- **cutXY**, **cutXZ**, **cutYZ**: (only for 3D cases in the one-component case) Booleans indicating whether to plot cuts of the wave function along the different axis at $x = 0$, $y = 0$ or $z = 0$.
- **Tecplot**: Boolean indicating whether to save data in the Tecplot format.
- **Tecplotddm**: for saving solution for Tecplot with DDM or not.
- **Paraview**: Boolean indicating whether to save data in the ParaView format (only in 2D and 3D).
- **adaptinit**: if `true`, the initial solution is recomputed after the first mesh adaptation.
- **adaptmeshFF**: determines if mesh adaptation is used (`true`) or not (`false`).
- **useShift**: Boolean indicating whether to use a shift when computing the BdG eigenvalues (see, Sec. 4).
- **Nadapt**: if mesh adaptation is used, then the mesh is adapted every **Nadapt** iterations during the continuation.
- **Nplot**: the wave function is plotted every **Nplot** iterations during the continuation.
- **Nsave**: the wave function is saved for ParaView or Tecplot every **Nsave** iterations during the continuation.
- **Nrst**: the wave function is saved for the BdG computation every **Nrst** iterations during the continuation.
- **tolerrF**: the tolerance value ϵ_F in Eq. (27).
- **tolNewton**: the tolerance value ϵ_q in Eq. (27).
- **shift**: the value of the shift σ used when computing eigenvalues.
- **shiftLL**: the value of the shift σ used when computing eigenvalues close to the linear limit.
- **shiftFLL**: the value of the shift σ used when computing eigenvalues far from the linear limit.
- **adaptboundary**: control parameter that determines whether to adapt ($==0$) or not ($==1$) the boundary of the mesh in 3D.
- **skipBdG**: the value to skip μ or β_{12} computed with GP for BdG computation.
- **muL, mubetaL**: to switch between using **shiftFLL** or **shiftLL**, if μ or μ_1 or μ_2 or β_{12} <**mubetaL** we use **shiftLL** otherwise we use **shiftFLL**.
- **LL**: Boolean indicating whether we want to compute the linear limit, *i.e.* eigenvalue problem for (μ_2, ϕ_2) for the second component or not.
- **NNZ**: contain the non zero elements (nnz) for the BdG matrix.
- **dmuk**: counter for dmu adaptation.

[5] http://www.gnuplot.info.

- **FINAL**: Boolean to run the final solution of **endmu** in the GP continuation or to stop the BdG computation.
- **newtonMax**: the maximum number of Newton iterations.

(2) In the file `$case.inc`, stored in the `INIT` directory, the parameters are:

- General parameters for the case:
  - **dimension**: the dimension of the problem (2 or 3).
  - **FEchoice**: the type of finite element used. Usually $P2$.
  - **nev**: the number of eigenvalues computed by the BdG code.
- Parameters used to restart a computation:
  - **restart**: Boolean indicating if the initial solution is a restart from a previous computation. If `true`, the solution and mesh stored in **dirrestart** for the value of μ given by **murestart** will be used as initial solution.
  - **murestart**: the initial value of μ in the case of a restart.
  - **dirrestart**: the folder where the initial solution is stored in the case of a restart.
- Parameters of the continuation:
  - **kpol**, **lpol**, **mpol**: integers defining the initial state in the linear limit.
  - **startmu**: the initial value of μ.
  - **endmu**: the final value of μ.
  - **dmu**: the increment in μ during the continuation.
  - **facmu**: when using the linear limit, the initial value of μ is given by **facmu** $\cdot\, \mu_{|klm\rangle}$. • **mubeta**: a macro that contains the name of the variable that we want to do the continuation over it: μ_1, μ_2, β_{12} or β_{21}.
- Coefficients of the GP equation:
  - **beta**: the nonlinear coefficient (we set $\beta = 1$ in all test cases except for the linear limit cases where $\beta = 1.03$).
  - **ax**, **ay**, **az**: the frequencies of the trapping potential along the three coordinate axes.
  - **Ctrap**: a function defining the trapping potential.
- Parameters for the mesh generation:
  - **Dx**: the distance between points on the mesh border.
  - **scaledom**: a coefficient used to control the size of the domain: the mesh radius is given by **Rdom** = **scaledom** $\cdot\, r_{\mathrm{TF}}$, where r_{TF} is the Thomas-Fermi radius.
  - **createMesh**: a macro creating the initial mesh `Th`.
- Parameters for the mesh adaptation:
  - **errU**: the interpolation error level.
  - **hmin**: the minimum length of a mesh element edge in the new mesh.
  - **hmax**: the maximum length of a mesh element edge in the new mesh.
  - **adaptratio**: the ratio for a prescribed smoothing of the metric. No smoothing is done if the value is less than 1.1.
- Parameters for the initial solution:
  - **initname**: the name given to the initial solution.
  - **initcond**: a macro defining the initial solution for the `phi` variable.
- Definitions of the boundary conditions:
  - **BCGP**: the boundary conditions used in the GP code for Eqs. (19) and (23)-(26).
  - **BCBdG**: the boundary conditions used in the BdG code for Eqs. (28) and (30).
  - **BCLL**: the boundary conditions used in the LL code for Eq. (34).
  - **fcase**: the name given to the current computation.
  - **dircase**: the directory where the results are stored.

(3) In a two component case, some new parameters are defined in the `$case.inc` file:

- Parameters used to restart a computation:
  - **mu1restart**, **mu2restart**: initial values of μ_1 and μ_2 in the case of a restart.
  - **beta12restart**, **beta21restart** initial values of β_{12} and β_{21} in the case of a restart.
- Parameters of the continuation:
  - **startmu1**, **startmu2**: initial values of μ_1 and μ_2.
  - **endmu1**, **endmu2**: final values of μ_1 and μ_2.
  - **dmu1**, **dmu2**: increments of μ_1 and μ_2 during the continuation.
  - **startbeta12**, **startbeta21**: initial values of β_{12} and β_{21}.
  - **endbeta12**, **endbeta21**: final values of β_{12} and β_{21}.
  - **dbeta12**, **dbeta21**: increments of β_{12} and β_{21} during the continuation.
- Coefficients of the GP equation:
  - **beta11**, **beta12**: nonlinear coefficients β_{11} and β_{22}.
- Parameters for the initial solution:
  - **initname1**: the name given to the initial solution for the first component.
  - **initname2**: the name given to the initial solution for the second component.
  - **initcond**: a macro defining the initial solution for `[phi1,phi2]` variables.

7.4. Outputs

When a computation starts, the `OUTPUT_$case` directory is created. It contains up to eight folders. The `RUNPARAM_GP`, `RUNPARAM_BdG`, and `RUNPARAM_LL` directories contain a copy of the code and data files, thus allowing an easy identification of each case, and preparing an eventual rerun of the same case at a later time. The other folders contain different output format files of the computed solution for its visualization using Tecplot, paraView or Gnuplot. The content of these subfolders depends on the case and on the computation parameters (differences in the two component code are given in parentheses):

1. The `Gnuplot` folder contains two files:
   - Information about the stationary states is stored in the `GP_results.dat` file (`GP_mucont_results.dat` or `GP_betacont_results.dat` file). The columns appear in the following order: the non-linear coefficient β (β_{12} and β_{21}), the imposed chemical potential μ (μ_1 and μ_2), the number of Newton iterations used for this value of μ, the norms associated with ϵ_F and ϵ_q in Eq. (27), the computed value of the chemical potential (computed values of μ_1 and μ_2), the number of atoms (5) (the number of atoms in the two components), the GP energy, the mesh size, the number of degrees of freedom, the CPU time to compute the stationary state, and the value of the current $\delta\mu$ ($\delta\mu_1$, $\delta\mu_2$).
   - BdG eigenvalues are stored in the `BdG_results.dat` file. The columns appear in the following order: the non-linear coefficient β (β_{12} and β_{21}), the imposed chemical potential μ (μ_1 and μ_2), the eigenvalue number between 0 and **nev**, the real and imaginary part of the eigenvalues, the Krein signature and its sign (the Krein signature and its sign for the two components).
   - BdG's numerical information is stored in the `BdG_num_results.dat` file. The columns appear in the following order: the non-linear coefficient β (β_{12} and β_{21}), the imposed chemical potential μ (μ_1 and μ_2), the non zero element for the BdG matrix, the number of degrees of freedom, the CPU time to compute the eigenvalues, and the cumulative CPU time.
2. The `Paraview` folder contains the wave functions stored as *.vtk* or *.vtu* and *.pvd* files:
   - `phi_init.vtu` and `phi_final.vtu` are the initial and final solutions.
   - `phi_mu_$mu.vtu` contains the stationary wave function for a given value of μ.
   - `phi_mu1_$mu1_mu2_$mu2.vtu` contains the stationary wave function for given values of μ_1 and μ_2 in the first continuation.

- `phi_beta12_$beta12_beta21_$beta21.vtu` contains the stationary wave function for given values of β_{12} and β_{21} in the second continuation.

3. The `Paraview_Eigenvectors` folder contains the eigenvectors stored as:
   - `eVec_mu_$mu_$nev.vtu` in the one-component code.
   - `eVec_beta12_$beta12_beta21_$beta21_mu1_$mu1_mu2_$nev.vtu` in the two-component code.
4. The `RST` folder contains the stationary states stored as `FreeFEM` files. The names are:
   - `RST-$mu.rst` or `RST-$mu1-$mu2-$beta12-$beta21.rst` for the data.
   - `RSTTh-$mu` or `RSTTh-$mu1-$mu2-$beta12-$beta21` for the mesh files. The file extensions are *.msh* (in 2D) or *.meshb* (in 3D).
5. The `RST_LL` folder contains the stationary states stored as `FreeFEM` files. The names are:
   - `LL_mu1-$mu1_ip-$mu2.rst` for the data.
   - `LLTh_mu1-$mu1_ip-$mu2` for the mesh files. The file extensions are *.msh* (in 2D) or *.meshb* (in 3D).
6. The `Tecplot` folder contains the wave functions stored as *.dat* Tecplot files:
   - `phi_init.dat` and `phi_final.dat` are the initial and final solutions.
   - `phi_mu_$mu.dat` contains the stationary wave function for a given value of μ.
   - `phi_mu1_$mu1_mu2_$mu2.dat` contains the stationary wave function for given values of μ_1 and μ_2 in the first continuation.
   - `phi_beta12_$beta12_beta21_$beta21.dat` contains the stationary wave function for given values of β_{12} and β_{21} in the second continuation.
7. The `Tecplot_Eigenvectors` folder contains the eigenvectors stored in the Tecplot format:
   - `eVec_mu_$mu_$nev.dat` in the one-component code.
   - `eVec_beta12_$beta12_beta21_$beta21_mu1_$mu1_mu2_$nev.dat` in the two-component code.
8. The `Tecplot_Eigenvalues` folder contains the file `BdG_results_eig.dat` with all the eigenvalues stored in the Tecplot format.

8. Summary and conclusions

The experimental realization of single- and two-component BECs in higher spatial dimensions has admittedly been an exciting journey in understanding the fundamental properties of matter at ultracold temperatures. In parallel, however, this journey has posed computational challenges pertaining about not only the existence of matter waves in GP equations (single and two-component versions thereof) but more crucially, their spectral stability analysis, *i.e.* BdG spectrum. The study of the BdG spectrum often results in solving a very large eigenvalue problem, a task that is computationally demanding and requires the use of parallelization. With the present work, we took up this challenge, and presented as well as delivered a parallel finite-element toolbox for computing the BdG spectrum of stationary solutions to one- and two-component GP equations in 2D and 3D.

The toolbox was created with the open-source, finite-element software `FreeFEM` which is now interfaced with parallel libraries such as `PETSc` and `SLEPc`. The ability of `FreeFEM` to perform adaptive mesh refinements, together with the use of parallel linear solvers such as domain decomposition and algebraic multigrid methods in `PETSc`, makes the present toolbox a versatile tool for studying 2D and 3D configurations to GP equations within reasonable CPU times. The computation of the BdG spectrum that is carried out in the present toolbox consists of two steps. At first, stationary states are identified by using Newton's method which now has access to parallel linear solvers from `PETSc`. Moreover, a natural parameter continuation method is adopted to obtain branches of solutions to GP equations over the chemical potential μ or the inter-component interaction parameters β_{12} and β_{21}. Upon tracing branches of solutions, the BdG spectrum is computed afterwards by solving the associated eigenvalue problem with `SLEPc`.

We successfully verified our toolbox's results against known theoretical and numerical findings that have been published in the open literature. We reported typical CPU times that render the toolbox to be used on ordinary laptops and small workstations (of course, depending on the complexity of the state of interest). The parameter files of the toolbox correspond to the test cases we presented in this paper, and they can be used by the user to reproduce the results. We further provide these files from the scope of getting used as templates, if the user intends to compute a new BEC setup or case of interest. We hope that the description and documentation of the toolbox will allow the user in a convenient way to consider other types of trapping potentials *e.g.* quartic ± quadratic trapping ones [75], and nonlinearities, such as the non-local ones appearing in dipolar settings, *e.g.* see [76].

There is clearly a broad array of future computational explorations and developments stemming from this work that we briefly mention here. First, we implemented a natural (or sequential) continuation approach to trace branches of solutions in the present toolbox. It will be quite interesting to consider other types of continuation approaches in `FreeFEM` including the pseudo-arclength continuation [77,78], asymptotic numerical method (ANM) [79], and deflation-based techniques [34,35,54,72], among many others. Another possibility concerns about the interfacing of other libraries for eigenvalue computations, including the FEAST eigenvalue solver [80] which enjoys multiple levels of parallelization [81]. Finally, with the recent experimental developments on spinor condensates [82,83] described by more than two GP equations (see, *e.g.* [33] where the authors considered a three-component GP system for studying monopoles and Alice rings), it is thus timely to bring forth state-of-the-art computing methodologies in order to elucidate the configuration space of solutions in these experimentally accessible systems. Such computational studies and software development in `FreeFEM` are currently in progress and will be reported in future contributions.

CRediT authorship contribution statement

Georges Sadaka: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Resources, Methodology, Investigation, Formal analysis, Data curation. **Pierre Jolivet:** Validation, Software, Resources, Data curation. **Efstathios G. Charalampidis:** Writing – review & editing, Writing – original draft, Validation, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Ionut Danaila:** Writing – original draft, Writing – review & editing, Visualization, Validation, Supervision, Software, Resources, Project administration, Methodology, Funding acquisition, Conceptualization.

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.

Data availability

Data will be made available on request.

Acknowledgements

The authors acknowledge financial support from the French ANR grant ANR-18-CE46-0013 QUTE-HPC. The work of EGC has been partially supported by the U.S. National Science Foundation under Grant No. DMS-2204782. He expresses his gratitude to the Centre national de la recherche scientifique (CNRS) for awarding him a visiting professorship during the summer of 2023 at the Laboratoire de Mathématiques Raphaël Salem (LMRS) at University of Rouen Normandie. Part

of this work used computational resources provided by the Institut du développement et des ressources en informatique scientifique (IDRIS) and Centre Régional Informatique et d'Applications Numériques de Normandie (CRIANN). The authors are grateful to Prof. P. Kevrekidis for stimulating discussions, and Dr. N. Boullé for providing his numerical results that helped for numerical validations in Sec. 6.2. They also thank Dr. V. Kalt for his insights and earlier collaboration.

References

[1] M.H. Anderson, J.R. Ensher, M.R. Matthews, C.E. Wieman, E.A. Cornell, Observation of Bose-Einstein condensation in a dilute atomic vapor, Science 269 (1995) 198–201.
[2] K.B. Davis, M.O. Mewes, M.R. Andrews, N.J. van Druten, D.S. Durfee, D.M. Kurn, W. Ketterle, Bose-Einstein condensation in a gas of sodium atoms, Phys. Rev. Lett. 75 (1995) 3969–3973.
[3] C. Pethick, H. Smith, Bose-Einstein Condensation in Dilute Gases, Cambridge University Press, 2011.
[4] L.P. Pitaevskii, S. Stringari, Bose-Einstein Condensation and Superfluidity, Oxford University Press, 2015.
[5] A.L. Fetter, A.A. Svidzinsky, Vortices in a trapped dilute Bose-Einstein condensate, J. Phys. Condens. Matter 13 (2001) R135.
[6] V.S.P. Engels, I. Coddington, E.A. Cornell, Vortex lattice dynamics in a dilute-gas bec, J. Low Temp. Phys. 134 (2004) 683–688.
[7] P.G. Kevrekidis, R. Carretero-González, D.J. Frantzeskakis, I.G. Kevrekidis, Vortices in Bose-Einstein condensates: some recent developments, Mod. Phys. Lett. B 18 (2004) 1481–1505.
[8] A.L. Fetter, Rotating trapped Bose-Einstein condensates, Rev. Mod. Phys. 81 (2009) 647–691.
[9] M.R. Matthews, B.P. Anderson, P.C. Haljan, D.S. Hall, C.E. Wieman, E.A. Cornell, Vortices in a Bose-Einstein condensate, Phys. Rev. Lett. 83 (1999) 2498–2501.
[10] A.E. Leanhardt, A. Görlitz, A.P. Chikkatur, D. Kielpinski, Y. Shin, D.E. Pritchard, W. Ketterle, Imprinting vortices in a Bose-Einstein condensate using topological phases, Phys. Rev. Lett. 89 (2002) 190403.
[11] C. Becker, S. Stellmer, P. Soltan-Panahi, S. Dörscher, M. Baumert, E.-M. Richter, J. Kronjäger, K. Bongs, K. Sengstock, Oscillations and interactions of dark and dark–bright solitons in Bose-Einstein condensates, Nat. Phys. 4 (2008) 496–501.
[12] K.W. Madison, F. Chevy, W. Wohlleben, J. Dalibard, Vortex formation in a stirred Bose-Einstein condensate, Phys. Rev. Lett. 84 (2000) 806–809.
[13] P.C. Haljan, I. Coddington, P. Engels, E.A. Cornell, Driving Bose-Einstein condensate vorticity with a rotating normal cloud, Phys. Rev. Lett. 87 (2001) 210403.
[14] D. Yan, J.J. Chang, C. Hamner, P.G. Kevrekidis, P. Engels, V. Achilleos, D.J. Frantzeskakis, R. Carretero-González, P. Schmelcher, Multiple dark-bright solitons in atomic Bose-Einstein condensates, Phys. Rev. A 84 (2011) 053630.
[15] G. Theocharis, A. Weller, J.P. Ronzheimer, C. Gross, M.K. Oberthaler, P.G. Kevrekidis, D.J. Frantzeskakis, Multiple atomic dark solitons in cigar-shaped Bose-Einstein condensates, Phys. Rev. A 81 (2010) 063604.
[16] D.R. Scherer, C.N. Weiler, T.W. Neely, B.P. Anderson, Vortex formation by merging of multiple trapped Bose-Einstein condensates, Phys. Rev. Lett. 98 (2007) 110402.
[17] C.N. Weiler, T.W. Neely, D.R. Scherer, A.S. Bradley, M.J. Davis, B.P. Anderson, Spontaneous vortices in the formation of Bose-Einstein condensates, Nature 455 (2008) 948–951.
[18] A. Aftalion, I. Danaila, Three-dimensional vortex configurations in a rotating Bose-Einstein condensate, Phys. Rev. A 68 (2003) 023603.
[19] D.J.F.P.G. Kevrekidis, R. Carretero-González, The Defocusing Nonlinear Schrödinger Equation: From Dark Solitons to Vortices and Vortex Rings, Society for Industrial and Applied Mathematics, Philadelphia, 2015.
[20] B.A. Malomed, (invited) vortex solitons: old results and new perspectives, Phys. D, Nonlinear Phenom. 399 (2019) 108–137.
[21] L.-C. Crasovan, V.M. Pérez-Garcìa, I. Danaila, D. Mihalache, L. Torner, Three–dimensional parallel vortex rings in Bose–Einstein condensates, Phys. Rev. A 70 (2004), 033605(1–5).
[22] R.N. Bisset, W. Wang, C. Ticknor, R. Carretero-González, D.J. Frantzeskakis, L.A. Collins, P.G. Kevrekidis, Robust vortex lines, vortex rings, and hopfions in three-dimensional Bose-Einstein condensates, Phys. Rev. A 92 (2015) 063611.
[23] W. Wang, R.N. Bisset, C. Ticknor, R. Carretero-González, D.J. Frantzeskakis, L.A. Collins, P.G. Kevrekidis, Single and multiple vortex rings in three-dimensional Bose-Einstein condensates: existence, stability, and dynamics, Phys. Rev. A 95 (2017) 043638.
[24] V. Kalt, G. Sadaka, I. Danaila, F. Hecht, Identification of vortices in quantum fluids: finite element algorithms and programs, Comput. Phys. Commun. 284 (2023) 108606.
[25] E.G. Charalampidis, P.G. Kevrekidis, D.J. Frantzeskakis, B.A. Malomed, Dark-bright solitons in coupled nonlinear Schrödinger equations with unequal dispersion coefficients, Phys. Rev. E 91 (2015) 012924.
[26] K.J.H. Law, P.G. Kevrekidis, L.S. Tuckerman, Stable vortex–bright-soliton structures in two-component Bose-Einstein condensates, Phys. Rev. Lett. 105 (2010) 160405.
[27] E.G. Charalampidis, P.G. Kevrekidis, D.J. Frantzeskakis, B.A. Malomed, Vortex-soliton complexes in coupled nonlinear Schrödinger equations with unequal dispersion coefficients, Phys. Rev. E 94 (2016) 022207.
[28] I. Danaila, M.A. Khamehchi, V. Gokhroo, P. Engels, P.G. Kevrekidis, Vector dark-antidark solitary waves in multicomponent Bose-Einstein condensates, Phys. Rev. A 94 (2016) 053617.
[29] W. Wang, P.G. Kevrekidis, Two-component dark-bright solitons in three-dimensional atomic Bose-Einstein condensates, Phys. Rev. E 95 (2017) 032201.
[30] J. Ruostekoski, J.R. Anglin, Creating vortex rings and three-dimensional skyrmions in Bose-Einstein condensates, Phys. Rev. Lett. 86 (2001) 3934–3937.
[31] R.A. Battye, N.R. Cooper, P.M. Sutcliffe, Stable skyrmions in two-component Bose-Einstein condensates, Phys. Rev. Lett. 88 (2002) 080401.
[32] J. Ruostekoski, J.R. Anglin, Monopole core instability and Alice rings in spinor Bose-Einstein condensates, Phys. Rev. Lett. 91 (2003) 190402.
[33] T. Mithun, R. Carretero-González, E.G. Charalampidis, D.S. Hall, P.G. Kevrekidis, Existence, stability, and dynamics of monopole and Alice ring solutions in antiferromagnetic spinor condensates, Phys. Rev. A 105 (2022) 053303.
[34] E. Charalampidis, N. Boullé, P. Farrell, P. Kevrekidis, Bifurcation analysis of stationary solutions of two-dimensional coupled Gross-Pitaevskii equations using deflated continuation, Commun. Nonlinear Sci. Numer. Simul. 87 (2020) 105255.
[35] N. Boullé, I. Newell, P.E. Farrell, P.G. Kevrekidis, Two-component three-dimensional atomic Bose-Einstein condensates supporting complex stable patterns, Phys. Rev. A 107 (2023) 012813.
[36] T. Kapitula, K. Promislow, Spectral and Dynamical Stability of Nonlinear Waves, Springer, 2013.
[37] N.N. Bogolyubov, On the theory of superfluidity, J. Phys. (USSR) 11 (1947) 23–32.
[38] P.G. De Gennes, Superconductivity of Metals and Alloys, CRC Press, 1966.
[39] C.M. Dion, E. Cancès, Ground state of the time-independent Gross-Pitaevskii equation, Comput. Phys. Commun. 177 (2007) 787–798.
[40] M. Caliari, S. Rainer, GSGPEs: a Matlab code for computing the ground state of systems of Gross-Pitaevskii equations, Comput. Phys. Commun. 184 (2013) 812–823.
[41] X. Antoine, R. Duboscq, GPELab, a Matlab toolbox to solve Gross-Pitaevskii equations I: computation of stationary solutions, Comput. Phys. Commun. 185 (2014) 2969–2991.
[42] Z. Marojević, E. Göklü, C. Lämmerzahl, ATUS-PRO: a FEM-based solver for the time-dependent and stationary Gross-Pitaevskii equation, Comput. Phys. Commun. 202 (2016) 216–232.
[43] G. Vergez, I. Danaila, S. Auliac, F. Hecht, A finite-element toolbox for the stationary Gross–Pitaevskii equation with rotation, Comput. Phys. Commun. 209 (2016) 144–162.
[44] H. Uecker, D. Wetzel, J. Rademacher, pde2path - a Matlab package for continuation and bifurcation in 2D elliptic systems, Numer. Math., Theory Methods Appl. 7 (2014) 58–106.
[45] P. Muruganandam, S. Adhikari, Fortran programs for the time-dependent Gross-Pitaevskii equation in a fully anisotropic trap, Comput. Phys. Commun. 180 (2009) 1888–1912.
[46] D. Vudragović, I. Vidanović, A. Balaz, P. Muruganandam, S.K. Adhikari, C programs for solving the time-dependent Gross-Pitaevskii equation in a fully anisotropic trap, Comput. Phys. Commun. 183 (2012) 2021–2025.
[47] L. Kong, J. Hong, J. Zhang, LOD-ms for Gross-Pitaevskii equation in Bose-Einstein condensates, Commun. Comput. Phys. 14 (2014) 219–241.
[48] U. Hohenester, OCTBEC a Matlab toolbox for optimal quantum control of Bose-Einstein condensates, Comput. Phys. Commun. 185 (2014) 194–216.
[49] R. Kishor Kumar, V. Lončar, P. Muruganandam, S.K. Adhikari, A. Balaž, C and Fortran OpenMP programs for rotating Bose-Einstein condensates, Comput. Phys. Commun. 240 (2019) 74–82.
[50] W. Bao, Q. Du, Computing the ground state solution of Bose-Einstein condensates by a normalized gradient flow, SIAM J. Sci. Comput. 25 (2004) 1674.
[51] E.L. Allgower, K. Georg, Spectral and Dynamical Stability of Nonlinear Waves, Springer-Verlag, 1990.
[52] C.T. Kelley, Solving Nonlinear Equations with Newton's Method, Society of Industrial and Applied Mathematics, Philadelphia, 2003.
[53] R. Carretero-González, P. Kevrekidis, T. Kolokolnikov, Vortex nucleation in a dissipative variant of the nonlinear Schrödinger equation under rotation, Phys. D, Nonlinear Phenom. 317 (2016) 1–14.
[54] N. Boullé, E.G. Charalampidis, P.E. Farrell, P.G. Kevrekidis, Deflation-based identification of nonlinear excitations of the three-dimensional Gross-Pitaevskii equation, Phys. Rev. A 102 (2020) 053307.
[55] A. Roy, S. Pal, S. Gautam, D. Angom, P. Muruganandam, FACt: FORTRAN toolbox for calculating fluctuations in atomic condensates, Comput. Phys. Commun. 256 (2020) 107288.
[56] G. Sadaka, V. Kalt, F. Hecht, I. Danaila, A finite element toolbox for the Bogoliubov-de gennes stability analysis of Bose-Einstein condensates, Comput. Phys. Commun. 294 (2024) 108948.
[57] F. Hecht, New developments in Freefem++, J. Numer. Math. 20 (2012) 251–266.
[58] S. Balay, et al., PETSc/TAO users manual, Argonne National Laboratory-21/39, https://petsc.org/release/docs/manual/manual.pdf, 2022.
[59] P. Jolivet, Introduction to FreeFEM with an emphasis on parallel computing, https://joliv.et/FreeFem-tutorial/main.pdf, 2023.
[60] V. Dolean, P. Jolivet, F. Nataf, An Introduction to Domain Decomposition Methods: Algorithms, Theory, and Parallel Implementation, Society for Industrial and Applied Mathematics, Philadelphia, 2016.
[61] P.-H. Tournier, P. Jolivet, F. Nataf, FFDDM: FreeFem domain decomposition method, https://doc.freefem.org/documentation/ffddm/index.html, 2019.

[62] V. Hernandez, J.E. Roman, V. Vidal, SLEPc: a scalable and flexible toolkit for the solution of eigenvalue problems, ACM Trans. Math. Softw. 31 (2005) 351–362.
[63] Y. Gao, Y. Cai, Numerical methods for Bogoliubov-de Gennes excitations of Bose-Einstein condensates, J. Comput. Phys. 403 (2020) 109058.
[64] W. Bao, Y. Cai, Mathematical models and numerical methods for spinor Bose-Einstein condensates, Commun. Comput. Phys. 24 (2018) 899–965.
[65] R.A. Adams, J.J.F. Fournier, Sobolev Spaces, Academic Press, 2003.
[66] H. Borouchaki, M.J. Castro-Diaz, P.L. George, F. Hecht, B. Mohammadi, Anisotropic adaptive mesh generation in two dimensions for CFD, in: 5th Inter. Conf. on Numerical Grid Generation in Computational Field Simulations, Mississipi State Univ., 1996.
[67] P.J. Frey, P.L. George, Maillages, Hermès, Paris, 1999.
[68] B. Mohammadi, O. Pironneau, Applied Shape Design for Fluids, Oxford Univ. Press, 2000.
[69] C. Dapogny, C. Dobrzynski, P. Frey, Three-dimensional adaptive domain remeshing, implicit domain meshing, and applications to free and moving boundary problems, J. Comput. Phys. 262 (2014) 358–378.
[70] I. Danaila, F. Hecht, A finite element method with mesh adaptivity for computing vortex states in fast-rotating Bose-Einstein condensates, J. Comput. Phys. 229 (2010) 6946–6960.
[71] P.G. Kevrekidis, D.E. Pelinovsky, Distribution of eigenfrequencies for oscillations of the ground state in the Thomas-Fermi limit, Phys. Rev. A 81 (2010) 023627.
[72] E. Charalampidis, P. Kevrekidis, P. Farrell, Computing stationary solutions of the two-dimensional Gross–Pitaevskii equation with deflated continuation, Commun. Nonlinear Sci. Numer. Simul. 54 (2018) 482–499.
[73] S. Middelkamp, P.G. Kevrekidis, D.J. Frantzeskakis, R. Carretero-González, P. Schmelcher, Bifurcations, stability, and dynamics of multiple matter-wave vortex states, Phys. Rev. A 82 (2010) 013646.
[74] R.N. Bisset, W. Wang, C. Ticknor, R. Carretero-González, D.J. Frantzeskakis, L.A. Collins, P.G. Kevrekidis, Bifurcation and stability of single and multiple vortex rings in three-dimensional Bose-Einstein condensates, Phys. Rev. A 92 (2015) 043601.
[75] V. Bretin, S. Stock, Y. Seurin, J. Dalibard, Fast rotation of a Bose-Einstein condensate, Phys. Rev. Lett. 92 (2004) 050403.
[76] Q. Tang, M. Xie, Y. Zhang, Y. Zhang, A spectrally accurate numerical method for computing the Bogoliubov-de Gennes excitations of dipolar Bose-Einstein condensates, SIAM J. Sci. Comput. 44 (2022) B100–B121.
[77] E. Doedel, L.S. Tuckerman, Numerical Methods for Bifurcation Problems and Large-Scale Dynamical Systems, The IMA Volumes in Mathematics and Its Applications, Springer, 2000.
[78] R. Seydel, Practical Bifurcation and Stability Analysis, Interdisciplinary Applied Mathematics, vol. 5, Springer-Verlag, New York, 2010.
[79] P. Ventura, M.P. Ferry, H. Zahrouni, A secure version of asymptotic numerical method via convergence acceleration, C. R., Méc. 348 (2020) 361–374.
[80] E. Polizzi, Density-matrix-based algorithm for solving eigenvalue problems, Phys. Rev. B 79 (2009) 115112.
[81] E. Polizzi, FEAST eigenvalue solver, https://www.feast-solver.org/index.htm, 2023.
[82] Y. Xiao, M.O. Borgh, L.S. Weiss, A.A. Blinova, J. Ruostekoski, D.S. Hall, Controlled creation and decay of singly-quantized vortices in a polar magnetic phase, Commun. Phys. 4 (2021).
[83] Y. Xiao, M.O. Borgh, A.A. Blinova, T. Ollikainen, J. Ruostekoski, D.S. Hall, Topological superfluid defects with discrete point group symmetries, Nat. Commun. 13 (2022).