



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

Georges Sadaka, Pierre Jolivet, Efstathios Charalampidis, Ionut Danaila

► To cite this version:

Georges Sadaka, Pierre Jolivet, Efstathios Charalampidis, Ionut Danaila. Parallel finite-element codes for the Bogoliubov-de Gennes stability analysis of Bose-Einstein condensates. 2024. hal-04528301

HAL Id: hal-04528301

<https://hal.science/hal-04528301>

Preprint submitted on 1 Apr 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

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

Georges Sadaka¹, Pierre Jolivet², Efsthios G. Charalampidis^{1,3} and Ionut Danaila^{1,*}

¹Univ Rouen Normandie, CNRS,
Laboratoire de Mathématiques Raphaël Salem,
UMR 6085, F-76000 Rouen, France

²Sorbonne Université, CNRS, LIP6,
4 place Jussieu, 75252 Paris, France.

³ Mathematics Department, California Polytechnic State University,
San Luis Obispo, CA 93407-0403, USA

* Corresponding author: ionut.danaila@univ-rouen.fr

March 26, 2024

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.

Programm summary

Program Title: FFEM_BdG_ddm_toolbox.zip

CPC Library link to program files:

Developer's repository link:

Code Ocean capsule: Licensing provisions: GPLv3

Programming language: FreeFem++ (v 4.12) free software (www.freefem.org)

Catalogue identifier:

Program summary URL:

Program obtainable from:

No. of lines in distributed program, including test data, etc.: 4,054

No. of bytes in distributed program, including test data, etc.: 200Ko

Distribution format: .zip

Computer: PC, Mac, Super-computer.

Operating system: Mac OS, Linux, Windows.

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 PETSc¹ 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 out by accurately solving, at each point in the parameter space, the underlying eigenvalue problem by using the SLEPc² 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<https://petsc.org/>

²<https://slepc.upv.es>

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 ([Anderson et al., 1995](#); [Davis et al., 1995](#)). Both theoretical and experimental developments have been summarized in ([Pethick and Smith, 2011](#); [Pitaevskii and Stringari, 2015](#)). These studies revealed the emergence of interesting wave configurations including vortices and vortex structures ([Fetter and Svidzinsky, 2001](#); [P. Engels and Cornell, 2004](#); [Kevrekidis et al., 2004](#); [Fetter, 2009](#)), 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 ([Matthews et al., 1999](#); [Leanhardt et al., 2002](#); [Becker et al., 2008](#)), stirring the condensate above a certain critical angular speed ([Madison et al., 2000](#); [Haljan et al., 2001](#)), counterflow techniques ([Yan et al., 2011](#)), the use of anisotropic potentials ([Theocharis et al., 2010](#)), nonlinear interference between different condensate fragments ([Scherer et al., 2007](#)) as well as the so-called Kibble-Zurek mechanism ([Weiler et al., 2008](#)), 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) ([Aftalion and Danaila, 2003](#)), as well as vortex rings ([P. G. Kevrekidis and Carretero-González, 2015](#)) (see also the review ([Malomed, 2019](#)) 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 ([Crasovan et al., 2004](#); [Bisset et al., 2015](#); [Wang et al., 2017](#))), together with recent computational techniques for vortex identification [V. Kalt and Danaila \(2023\)](#). Alongside prototypical bound modes in multi-component BECs that can be identified (*e.g.* dark-bright ([Charalampidis et al., 2015](#)), vortex-bright ([Law et al., 2010](#); [Charalampidis et al., 2016](#)) and dark-antidark ([Danaila et al., 2016](#)), as well as vortex-ring-bright and vortex-line-bright solitons ([Wang and Kevrekidis, 2017](#))), more exotic configurations have been reported, including skyrmions ([Ruostekoski and Anglin, 2001](#); [Battye et al., 2002](#)), monopoles and Alice rings ([Ruostekoski and Anglin, 2003](#); [Mithun et al., 2022](#)). 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 ([Charalampidis et al., 2020](#); [Boullé et al., 2023](#)).

The principal model for the above theoretical and computational studies has been the Gross-Pitaevskii (GP) equation ([Pitaevskii and Stringari, 2015](#)) (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 ([Pitaevskii and Stringari, 2015](#)). 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 ([Kapitula and Promislow, 2013](#)) 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 (Bogolyubov, 1947; De Gennes, 1966). 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 (Dion and Cancès, 2007; Caliari and Rainer, 2013; Antoine and Duboscq, 2014), finite elements (Marojević et al., 2016; Vergez et al., 2016) (see, also (Uecker et al., 2014) which includes a working example on GP equations), and finite differences (Muruganandam and Adhikari, 2009; Vudragović et al., 2012; Kong et al., 2014; Hohenester, 2014; Kishor Kumar et al., 2019). In almost all of these studies, stationary solutions are computed when the L2-norm of the wave function is fixed (this is accomplished by imposing a Lagrange multiplier constraint for the number of atoms). Alternative approaches for the computation of solutions to the GP equation involves 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 (Allgower and Georg, 1990) coupled with Newton’s method (Kelley, 2003). This approach has been adopted in a series of studies that employ finite-element (Danaila et al., 2016; Carretero-González et al., 2016; Boullé et al., 2020; Charalampidis et al., 2020), finite-difference, (Bisset et al., 2015; Wang et al., 2017; Charalampidis et al., 2020), as well as spectral (spatial) discretization methods (Wang et al., 2017). 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 (Roy et al., 2020) 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 (SADAKA et al., 2024). It utilizes mesh adaptation techniques (that are built-in in FreeFem++) and employs the ARPACK eigenvalue solver (Hecht, 2012) (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 (SADAKA et al., 2024), 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 Balay and et. al. (2022) (see also Jolivet (2023)). 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 Dolean et al. (2016); Tournier et al. (2019)) 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 ([Hernandez et al., 2005](#)) 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 existence and 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.

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 adimensionalized form, and further details about the physical units of the model equations together with their scaling can be found in our recent contribution [SADAKA et al. \(2024\)](#) (and references therein). For the user's convenience, we include with this toolbox the 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 a complex-valued wave function $\psi(\mathbf{x}, t) : \mathcal{D} \times \mathbb{R}^+ \rightarrow \mathbb{C}$, where $\mathcal{D} \in \mathbb{R}^d$ and d is the spatial dimension, *i. e.* $d = 1, 2, 3$. In the mean-field approximation for the interparticle interactions in a BEC,

the wave function $\psi(\mathbf{x}, t)$ is a solution of the Gross-Pitaevskii (GP) equation [Pitaevskii and Stringari \(2015\)](#):

$$i\frac{\partial\psi}{\partial t} = -\frac{1}{2}\nabla^2\psi + C_{\text{trap}}\psi + \beta|\psi|^2\psi, \quad (1)$$

where ∇^2 stands for the Laplacian, β is the nonlinearity strength corresponding to repulsive ($\beta > 0$) or attractive ($\beta < 0$) interactions and i is the imaginary unit ($i^2 = -1$). The external potential $C_{\text{trap}}(\mathbf{x})$ confining the atoms in the condensate is taken to be the harmonic oscillator potential

$$C_{\text{trap}}(x, y, z) = \frac{1}{2}(\omega_x^2 x^2 + \omega_y^2 y^2 + \omega_z^2 z^2), \quad (2)$$

where $\omega_x, \omega_y, \omega_z$ are the trapping frequencies. Note that the interplay of different aspect ratios of the trapping frequencies can lead to different BEC scenarios [P. G. Kevrekidis and Carretero-González \(2015\)](#). For example, the case $\omega_x \equiv \omega_y \gg \omega_z$ leads to 1D BEC configurations (ideal to study bright/dark solitons), whereas $\omega_x \equiv \omega_y \ll \omega_z$ leads to 2D configurations (*e.g.* ideal to study vortices). For 3D configurations (*e.g.* where vortex lines and vortex rings could be observed), usually $\omega_x \equiv \omega_y \sim \omega_z$ (the interested reader may also want to see the so-called dimension-reduction approach in [\(Frantzeskakis, 2010; Bao and Cai, 2013\)](#)).

Stationary solutions to Eq. (1) are sought by using the standing wave ansatz

$$\psi(\mathbf{x}, t) = \phi(\mathbf{x})e^{-i\mu t}, \quad (3)$$

where μ is the chemical potential. The stationary GP equation is then obtained:

$$-\frac{1}{2}\nabla^2\phi + C_{\text{trap}}\phi + \beta|\phi|^2 = \mu\phi. \quad (4)$$

The presence of the external potential C_{trap} in (4) makes the atomic density $n(\mathbf{x}) = |\phi(\mathbf{x})|^2$ vanish rapidly outside the condensate. This necessitates the use of homogeneous Dirichlet boundary conditions for the stationary wave function, *i.e.* we impose $\phi = 0$ on $\partial\mathcal{D}$.

In this work, we compute stationary solutions for fixed μ . Branches of such solutions are obtained by performing numerical continuation [Allgower and Georg \(1990\)](#) over μ , which corresponds here to a bifurcation parameter. At each step in the continuation process, we monitor the energy

$$\mathcal{E}(\phi) = \int_{\mathcal{D}} \left(\frac{1}{2}|\nabla\phi(\mathbf{x})|^2 + V_{\text{trap}}(\mathbf{x})|\phi(\mathbf{x})|^2 + \frac{\beta}{2}|\phi(\mathbf{x})|^4 \right) d\mathbf{x}, \quad (5)$$

and the L2-norm of the solution ϕ

$$N(\phi) = \int_{\mathcal{D}} \phi\bar{\phi} d\mathbf{x} = \int_{\mathcal{D}} |\phi|^2 d\mathbf{x}, \quad (6)$$

the latter representing the total number of atoms in the condensate (the overbar stands for complex conjugation). Note that both the energy and number of atoms are conserved quantities for the time-dependent GP equation (1).

We proceed with the setup of the spectral stability analysis problem. This is obtained by linearizing Eq. (1), and this process gives the Bogoliubov-de Gennes (BdG) problem we aim to solve. We first consider the ansatz

$$\psi(\mathbf{x}, t) = \left[\phi(\mathbf{x}) + \delta \left(A(\mathbf{x})e^{-i\omega t} + \overline{B}(\mathbf{x})e^{i\overline{\omega}t} \right) \right] e^{-i\mu t}, \quad \delta \ll 1, \quad (7)$$

where $\phi(\mathbf{x})$ is a stationary state (that we linearize the GP equation about), A and B are complex-valued functions, and ω is a complex number. Upon inserting (7) into (1), we obtain at order $\mathcal{O}(\delta)$ the linear eigenvalue problem called the BdG equation:

$$\begin{pmatrix} \mathcal{H} - \mu + 2\beta|\phi|^2 & \beta\phi^2 \\ -\beta\overline{\phi}^2 & -(\mathcal{H} - \mu + 2\beta|\phi|^2) \end{pmatrix} \begin{pmatrix} A \\ B \end{pmatrix} = \omega \begin{pmatrix} A \\ B \end{pmatrix}, \quad (8)$$

where

$$\mathcal{H} \equiv -\frac{1}{2}\nabla^2 + C_{\text{trap}}. \quad (9)$$

The present toolbox computes the eigenvalue-eigenvector pair (ω, A, B) for a given stationary solution $\phi \in \mathbb{C}$. Note that A and B represent the components of the eigenvector (A, B) in Eq. (8). We direct the reader to [SADAKA et al. \(2024\)](#) for a detailed discussion on the properties of the BdG problem mentioned above. We recall the definition of the Krein signature

$$K = \text{sign}(\omega \int_{\mathcal{D}} (|A|^2 - |B|^2) d\mathbf{x}). \quad (10)$$

If $\int_{\mathcal{D}} (|A|^2 - |B|^2) d\mathbf{x} \neq 0$, only real eigenvalues ω are possible and K becomes an important diagnostic tool assessing on the energetic stability of a solution ϕ : if $K > 0$ for all modes, then ϕ is the global minimum of the energy, *i. e.* the ground state; if there exists a mode with $K < 0$, then the excitation reduces the energy of the system and the stationary state is thus energetically unstable, *i. e.* excited state (or local minimum of the energy). If $\int_{\mathcal{D}} (|A|^2 - |B|^2) d\mathbf{x} = 0$, complex eigenvalues $\omega = \omega_r + i\omega_i$ are possible; moreover, if $\omega_i \neq 0$, then the BdG mode is dynamically unstable.

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

In the mean-field approximation, a mixture of two BECs (*e. g.* different hyperfine states of the same species) is described by a coupled system of GP equations [Pethick and Smith \(2011\)](#); [Pitaevskii and Stringari \(2015\)](#). We consider here two wave functions ψ_1 and ψ_2 accounting

for the two components and satisfying the dimensionless coupled system of GP equations [P. G. Kevrekidis and Carretero-González \(2015\)](#):

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

Coefficients β_{11} and β_{22} in (11) represent the interaction strengths between atoms of same species (or spin states) whereas the β_{12} and β_{21} represent the ones between different species. Note that Eqs. (11) consider (for the sake of simplicity) the same trapping potential C_{trap} (given by Eq. (2)) across the two components. If necessary, the user of the present toolbox can easily implement different potentials in the provided scripts (see the next section).

Similarly to the one-component case, we compute stationary states to Eqs. (11) by using the separation of variables Ansätze

$$\psi_j(\mathbf{x}, t) = \phi_j(\mathbf{x})e^{-i\mu_j t}, \quad j = 1, 2, \quad (12)$$

with chemical potentials μ_1 and μ_2 . Indeed, if we plug (12) into (11), we obtain the following coupled system of (stationary) GP equations

$$\begin{cases} \mu_1\phi_1 = \left(-\frac{1}{2}\nabla^2 + C_{\text{trap}} + \beta_{11}|\phi_1|^2 + \beta_{12}|\phi_2|^2\right)\phi_1, \\ \mu_2\phi_2 = \left(-\frac{1}{2}\nabla^2 + C_{\text{trap}} + \beta_{21}|\phi_1|^2 + \beta_{22}|\phi_2|^2\right)\phi_2, \end{cases} \quad (13)$$

with homogeneous Dirichlet boundary conditions, $\phi_j = 0$ on $\partial\mathcal{D}$ for $j = 1, 2$. The system (13) is solved (for fixed values of μ_1 and μ_2) using a Newton method that we discuss in the next section. The characterization of a stationary solution (and branches of solutions thereof) to the GP system (13) is based on the total energy

$$\mathcal{E}(\phi_1, \phi_2) = \int_{\mathcal{D}} \sum_{i=1}^2 \left(\frac{1}{2} |\nabla \phi_i|^2 + C_{\text{trap}} |\phi_i|^2 + \frac{1}{2} \sum_{j=1}^2 \beta_{ij} |\phi_i|^2 |\phi_j|^2 \right) d\mathbf{x}, \quad (14)$$

as well as the total number of atoms

$$N(\phi_1, \phi_2) = N(\phi_1) + N(\phi_2), \quad (15)$$

where $N(\cdot)$ is given by Eq. (6).

To study the spectral stability of the solutions we proceed as follows. We consider the perturbation Ansätze

$$\psi_1(\mathbf{x}, t) = \left[\phi_1(\mathbf{x}) + \delta \left(A(\mathbf{x})e^{-i\omega t} + \overline{B}(\mathbf{x})e^{i\overline{\omega}t} \right) \right] e^{-i\mu_1 t}, \quad (16a)$$

$$\psi_2(\mathbf{x}, t) = \left[\phi_2(\mathbf{x}) + \delta \left(C(\mathbf{x})e^{-i\omega t} + \overline{D}(\mathbf{x})e^{i\overline{\omega}t} \right) \right] e^{-i\mu_2 t}, \quad (16b)$$

where $A, B, C, D, \omega \in \mathbb{C}$. After plugging Eqs. (16a)-(16b) into Eqs. (11), we obtain the BdG equations at order $\mathcal{O}(\delta)$ for the two-component case, conveniently written as

$$M \begin{pmatrix} A \\ B \\ C \\ D \end{pmatrix} = \omega \begin{pmatrix} A \\ B \\ C \\ D \end{pmatrix}, \quad (17)$$

where the matrix M is:

$$M = \begin{pmatrix} M_{11} & \beta_{11}\phi_1^2 & \beta_{12}\phi_1\bar{\phi}_2 & \beta_{12}\phi_1\phi_2 \\ -\beta_{11}\bar{\phi}_1^2 & M_{22} & -\beta_{12}\bar{\phi}_1\phi_2 & -\beta_{12}\bar{\phi}_1\phi_2 \\ \beta_{21}\bar{\phi}_1\phi_2 & \beta_{21}\phi_1\phi_2 & M_{33} & \beta_{22}\phi_2^2 \\ -\beta_{21}\bar{\phi}_1\phi_2 & -\beta_{21}\phi_1\bar{\phi}_2 & -\beta_{22}\bar{\phi}_2^2 & M_{44} \end{pmatrix}, \quad (18)$$

with matrix elements

$$\begin{cases} M_{11} &= \mathcal{H} - \mu_1 + 2\beta_{11}|\phi_1|^2 + \beta_{12}|\phi_2|^2, \\ M_{22} &= -M_{11}, \\ M_{33} &= \mathcal{H} - \mu_2 + \beta_{21}|\phi_1|^2 + 2\beta_{22}|\phi_2|^2, \\ M_{44} &= -M_{33}, \end{cases} \quad (19)$$

and \mathcal{H} is given by Eq. (9).

3 The computation of stationary solutions to the GP equations

In this section we discuss the computational methods we employed in FreeFem++ in order to obtain stationary solutions to the GP equations. We begin our discussion by considering the existence problem for the one-component setting in Sec. 3.1 first, and then move on with the two-component case in Sec. 3.2. Both implementations in FreeFem++ are discussed in Sec. 3.3.

3.1 Newton's method for a single-component BEC

For the computation of stationary solutions to Eq. (4), we use Newton's method Kelley (2003). We first split the complex-valued wave function ϕ into real and imaginary parts via $\phi = \phi_r + i\phi_i$ and obtain from (4) the following coupled system of nonlinear equations

$$\begin{cases} -\frac{1}{2}\nabla^2\phi_r + C_{\text{trap}}\phi_r + \beta f(\phi_r, \phi_i)\phi_r - \mu\phi_r = 0, \\ -\frac{1}{2}\nabla^2\phi_i + C_{\text{trap}}\phi_i + \beta f(\phi_r, \phi_i)\phi_i - \mu\phi_i = 0. \end{cases} \quad (20)$$

We introduced in (20) 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 adopted in the toolbox.

The homogeneous Dirichlet conditions for the complex-valued wave function ϕ translate into imposing the same boundary conditions for ϕ_r and ϕ_i , that is $\phi_r = \phi_i = 0$ on $\partial\mathcal{D}$. After setting the classical Sobolev spaces Adams and Fournier (2003) $V = H_0^1(\mathcal{D})$ for ϕ_r and ϕ_i , we define the weak formulation (mandatory for the finite-element implementation) of Eq. (20) as: find $(\phi_r, \phi_i) \in V \times V = V^2$, such that for all test functions $(v_r, v_i) \in V^2$

$$\begin{cases} \mathcal{F}_r(\phi_r, \phi_i, v_r) = \int_{\mathcal{D}} (C_{\text{trap}} - \mu) \phi_r v_r + \int_{\mathcal{D}} \frac{1}{2} \nabla \phi_r \cdot \nabla v_r + \int_{\mathcal{D}} \beta f(\phi_r, \phi_i) \phi_r v_r = 0, \\ \mathcal{F}_i(\phi_r, \phi_i, v_i) = \int_{\mathcal{D}} (C_{\text{trap}} - \mu) \phi_i v_i + \int_{\mathcal{D}} \frac{1}{2} \nabla \phi_i \cdot \nabla v_i + \int_{\mathcal{D}} \beta f(\phi_r, \phi_i) \phi_i v_i = 0. \end{cases} \quad (21)$$

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

$$q = \phi_r^k - \phi_r^{k+1}, \quad s = \phi_i^k - \phi_i^{k+1}, \quad k \geq 0, \quad (22)$$

where q and s are solutions of the linearized equations

$$\begin{pmatrix} \left(\frac{\partial \mathcal{F}_r}{\partial \phi_r} \right)_{\phi_r=\phi_r^k, \phi_i=\phi_i^k} & \left(\frac{\partial \mathcal{F}_r}{\partial \phi_i} \right)_{\phi_r=\phi_r^k, \phi_i=\phi_i^k} \\ \left(\frac{\partial \mathcal{F}_i}{\partial \phi_r} \right)_{\phi_r=\phi_r^k, \phi_i=\phi_i^k} & \left(\frac{\partial \mathcal{F}_i}{\partial \phi_i} \right)_{\phi_r=\phi_r^k, \phi_i=\phi_i^k} \end{pmatrix} \begin{pmatrix} q \\ s \end{pmatrix} = \begin{pmatrix} \mathcal{F}_r(\phi_r^k, \phi_i^k, v_r) \\ \mathcal{F}_i(\phi_r^k, \phi_i^k, v_i) \end{pmatrix}, \quad (23)$$

with the corresponding weak formulation

$$\begin{cases} \int_{\mathcal{D}} (C_{\text{trap}} - \mu) q v_r + \int_{\mathcal{D}} \frac{1}{2} \nabla q \cdot \nabla v_r + \int_{\mathcal{D}} \beta \left(\frac{\partial f}{\partial \phi_r}(\phi_r^k, \phi_i^k) \phi_r^k q + \frac{\partial f}{\partial \phi_i}(\phi_r^k, \phi_i^k) \phi_r^k s + f(\phi_r^k, \phi_i^k) q \right) v_r \\ \quad = \int_{\mathcal{D}} (C_{\text{trap}} - \mu) \phi_r^k v_r + \int_{\mathcal{D}} \frac{1}{2} \nabla \phi_r^k \cdot \nabla v_r + \int_{\mathcal{D}} \beta f(\phi_r^k, \phi_i^k) \phi_r^k v_r, \\ \int_{\mathcal{D}} (C_{\text{trap}} - \mu) s v_i + \int_{\mathcal{D}} \frac{1}{2} \nabla s \cdot \nabla v_i + \int_{\mathcal{D}} \beta \left(\frac{\partial f}{\partial \phi_r}(\phi_r^k, \phi_i^k) \phi_i^k q + \frac{\partial f}{\partial \phi_i}(\phi_r^k, \phi_i^k) \phi_i^k s + f(\phi_r^k, \phi_i^k) s \right) v_i \\ \quad = \int_{\mathcal{D}} (C_{\text{trap}} - \mu) \phi_i^k v_i + \int_{\mathcal{D}} \frac{1}{2} \nabla \phi_i^k \cdot \nabla v_i + \int_{\mathcal{D}} \beta f(\phi_r^k, \phi_i^k) \phi_i^k v_i. \end{cases} \quad (24)$$

Note that the implementation of Eqs. (24) 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 method for the two-component BEC

The two-component GP system (13) is solved similarly by means of Newton's method, after splitting ϕ_1 and ϕ_2 into real and imaginary parts via $\phi_1 = \phi_{1r} + i\phi_{1i}$ and $\phi_2 = \phi_{2r} + i\phi_{2i}$. Equations (13) are thus converted into a system consisting of four real-valued (nonlinear) equations:

$$\begin{cases} -\frac{1}{2}\nabla^2\phi_{1r} + (C_{\text{trap}} - \mu_1)\phi_{1r} + \beta_{11}f(\phi_{1r}, \phi_{1i})\phi_{1r} + \beta_{12}f(\phi_{2r}, \phi_{2i})\phi_{1r} = 0, \\ -\frac{1}{2}\nabla^2\phi_{1i} + (C_{\text{trap}} - \mu_1)\phi_{1i} + \beta_{11}f(\phi_{1r}, \phi_{1i})\phi_{1i} + \beta_{12}f(\phi_{2r}, \phi_{2i})\phi_{1i} = 0, \\ -\frac{1}{2}\nabla^2\phi_{2r} + (C_{\text{trap}} - \mu_2)\phi_{2r} + \beta_{21}f(\phi_{1r}, \phi_{1i})\phi_{2r} + \beta_{22}f(\phi_{2r}, \phi_{2i})\phi_{2r} = 0, \\ -\frac{1}{2}\nabla^2\phi_{2i} + (C_{\text{trap}} - \mu_2)\phi_{2i} + \beta_{21}f(\phi_{1r}, \phi_{1i})\phi_{2i} + \beta_{22}f(\phi_{2r}, \phi_{2i})\phi_{2i} = 0. \end{cases} \quad (25)$$

Again, homogeneous Dirichlet boundary conditions on ϕ_1 and ϕ_2 are imposed: $\phi_{1r} = \phi_{1i} = \phi_{2r} = \phi_{2i} = 0$ on $\partial\mathcal{D}$. The weak formulation of Eqs. (25) can be written as follows: find $(\phi_{1r}, \phi_{1i}, \phi_{2r}, \phi_{2i}) \in V^4$, such that for all test functions $(v_{1r}, v_{1i}, v_{2r}, v_{2i}) \in V^4$

$$\begin{cases} \mathcal{F}_{1r} = \int_{\mathcal{D}} (C_{\text{trap}} - \mu_1)\phi_{1r}v_{1r} + \int_{\mathcal{D}} \frac{1}{2}\nabla\phi_{1r} \cdot \nabla v_{1r} + \int_{\mathcal{D}} (\beta_{11}f(\phi_{1r}, \phi_{1i}) + \beta_{12}f(\phi_{2r}, \phi_{2i}))\phi_{1r}v_{1r} = 0, \\ \mathcal{F}_{1i} = \int_{\mathcal{D}} (C_{\text{trap}} - \mu_1)\phi_{1i}v_{1i} + \int_{\mathcal{D}} \frac{1}{2}\nabla\phi_{1i} \cdot \nabla v_{1i} + \int_{\mathcal{D}} (\beta_{11}f(\phi_{1r}, \phi_{1i}) + \beta_{12}f(\phi_{2r}, \phi_{2i}))\phi_{1i}v_{1i} = 0, \\ \mathcal{F}_{2r} = \int_{\mathcal{D}} (C_{\text{trap}} - \mu_2)\phi_{2r}v_{2r} + \int_{\mathcal{D}} \frac{1}{2}\nabla\phi_{2r} \cdot \nabla v_{2r} + \int_{\mathcal{D}} (\beta_{21}f(\phi_{1r}, \phi_{1i}) + \beta_{22}f(\phi_{2r}, \phi_{2i}))\phi_{2r}v_{2r} = 0, \\ \mathcal{F}_{2i} = \int_{\mathcal{D}} (C_{\text{trap}} - \mu_2)\phi_{2i}v_{2i} + \int_{\mathcal{D}} \frac{1}{2}\nabla\phi_{2i} \cdot \nabla v_{2i} + \int_{\mathcal{D}} (\beta_{21}f(\phi_{1r}, \phi_{1i}) + \beta_{22}f(\phi_{2r}, \phi_{2i}))\phi_{2i}v_{2i} = 0. \end{cases} \quad (26)$$

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

$$q_1 = \phi_{1r}^k - \phi_{1r}^{k+1}, \quad s_1 = \phi_{1i}^k - \phi_{1i}^{k+1}, \quad q_2 = \phi_{2r}^k - \phi_{2r}^{k+1}, \quad s_2 = \phi_{2i}^k - \phi_{2i}^{k+1}, \quad (27)$$

which are solutions to the following system of linear equations:

$$\begin{aligned} & \int_{\mathcal{D}} (C_{\text{trap}} - \mu_1)q_1v_{1r} + \int_{\mathcal{D}} \frac{1}{2}\nabla q_1 \cdot \nabla v_{1r} + \int_{\mathcal{D}} (\beta_{11}f(\phi_{1r}^k, \phi_{1i}^k) + \beta_{12}f(\phi_{2r}^k, \phi_{2i}^k))q_1v_{1r} \\ & + \int_{\mathcal{D}} \beta_{11} \left(\frac{\partial f}{\partial \phi_r}(\phi_{1r}^k, \phi_{1i}^k)\phi_{1r}^k q_1 + \frac{\partial f}{\partial \phi_i}(\phi_{1r}^k, \phi_{1i}^k)\phi_{1r}^k s_1 \right) v_{1r} \\ & + \int_{\mathcal{D}} \beta_{12} \left(\frac{\partial f}{\partial \phi_r}(\phi_{2r}^k, \phi_{2i}^k)\phi_{1r}^k q_2 + \frac{\partial f}{\partial \phi_i}(\phi_{2r}^k, \phi_{2i}^k)\phi_{1r}^k s_2 \right) v_{1r} \\ & = \int_{\mathcal{D}} (C_{\text{trap}} - \mu_1)\phi_{1r}^k v_{1r} + \int_{\mathcal{D}} \frac{1}{2}\nabla\phi_{1r}^k \cdot \nabla v_{1r} + \int_{\mathcal{D}} (\beta_{11}f(\phi_{1r}^k, \phi_{1i}^k) + \beta_{12}f(\phi_{2r}^k, \phi_{2i}^k))\phi_{1r}^k v_{1r}, \end{aligned} \quad (28)$$

$$\begin{aligned}
& \int_{\mathcal{D}} (C_{\text{trap}} - \mu_1) s_1 v_{1i} + \int_{\mathcal{D}} \frac{1}{2} \nabla s_1 \cdot \nabla v_{1i} + \int_{\mathcal{D}} (\beta_{11} f(\phi_{1r}^k, \phi_{1i}^k) + \beta_{12} f(\phi_{2r}^k, \phi_{2i}^k)) s_1 v_{1i} \\
& + \int_{\mathcal{D}} \beta_{11} \left(\frac{\partial f}{\partial \phi_r}(\phi_{1r}^k, \phi_{1i}^k) \phi_{1i}^k q_1 + \frac{\partial f}{\partial \phi_i}(\phi_{1r}^k, \phi_{1i}^k) \phi_{1i}^k s_1 \right) v_{1i} \\
& + \int_{\mathcal{D}} \beta_{12} \left(\frac{\partial f}{\partial \phi_r}(\phi_{2r}^k, \phi_{2i}^k) \phi_{1i}^k q_2 + \frac{\partial f}{\partial \phi_i}(\phi_{2r}^k, \phi_{2i}^k) \phi_{1i}^k s_2 \right) v_{1i} \\
& = \int_{\mathcal{D}} (C_{\text{trap}} - \mu_1) \phi_{1i}^k v_{1i} + \int_{\mathcal{D}} \frac{1}{2} \nabla \phi_{1i}^k \cdot \nabla v_{1i} + \int_{\mathcal{D}} (\beta_{11} f(\phi_{1r}^k, \phi_{1i}^k) + \beta_{12} f(\phi_{2r}^k, \phi_{2i}^k)) \phi_{1i}^k v_{1i},
\end{aligned} \tag{29}$$

$$\begin{aligned}
& \int_{\mathcal{D}} (C_{\text{trap}} - \mu_2) q_2 v_{2r} + \int_{\mathcal{D}} \frac{1}{2} \nabla q_2 \cdot \nabla v_{2r} + \int_{\mathcal{D}} (\beta_{22} f(\phi_{2r}^k, \phi_{2i}^k) + \beta_{21} f(\phi_{1r}^k, \phi_{1i}^k)) q_2 v_{2r} \\
& + \int_{\mathcal{D}} \beta_{21} \left(\frac{\partial f}{\partial \phi_r}(\phi_{1r}^k, \phi_{1i}^k) \phi_{2r}^k q_1 + \frac{\partial f}{\partial \phi_i}(\phi_{1r}^k, \phi_{1i}^k) \phi_{2r}^k s_1 \right) v_{2r} \\
& + \int_{\mathcal{D}} \beta_{22} \left(\frac{\partial f}{\partial \phi_r}(\phi_{2r}^k, \phi_{2i}^k) \phi_{2r}^k q_2 + \frac{\partial f}{\partial \phi_i}(\phi_{2r}^k, \phi_{2i}^k) \phi_{2r}^k s_2 \right) v_{2r} \\
& = \int_{\mathcal{D}} (C_{\text{trap}} - \mu_2) \phi_{2r}^k v_{2r} + \int_{\mathcal{D}} \frac{1}{2} \nabla \phi_{2r}^k \cdot \nabla v_{2r} + \int_{\mathcal{D}} (\beta_{21} f(\phi_{1r}^k, \phi_{1i}^k) + \beta_{22} f(\phi_{2r}^k, \phi_{2i}^k)) \phi_{2r}^k v_{2r},
\end{aligned} \tag{30}$$

$$\begin{aligned}
& \int_{\mathcal{D}} (C_{\text{trap}} - \mu_2) s_2 v_{2i} + \int_{\mathcal{D}} \frac{1}{2} \nabla s_2 \cdot \nabla v_{2i} + \int_{\mathcal{D}} (\beta_{22} f(\phi_{2r}^k, \phi_{2i}^k) + \beta_{21} f(\phi_{1r}^k, \phi_{1i}^k)) s_2 v_{2i} \\
& + \int_{\mathcal{D}} \beta_{21} \left(\frac{\partial f}{\partial \phi_r}(\phi_{1r}^k, \phi_{1i}^k) \phi_{2i}^k q_1 + \frac{\partial f}{\partial \phi_i}(\phi_{1r}^k, \phi_{1i}^k) \phi_{2i}^k s_1 \right) v_{2i} \\
& + \int_{\mathcal{D}} \beta_{22} \left(\frac{\partial f}{\partial \phi_r}(\phi_{2r}^k, \phi_{2i}^k) \phi_{2i}^k q_2 + \frac{\partial f}{\partial \phi_i}(\phi_{2r}^k, \phi_{2i}^k) \phi_{2i}^k s_2 \right) v_{2i} \\
& = \int_{\mathcal{D}} (C_{\text{trap}} - \mu_2) \phi_{2i}^k v_{2i} + \int_{\mathcal{D}} \frac{1}{2} \nabla \phi_{2i}^k \cdot \nabla v_{2i} + \int_{\mathcal{D}} (\beta_{21} f(\phi_{1r}^k, \phi_{1i}^k) + \beta_{22} f(\phi_{2r}^k, \phi_{2i}^k)) \phi_{2i}^k v_{2i}.
\end{aligned} \tag{31}$$

Again, the implementation of Eqs. (28)-(31) 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++ (Hecht, 2012) 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 formulation. For example, the weak form of the system of linear equations given by Eqs. (24) and used in Newton's method is conveniently implemented as a list of rules and expressions embodied in a `Macro` (see, in particular, `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, in particular, the file `BdG_1comp_ddm/INIT/2D_TF.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 given by Eqs. (28)-(29) 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 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 $P2$ (piece-wise 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 gradients 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. We briefly discuss next the key points of the mesh adaptation techniques the toolbox employs (further details can be found in our recent contribution ([SADAKA et al., 2024](#))).

For solutions to the 2D GP equations at hand, the mesh is adapted by using the built-in `adaptmesh` command in FreeFem++. In short, the underlying algorithm modifies the inner products (which consider the Hessian of the solution by default) in the mesh generator to evaluate distance and volume ([Borouchaki et al., 1996](#); [Frey and George, 1999](#); [Mohammadi and Pironneau, 2000](#)). This way, quasi-equilateral elements are constructed, accordingly to the new metric. On the other hand, and for 3D configurations, adaptive mesh refinement in FreeFem++ is performed through the use of the libraries `mshmet` and `mmg` ([Dapogny et al., 2014](#)) where similar algorithms are employed. Note that the user can adjust the values of `hmax` and `hmin` representing the maximum and minimum edge sizes of the triangular mesh, respectively during the mesh adaptation process. This offers the possibility to control the size of the mesh, and thus find a trade-off between accuracy and computational cost. While performing mesh adaptation, however, one faces with the important question about what is the right choice of variables that is suitable for controlling mesh adaptivity. 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 [Danaila and Hecht \(2010\)](#), 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:

$$\left\| \begin{pmatrix} q \\ s \end{pmatrix} \right\|_{\infty} < \epsilon_q, \quad \left\| \begin{pmatrix} \mathcal{F}_r \\ \mathcal{F}_i \end{pmatrix} \right\|_2 < \epsilon_F, \quad (32)$$

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 being 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 ([Allgower and Georg, 1990](#)) 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, and it is described briefly next. 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} (as for the 2D) 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 of using the SLEPc library (Hernandez et al., 2005) which is now interfaced with FreeFem++ . First, we need to 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. (8) reads:

$$\begin{cases} \int_{\mathcal{D}} \frac{1}{2} \nabla A \cdot \nabla v_1 + \int_{\mathcal{D}} (C_{\text{trap}} - \mu) A v_1 + \int_{\mathcal{D}} 2\beta |\phi|^2 A v_1 + \int_{\mathcal{D}} \beta \phi^2 B v_1 = \omega \int_{\mathcal{D}} A v_1, \\ - \int_{\mathcal{D}} \frac{1}{2} \nabla B \cdot \nabla v_2 - \int_{\mathcal{D}} (C_{\text{trap}} + \mu) B v_2 - \int_{\mathcal{D}} 2\beta |\phi|^2 B v_2 - \int_{\mathcal{D}} \beta \bar{\phi}^2 A v_2 = \omega \int_{\mathcal{D}} B v_2. \end{cases} \quad (33)$$

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. (33) 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. (33) and its implementation in the above macro (see, also, the file `BdG_1comp_ddm/A_macro/Macro_problem.edp`). For the computation of the BdG spectrum, we apply a small shift, *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"`
to slightly regularize the eigenproblem.

Upon computing the eigenvalues and eigenvectors of the BdG problem in SLEPc, we further check their accuracy by displaying the residual or Eq. (8):

$$\left\| M \begin{pmatrix} A \\ B \end{pmatrix} - \omega \begin{pmatrix} A \\ B \end{pmatrix} \right\|_{\infty} \quad (34)$$

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. (17)-(19):

$$\left\{ \begin{aligned} & \int_{\mathcal{D}} \frac{1}{2} \nabla A \cdot \nabla v_1 + \int_{\mathcal{D}} (C_{\text{trap}} - \mu_1) A v_1 + \int_{\mathcal{D}} (2\beta_{11}|\phi_1|^2 + \beta_{12}|\phi_2|^2) A v_1 \\ & \quad + \int_{\mathcal{D}} \beta_{11}\phi_1^2 B v_1 + \int_{\mathcal{D}} \beta_{12}\phi_1 \overline{\phi_2} C v_1 + \int_{\mathcal{D}} \beta_{12}\phi_1 \phi_2 D v_1 = \omega \int_{\mathcal{D}} A v_1, \\ & - \int_{\mathcal{D}} \frac{1}{2} \nabla B \cdot \nabla v_2 - \int_{\mathcal{D}} (C_{\text{trap}} - \mu) B v_2 - \int_{\mathcal{D}} (2\beta_{11}|\phi_1|^2 + \beta_{12}|\phi_2|^2) B v_2 \\ & \quad - \int_{\mathcal{D}} \beta_{11}\overline{\phi_1}^2 A v_2 - \int_{\mathcal{D}} \beta_{12}\overline{\phi_1}\phi_2 C v_2 - \int_{\mathcal{D}} \beta_{12}\overline{\phi_1}\phi_2 D v_2 = \omega \int_{\mathcal{D}} B v_2, \\ & \int_{\mathcal{D}} \frac{1}{2} \nabla C \cdot \nabla v_3 + \int_{\mathcal{D}} (C_{\text{trap}} - \mu) C v_3 + \int_{\mathcal{D}} (2\beta_{22}|\phi_2|^2 + \beta_{21}|\phi_1|^2) C v_3 \\ & \quad + \int_{\mathcal{D}} \beta_{21}\overline{\phi_1}\phi_2 A v_3 + \int_{\mathcal{D}} \beta_{21}\phi_1 \phi_2 B v_3 + \int_{\mathcal{D}} \beta_{22}\phi_2^2 D v_3 = \omega \int_{\mathcal{D}} C v_3, \\ & - \int_{\mathcal{D}} \frac{1}{2} \nabla D \cdot \nabla v_4 - \int_{\mathcal{D}} (C_{\text{trap}} - \mu) D v_4 - \int_{\mathcal{D}} (2\beta_{22}|\phi_2|^2 + \beta_{21}|\phi_1|^2) D v_4 \\ & \quad - \int_{\mathcal{D}} \beta_{21}\overline{\phi_1}\phi_2 A v_4 - \int_{\mathcal{D}} \beta_{21}\phi_1 \overline{\phi_2} B v_4 - \int_{\mathcal{D}} \beta_{22}\overline{\phi_2}^2 C v_4 = \omega \int_{\mathcal{D}} D v_4. \end{aligned} \right. \quad (35)$$

Again, the implementation of the BdG problem of Eq. (35) 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

```

The validation of the new toolbox is a necessary task for assessing its performance, fidelity and reliability. In Secs. 5 and 6, we perform a series of validation tests that exist in the literature. Note that the test cases we present next for the one-component case are the same as the ones considered in (SADAKA et al., 2024).

5 Validation test cases for the one-component BEC

We begin our discussion on validation test cases of our toolbox by considering first the one-component GP model in 2D and 3D. We note that all the cases we present below consider repulsive interactions (we fix $\beta = 1$) and an isotropic trap $C_{\text{trap}} = \frac{1}{2}\omega_{\perp}^2 r^2$, with $r^2 = x^2 + y^2 + z^2$.

To ease our discussion, we further present a summary of the considered cases, together with typical computational times and mesh sizes (*i. e.* the number of elements), in Table 1. 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. We note too, however, that it is safer to use adaptive mesh refinement while exploring branches of solutions for which their topology is unknown.

	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 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.

GP test cases	Processors	CPU time	niter	ndof	nt	maxRSS
3D dark soliton	28	00:01:01	168	205,822	76,455	0.78 Gb
3D vortex line (1VL)	28	00:04:02	168	329,988	122,969	0.79 Gb
3D vortex ring + 2VLs	56	00:05:19	201	654,802	244,597	1.10 Gb

BdG test cases	Processors	CPU time	niter	ndof	nnz	maxRSS
3D dark soliton	28	00:01:30	56	103,116	11,782,505	3.07 Gb
3D vortex line (1VL)	28	00:03:41	56	165,362	18,934,023	7.09 Gb
3D vortex ring + 2VLs	56	00:08:02	67	327,887	37,600,455	8.57 Gb

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 performed (for tracing the respective branches) are shown. Moreover, the table contains the number of times the BdG problem was solved (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.

5.1 2D case: Ground state

The distribution of the BdG modes for oscillations of the ground state in the TF limit for repulsive BECs was derived in (Kevrekidis and Pelinovsky, 2010), and is given by

$$\omega_{m,k}^{\text{TF}} = \omega_{\perp} \sqrt{m + 2k^2 + 2k(1 + m)}, \quad (36)$$

where $m, k \geq 0$ are integers.

In Table 3, we present the first 20 BdG modes our toolbox computed for $\mu = 6$ and $\omega_{\perp} = 0.2$. Moreover, we compare results that were obtained with 4 MPI processors and without MPI against the ones from Eq. (36). The Krein signatures (see, Eq. (10)) were computed too by the toolbox, and found all to be 1, thus suggesting the absence of negative energy modes. Note the perfect match between numerical and theoretical results together with the fact that mesh adaptation provides the same results as computations with a refined fixed mesh.

5.2 2D case: Dark soliton

We test next the 2D dark soliton known also as the dark-soliton stripe (see Charalampidis et al. (2018) and references therein). At the linear limit, this state is constructed as

$$\phi_{DS} = \sqrt{\frac{\omega_{\perp}}{2\pi}} H_0(\sqrt{\omega_{\perp}} x) H_1(\sqrt{\omega_{\perp}} y) e^{-\frac{1}{2}\omega_{\perp}(x^2+y^2)}, \quad (37)$$

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.

We present the numerical results we obtained from the toolbox in Figs. 1-b) and 2-c)-d), and we compare them to the ones reported in (SADAKA et al., 2024) in Figs. 1-a) and 2-a)-b). Note that we used the new μ -continuation strategy, with adapted step (see the last paragraphs in Sec. 3.2). In particular, the real and imaginary parts of ω as a function of μ are depicted in panels a) and b) of Fig. 1 without MPI and with 4 MPI processors, respectively. The results are identical to the results of Middelkamp et al. (2010) (note also the emergence of a cascade of pitchfork, *i. e.* symmetry-breaking bifurcations). Panels a) and c) (resp. b) and d)) in Fig. 2 show the final adapted mesh and the respective density $|\phi|^2$ of the solution for $\mu = 3$ without MPI and with 4 MPI processors, respectively. A few remarks are in order. Per the discretized problem the toolbox solves, there 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.

	without MPI			with 4 MPI processors			$\omega_{m,k}$ from (36)
	$Re(\omega)$	$Im(\omega)$	K	$Re(\omega)$	$Im(\omega)$	K	
ω_1	-5.86493e-07	8.69311e-14	1	-6.15872e-07	8.07869e-14	1	$\omega_{0,0}^{\text{TF}} = 0$
ω_2	5.86493e-07	-8.69062e-14	1	6.15872e-07	-8.07646e-14	1	
ω_3	-0.200005	4.07202e-15	1	-0.200005	8.1375e-15	1	$\omega_{1,0}^{\text{TF}} = 0.2$
ω_4	0.200005	7.56706e-16	1	0.200005	7.12452e-15	1	
ω_5	-0.200005	2.10494e-15	1	-0.200005	-5.09397e-15	1	
ω_6	0.200005	-8.97438e-18	1	0.200005	4.42765e-15	1	
ω_7	-0.283448	-1.87346e-14	1	-0.283448	-3.8324e-16	1	$\omega_{2,0}^{\text{TF}} = 0.28284271$
ω_8	0.283448	2.15388e-15	1	0.283448	7.56616e-16	1	
ω_9	-0.283467	1.286e-14	1	-0.283467	-1.40959e-15	1	
ω_{10}	0.283467	8.67883e-15	1	0.283467	-2.90574e-15	1	
ω_{11}	-0.348769	-5.37074e-16	1	-0.348769	1.69936e-16	1	$\omega_{3,0}^{\text{TF}} = 0.34641016$
ω_{12}	0.348769	-6.03743e-15	1	0.348769	-1.13813e-15	1	
ω_{13}	-0.348769	-4.43208e-15	1	-0.348769	-9.10584e-17	1	
ω_{14}	0.348769	2.56399e-15	1	0.348769	-3.7267e-16	1	
ω_{15}	-0.400018	-5.01133e-15	1	-0.400018	-1.45788e-14	1	$\omega_{4,0}^{\text{TF}} = \omega_{0,1}^{\text{TF}} = 0.4$
ω_{16}	0.400018	-5.34787e-15	1	0.400018	-3.09639e-14	1	
ω_{17}	-0.405642	-4.75383e-15	1	-0.405642	3.36622e-16	1	
ω_{18}	0.405642	-3.58917e-15	1	0.405642	2.00132e-16	1	
ω_{19}	-0.405679	1.04733e-15	1	-0.405679	1.19496e-17	1	
ω_{20}	0.405679	7.04663e-15	1	0.405679	-2.51016e-16	1	

Table 3: Numerical results on the BdG modes that were obtained with and without parallelization for the 2D ground state and with the same shift $\sigma = 0.01$. In addition, the Krein signatures shown as K are included in the table together with the theoretical prediction of Eq. (36).

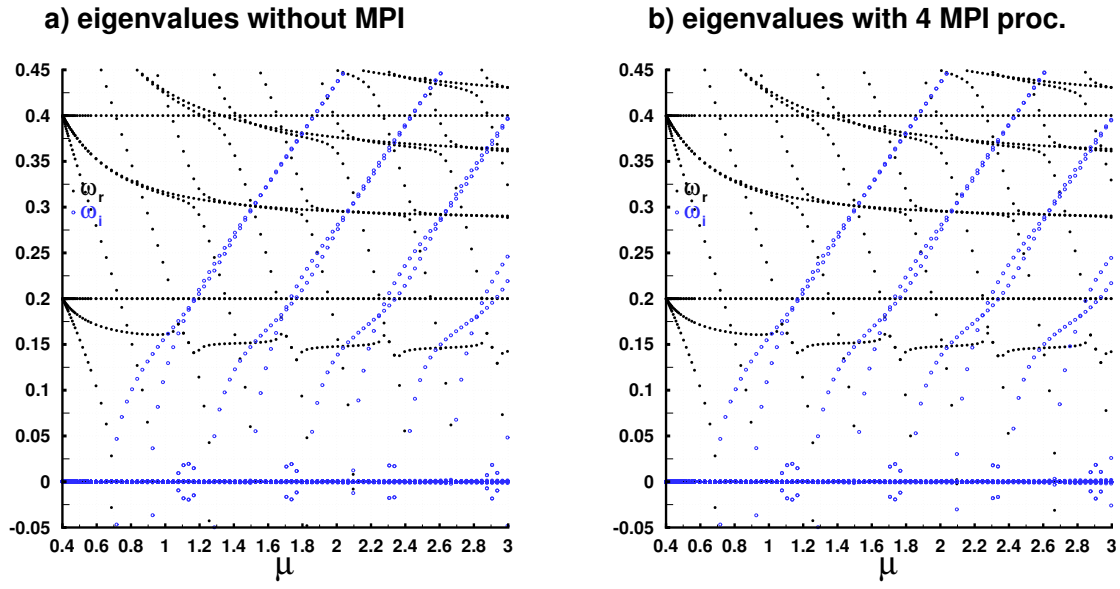


Figure 1: BdG results for the 2D dark soliton: a) real and imaginary parts of the eigenvalues, *i. e.* ω_r and ω_i , respectively, as a function of μ , and without using MPI; b) same as panel a) but using 4 MPI processors. Blue open and dark filled circles in both panels depict ω_i and ω_r , respectively.

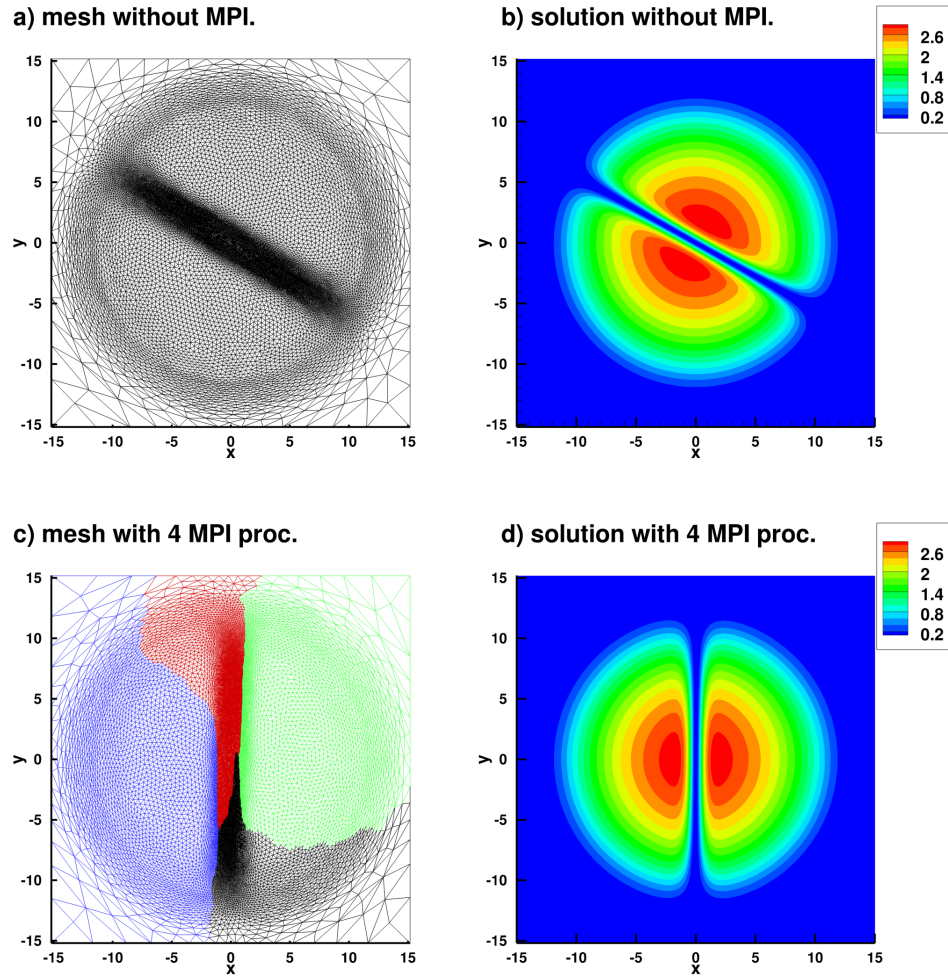


Figure 2: 2D dark soliton for $\mu = 3$: without MPI a) adapted mesh, b) density $|\phi|^2$ and with 4 MPI processors c) adapted mesh, d) density $|\phi|^2$.

5.3 3D case: Ground state

Considering 3D configurations renders the computation of the BdG spectrum a very challenging problem. This is the case because even with mesh adaptation, the number of degrees of freedom is still high, and increases with the size of the BEC which is controlled by the chemical potential μ (or, equivalently, by the number of atoms). In our previous contribution (SADAKA et al., 2024), as a prototypical case to test the published toolbox therein, we considered the 3D ground state whose BdG spectrum was computed for $\omega_{\perp} = 1$. We argued that the computation of more complex states requires the use of parallelization which is capable of reducing the computational time and memory requirements. Indeed, with the present parallel toolbox, we accomplish this goal. Illustratively, we compute the BdG spectrum of the 3D ground state, and respective results are shown presented in Fig. 3. The real part of the eigenvalues (ω_r) as a function of μ computed without MPI (using the sequential toolbox published in (SADAKA et al., 2024)) and with 4 MPI processors (and the μ -adaptivity continuation strategy discussed previously) are shown in the same panel. Both numerical results are in full agreement with the numerical results reported in (Bisset et al., 2015).

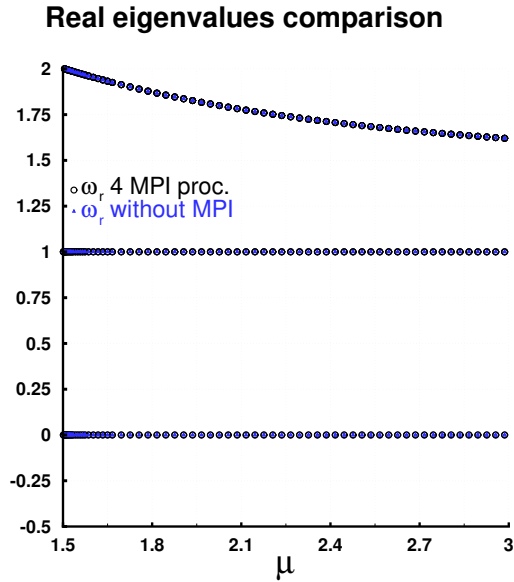


Figure 3: 3D ground state. Variation of the real part ω_r of the BdG spectrum as a function of μ . Results obtained using 4 MPI processors (black triangles) and without MPI (blue open circles).

5.4 Dark soliton in 3D

We now proceed with another 3D case to validate our parallel toolbox. The existence and BdG analysis of the dark soliton in 3D was considered in (Bisset et al., 2015), where the azimuthal symmetry of the state was taken into account in order to reduce the 3D BdG problem to a 2D one. Here, we perform a full 3D BdG analysis by using (similarly to (Bisset et al., 2015)) an isotropic potential with $\omega_{\perp} = 1$. We note that 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 (Bisset et al., 2015) (see also (Boullé et al., 2020)). 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. Upon using this eigenstate as a seed to Newton's method, we performed a continuation over μ . The results are shown in Fig. 4: the left panel depicts the BdG spectrum (real and imaginary parts of the eigenvalues are shown with black filled circles and blue open circles, respectively); the right panel shows the isosurface of the density $|\phi|^2$ for $\mu = 4.5$. It can be discerned from the left panel that our results match perfectly the ones published in (Bisset et al., 2015) (see Fig. 1 therein).

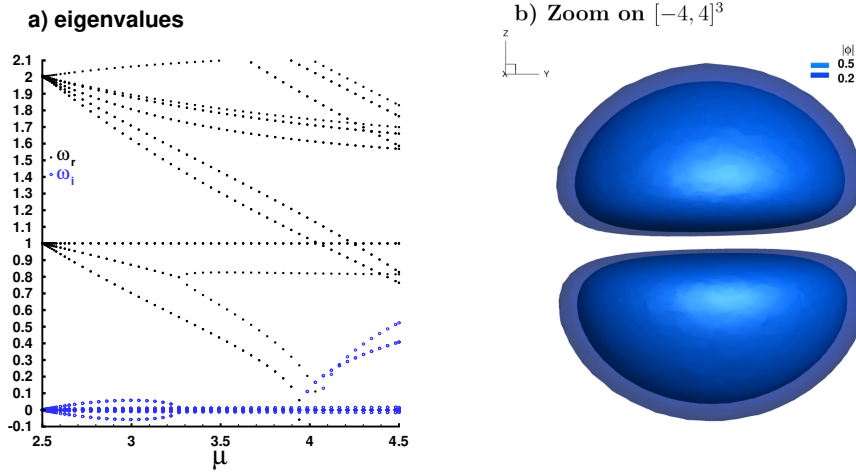


Figure 4: 3D dark soliton. a) The BdG spectrum and b) density $|\phi|^2$ for $\mu = 4.5$. The computational domain is the cube $[-5.4, 5.4]^3$.

5.5 3D case: Vortex-lines and beyond

We conclude our series of validation cases for the single-component GP equation in 3D by considering two extra cases (we set $\omega_{\perp} = 1$ as previously): a single-charged vortex-line (VL) state (Bisset et al., 2015), and a vortex-ring (VR) configuration bearing two (oppositely charged) VL handles Boullé et al. (2020). 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 $|0, 1, 0\rangle_{\text{cyl}} = r^2 L_0^1(x^2 + y^2) e^{i\theta} e^{-(x^2+y^2+z^2)/2}$ Boullé et al. (2020) (where L_0^1 stands for the associated Laguerre polynomial). Similarly, the vortex-ring (VR) with two handles bifurcates at $\mu = 3.5$ from the linear limit, and is constructed by the combination of Hermite polynomials (in Cartesian coordinates) $|2, 0, 0\rangle + |0, 2, 0\rangle + i|1, 0, 1\rangle$. Our toolbox was capable of tracing branches of solutions for both cases where the respective results are shown in Figs. 5 and 6, respectively. In particular, the panel a) in the figure depicts the BdG spectra of the pertinent states that match with the findings in (Bisset et al., 2015) and Boullé et al. (2020). Panel b) shows two snapshots of isosurfaces of the densities $|\phi|^2$ of the solutions for $\mu = 4.5$ and $\mu = 6$, respectively.

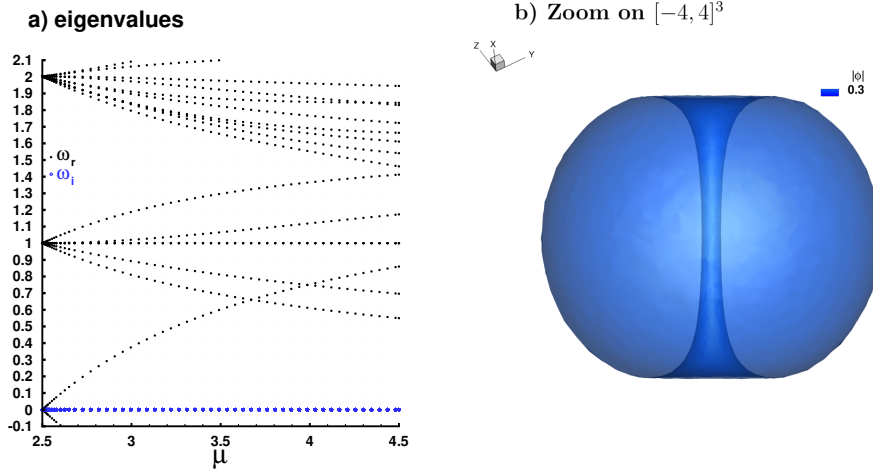


Figure 5: 3D single-charged vortex-line (VL) configuration inside a cube $[-5.4, 5.4]^3$. The format of the figure is the same as in Fig. 4 with the density $|\phi|^2$ shown in b) for $\mu = 4.5$.

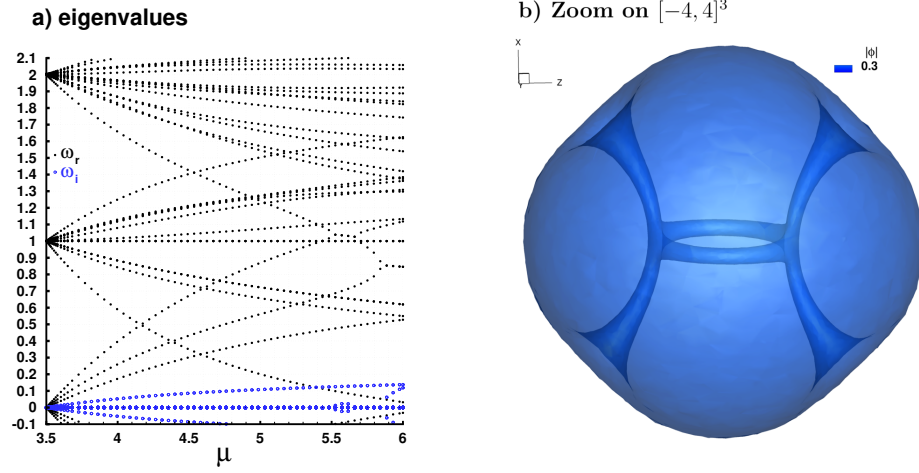


Figure 6: 3D vortex-ring (VR) with two (oppositely charged) VL handles configuration inside a cube $[-6.23, 6.23]^3$. The format of the figure is the same as in Fig. 4 with the density $|\phi|^2$ shown in b) for $\mu = 6$.

6 Validation test cases for the two-component BEC

We now move on to the study of localized configurations in two-component GP equations (13). Admittedly, the study of their existence, and more crucially, their BdG spectrum (upon solving Eqs. (17)-(19)) 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 4 summarizes the test cases we considered, and has the same format as Table 2.

GP test cases	Processes	CPU time	niter	ndof	nt	maxRSS
2D vortex-antidark state	4	00:00:17	46	30,506	3,744	1,11 Gb
2D ring-antidark state	4	00:00:09	105	37,847	4,634	4,53 Gb
2D Hermite LL 1	4	00:00:12	81	27,775	3,399	0,34 Gb
2D Hermite LL 7	4	00:00:29	81	58,081	7,175	2,22 Gb
2D Hermite LL 8	4	00:00:28	81	41,717	5,131	1,42 Gb
3D Hermite LL 1	28	00:02:18	81	24,698	3,936	1,22 Gb
3D Hermite LL 7	28	00:03:52	81	24,578	3,935	1,16 Gb

BdG test cases	Processes	CPU time	niter	ndof	nnz	maxRSS
2D vortex-antidark state	4	00:00:31	46	27,787	5,125,301	0,78 Gb
2D ring-antidark state	4	00:00:41	91	37,791	6,946,743	1,02 Gb
2D Hermite LL 1	4	00:00:33	81	25,303	4,644,479	0,75 Gb
2D Hermite LL 7	4	00:01:03	81	55,155	10,136,839	1,43 Gb
2D Hermite LL 8	4	00:00:37	81	38,882	7,142,816	1,11 Gb
3D Hermite LL 1	28	00:03:56	81	72,202	32,963,342	6,70 Gb
3D Hermite LL 7	28	00:03:21	81	71,627	32,681,221	7,03 Gb

Table 4: 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 .

6.1 2D two-component case: vortex dark-antidark configurations

For illustration purposes, we consider two test cases taken from (Danaila et al., 2016) corresponding to 2D vortex-antidark and dark-antidark ring solutions (see also Tab. 4 for a summary of our

results). 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. We consider the GP system (13) in the case of repulsive inter-component interactions with miscibility condition $0 \leq \beta_{12} < \sqrt{\beta_{11}\beta_{22}}$ which ensures that the two components co-exist outside the dark-antidark state. To simplify the case study (and following SADAKA et al. (2024)), we set $\beta_{11} = \beta_{22} = \beta = 1$, $\beta_{12} = \beta_{21}$, and $0 < \beta_{12} < \beta$ since the ratio between non-linear interaction coefficients matters only.

For this two-component case in 2D, we compute the BdG spectra of the vortex-antidark and dark-antidark ring solutions studied in (Danaila et al., 2016). The respective BdG results we obtained from our toolbox are summarized in panels a) and b) of Fig. 7, and densities of bound modes are shown in Figs. 8 (vortex-antidark states) and 9 (dark-antidark states). Note that ϕ_1 contains a ring soliton whereas ϕ_2 corresponds to the ground state in Fig. 9. Finally, we highlight the additional feature of our toolbox in considering distinct values of the interaction coefficients β_{ij} , $1 \leq i, j \leq 2$. This way, the user has the flexibility to adjust these values according to other configurations of interest (and potentially, experimental setups) for performing distinct numerical studies other than those reported in (Danaila et al., 2016).

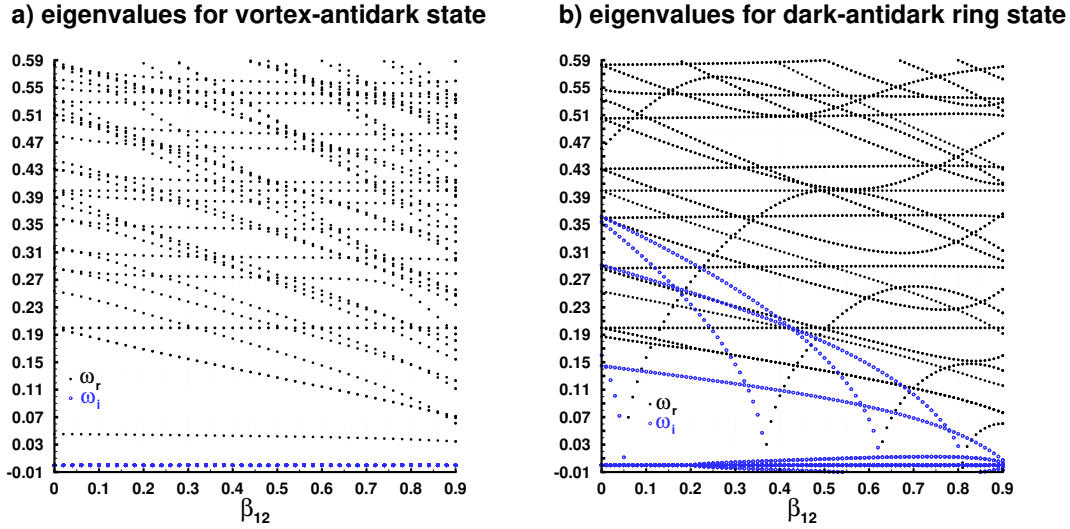


Figure 7: 2D two-component vortex dark-antidark configurations. The BdG spectrum of a) vortex-antidark and b) dark-antidark ring states as a function of β_{12} . Black filled and blue open circles correspond to the real (ω_r) and imaginary parts (ω_i) of the eigenvalues, respectively.

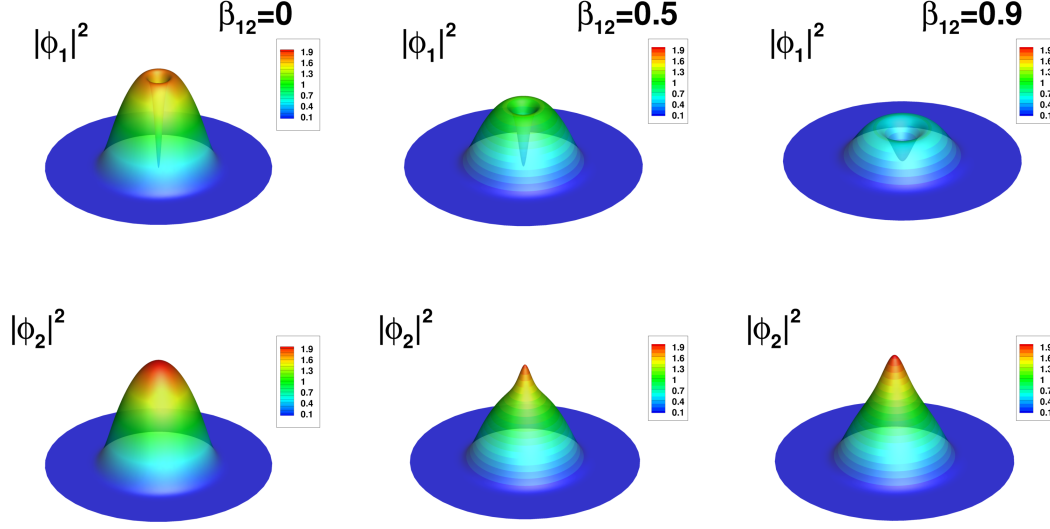


Figure 8: 2D two-component vortex-antidark case. Densities of components $|\phi_1|^2$ (top) and $|\phi_2|^2$ (bottom) corresponding to the vortex-antidark bound mode for different values of β_{12} . The computational domain is a disk centered at the origin and with radius 19.

6.2 2D two-component case: from soliton necklaces to multipoles

The last 2D cases that we considered for testing our parallel toolbox stem from [Charalampidis et al. \(2020\)](#). At first, we construct the ground state of Eq. (4) by

$$\phi = \sqrt{\frac{\omega_{\perp}}{2\pi}} H_0(\sqrt{\omega_{\perp}} x) H_0(\sqrt{\omega_{\perp}} y) e^{-\frac{1}{2}\omega_{\perp}(x^2+y^2)}, \quad (38)$$

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 $\mu = 1$. The terminal profile ϕ , now called ϕ_1 , is extracted while setting $\mu_1 = \mu = 1$. Then, we focus on Eq. (13) 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 [Charalampidis et al. \(2015\)](#), we plug the (terminal) profile ϕ_1 into the equation for ϕ_2 [cf. Eq. (13)], and linearize it with respect to ϕ_2 . This process results in the following eigenvalue problem for (μ_2, ϕ_2)

$$-\frac{1}{2}\nabla^2\phi_2 + C_{\text{eff}}\phi_2 = \mu_2\phi_2 \quad (39)$$

with $C_{\text{eff}} = C_{\text{trap}} + \beta_{21}|\phi_1|^2$ being the effective potential [Charalampidis et al. \(2015\)](#) which is responsible for “trapping” bound modes in the ϕ_2 component of Eq. (13). Upon solving Eq. (39)

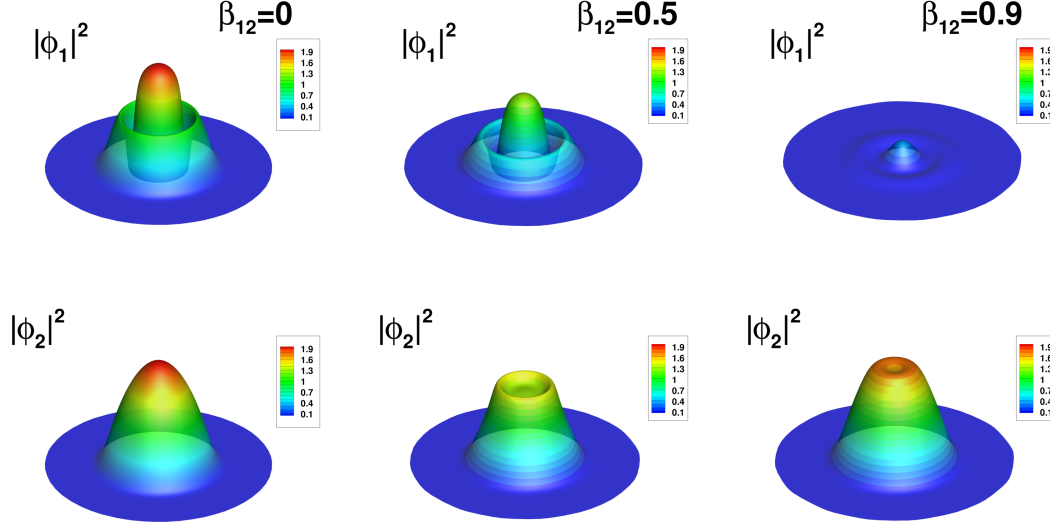


Figure 9: 2D two-component dark-antidark ring case. Same legend and computational domain as in Fig. 8.

numerically, we 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. (13), 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 that we present next.

The results for these test cases are depicted in Fig. 10 and Figs. 11-13 showcasing the BdG spectra and density profiles of the components, respectively. Specifically, the panels a), b), and c) of Fig. 10 are associated with density profiles of Figs. 11, 12, and 13, each of the latter shown for different values of μ_2 . The bound mode depicted in Fig. 11 corresponds to the dark-bright branch that bifurcates from the linear limit at $\mu_2 \approx 1.05133$. This state is unstable over μ_2 (see, panel a) of Fig. 10) except from a very narrow window of stability close to the linear limit (see also Fig. 2(a) in Charalampidis et al. (2020), and references therein). Our toolbox traced the branch of Fig. 12 that bifurcates at $\mu_2 \approx 1.23276$, and involves a soliton necklace in ϕ_2 (note its imprint on ϕ_1). Similar to the case of Fig. 10, this state is unstable (see also Fig. 1(a) in Charalampidis et al. (2020)) but features a very narrow window of stability as is shown in panel b) of Fig. 10. Finally, the so-called multipole branch (see Fig. 16(c) in Charalampidis et al. (2020)) was traced by our toolbox, with BdG spectrum and density profiles shown in the panel c) of Fig. 10, and Fig. 13, respectively. This state bifurcates from the linear limit at $\mu_2 = 1.29325$, and 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 [Charalampidis et al. \(2020\)](#).

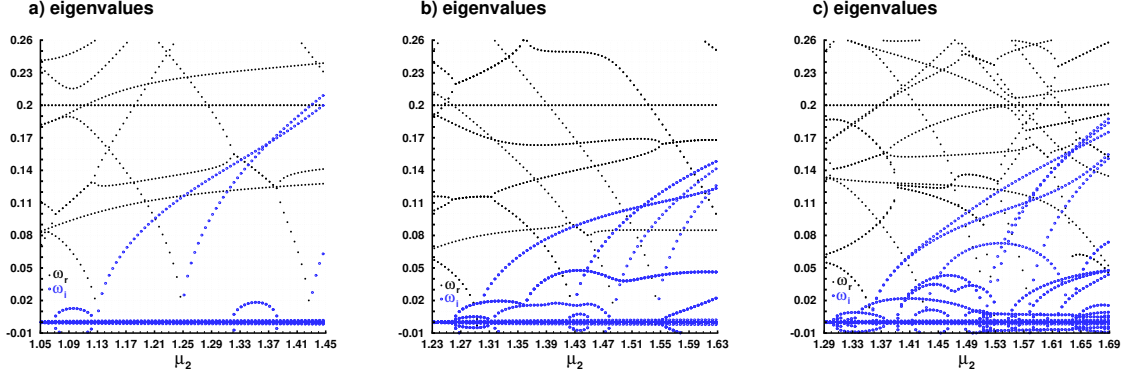


Figure 10: 2D two-component soliton and necklace configurations. The BdG spectra as functions of μ_2 corresponding to a) the dark-bright soliton branch, b) ground state (ϕ_1) and soliton necklace (ϕ_2) branch, and c) the multipole branch. Each of these branches bifurcate from $(\mu_1, \mu_2) \approx (1, 1.05133)$, $(1, 1.23276)$, and $(1, 1.29325)$. Their respective density profiles (for different values of μ_2) are shown in Figs. 11, 12, and 13.

6.3 3D two-component case: ground state, planar dark soliton, and vortex-ring state

We conclude our discussion on test cases for validating our toolbox by presenting two 3D two-component configurations. Note that we employ the same “trapping” technique that was discussed for the 2D, two-component cases, where the ϕ_1 carries the ground state, but now in 3D. The latter is continued over μ from its linear limit, *i. e.* $\mu \approx 1.501$, and in the single-component case first [cf. Eq. (4)] with $\beta = 1.03$ and $\omega_\perp = 1$, until $\mu = 2$. Then, the eigenvalue problem (39) is solved 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. (13) 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 .

The first case we considered involves the ground state in ϕ_1 , and the planar dark soliton [P. G. Kevrekidis and Carretero-González \(2015\)](#) in ϕ_2 , *i. e.* a dark-bright soliton stripe in 3D. The latter bifurcates from its linear limit at $\mu_2 \approx 2.79467$, and can be classified in terms of Cartesian eigenfunctions as $|0, 1, 0\rangle$. Our numerical results on its BdG spectrum depicted in panel a) of Fig. 14 show that the state is stable from its inception until $\mu_2 \approx 2.902$ when it becomes unstable. Snapshots of densities $|\phi_1|^2$ and $|\phi_2|^2$ are shown in the top and bottom rows

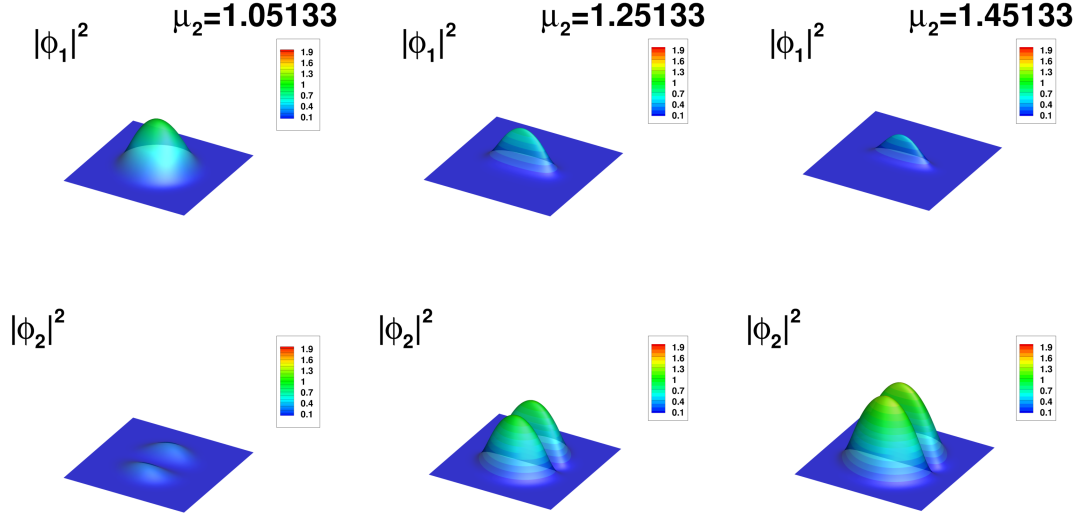


Figure 11: 2D two-component dark-bright soliton. Density profiles of each of the components, *i. e.* $|\phi_1|^2$ (top row) and $|\phi_2|^2$ (bottom row), for different values of μ_2 . For this case $\mu_1 = 1$ (see text), and the computational domain is $[-11.62, 11.62]^2$.

of Fig. 15, respectively, for different values of μ_2 . Finally, the panel b) of Fig. 14 and Fig. 16 present the BdG spectrum and density profiles corresponding to the bound mode involving the ground state in ϕ_1 , and the vortex-ring state in ϕ_2 , *i. e.* a vortex-ring-bright state. This configuration (obtained for $\mu_1 = 2$), bifurcates from the linear limit of ϕ_2 at $\mu_2 = 3.67602$, and is generically unstable as shown in the panel b) of Fig. 14. Note that in the one-component setting, the vortex-ring state can be classified in terms of a combination of Cartesian eigenstates as $\frac{1}{\sqrt{2}}(|2, 0, 0\rangle + |0, 2, 0\rangle) + i|0, 0, 1\rangle$ [Boullé et al. \(2020\)](#). Density profiles $|\phi_1|^2$ and $|\phi_2|^2$ of this bound mode are shown for different values of μ_2 in the top and bottom rows, of Fig. 16.

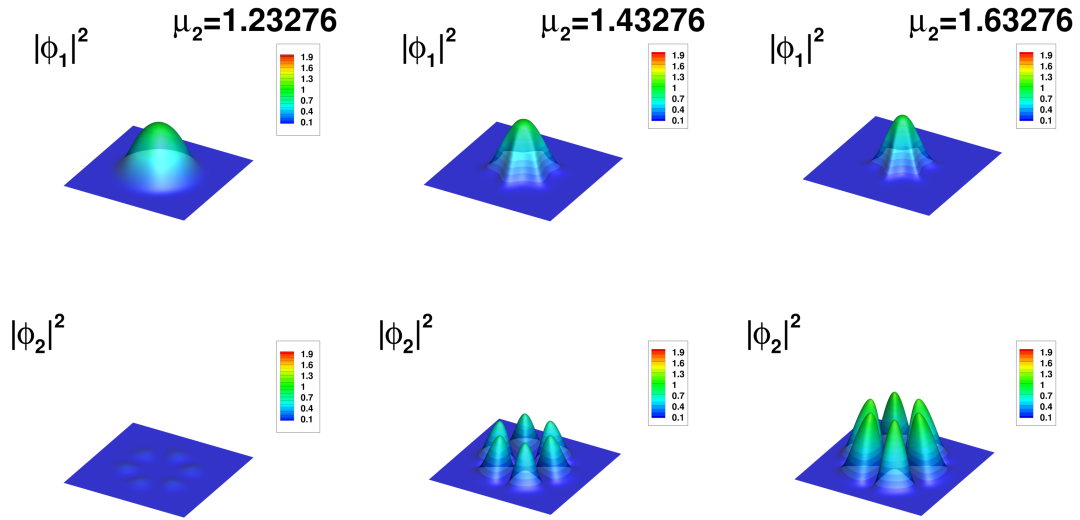


Figure 12: 2D two-component ground-state-soliton-necklace case. Same caption and computational domain as in Fig. 11. As the chemical potential μ_2 increases, the imprint of ϕ_2 on ϕ_1 becomes more apparent (see the third column).

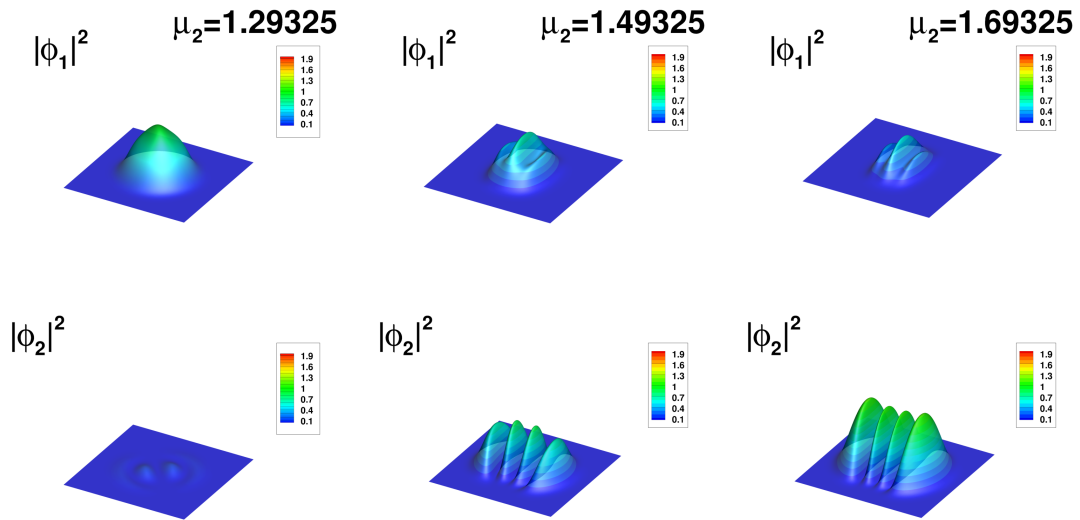
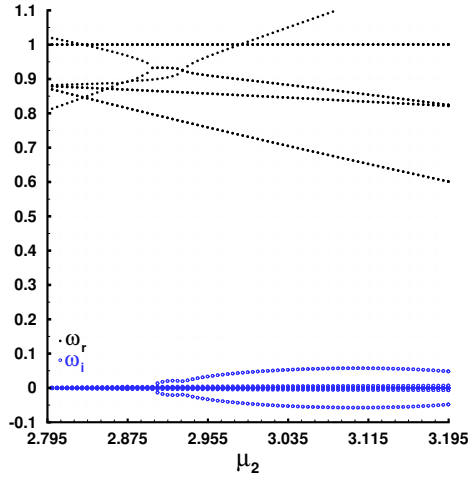


Figure 13: 2D two-component ground-state-multipole case. Same caption and computational domain as in Fig. 11. The ground state is depicted by ϕ_1 and the multipole state by ϕ_2 .

a) eigenvalues for dark-bright soliton stripe



b) eigenvalues for vortex-ring-bright state

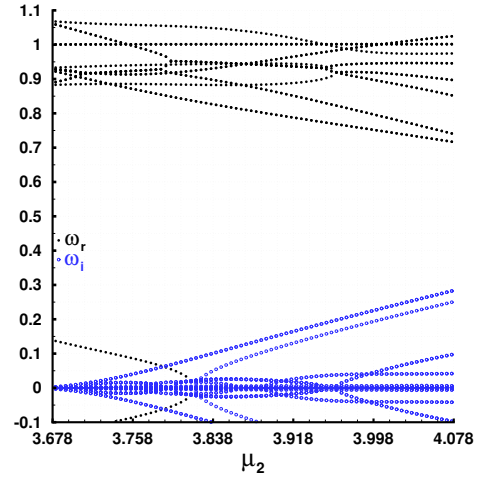


Figure 14: 3D two-component cases. The BdG spectra over μ_2 for a) the dark-bright soliton stripe (with the ground state in ϕ_1 and the planar dark soliton in ϕ_2), and b) vortex-ring-bright state (with the vortex-ring state in ϕ_2). These states bifurcate from $(\mu_1, \mu_2) = (2, 2.79467)$ and $(2, 3.67602)$, respectively. Their density profiles (for distinct values of μ_2) are shown in Figs. 15 and 16.

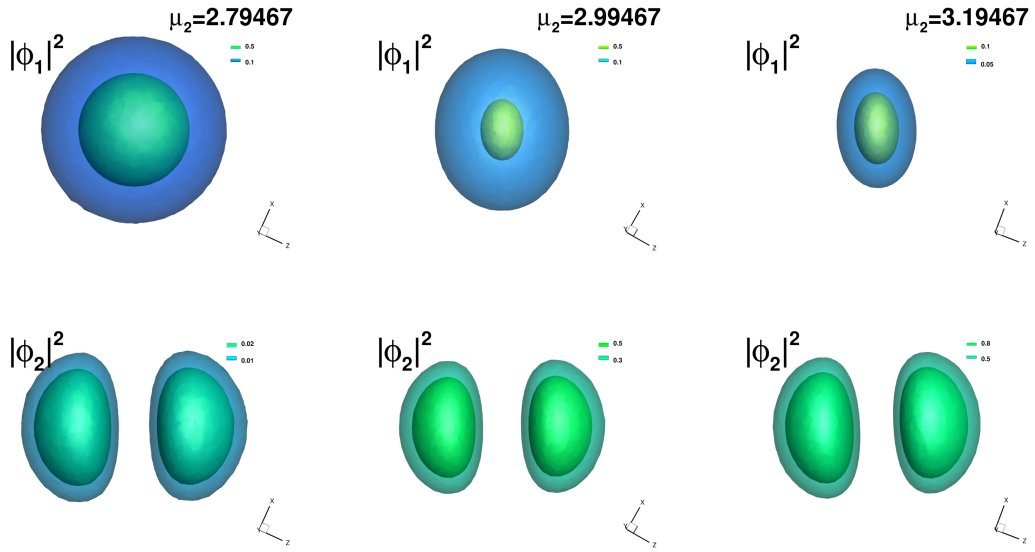


Figure 15: 3D two-component dark-bright soliton stripe state. Density profiles $|\phi_1|^2$ and $|\phi_2|^2$ are shown in the top and bottom rows, respectively, and for different values of μ_2 . Note that the ϕ_1 component carries the ground state whereas the ϕ_2 one the planar dark soliton (with a zero cut in the y direction). The computational domain was the cube $[-4.4, 4.4]^3$, and the above profiles are zoom-ins in $[-2, 2]^3$.

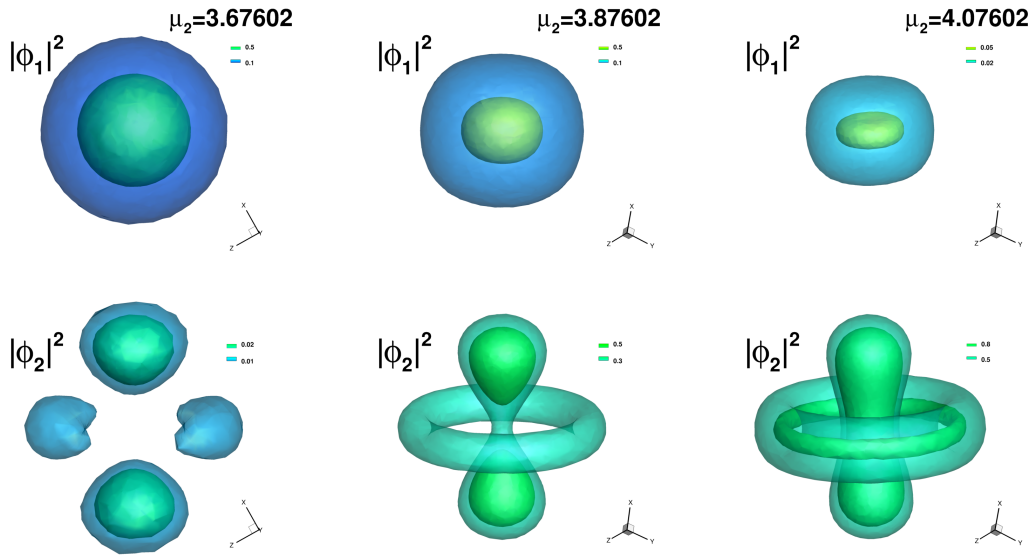


Figure 16: 3D two-component vortex-ring-bright state. Same caption and computational domain as in Fig. 15. Note that the ϕ_1 component carries (again) the ground state whereas the ϕ_2 one the vortex-ring state.

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_toolbox` 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 Tecplot³, Paraview⁴, and Gnuplot⁵. 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` directory is the following (the architecture of the `BdG_2comp_ddm` directory is almost identical):

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<https://tecplot.com>

⁴<https://www.paraview.org>

⁵<http://www.gnuplot.info>

3. *FFEM_LL_\$case_ddm.edp*: the main script for solving the eigenvalue problem of Eq. (39) (*i. e.* for finding the eigenvalue-eigenvector pairs (μ_2, ϕ_2)).
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. (33) and (35).
- *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. (24) and (28)-(31)).
- *Macro_LLsolve.edp*: macro for computing the eigenvalues of Eq. (39).
- *Macro_meshAdapt.edp*: macro for adapting the mesh to the wave function.
- *Macro_onedomainisol.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 (5), chemical potential, Hermite polynomials, etc. Also contains a macro creating 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 "k" 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. (24) or (35)], the BdG problems [cf. Eqs. (33) or (28)-(31)] and the linear limit problem [cf. Eq. (39)].
- *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).
- **await**: 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 adaptated 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. (32).
- **tolNewton**: the tolerance value ϵ_q in Eq. (32).
- **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**: 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 $\beta_{12} < \text{mubetaL}$ 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 no.

- **NNZ**: contain the non zero elements (nnz) for the BdG matrix.
- **dmuk**: counter for dm μ adaptation.
- **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 $\mathbf{Rdom} = \mathbf{scaledom} \cdot r_{\text{TF}}$, 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. (24) and (28)-(31).
 - **BCBdG**: the boundary conditions used in the BdG code for Eqs. (33) and (35).
 - **BCLL**: the boundary conditions used in the LL code for Eqs. (39).
 - **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:
 - **mulrestart**, **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 parenthesis):

1. The `Gnuplot` folder contains two files:
 - Information about the stationary states are 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. (32), the computed value of the chemical potential (computed values of μ_1 and μ_2), the number of atoms (6) (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` that contain 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 quite 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 \pm quadratic trapping ones (Bretin et al., 2004), and nonlinearities, such as the non-local ones appearing in dipolar settings, *e.g.* see (Tang et al., 2022).

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 Doedel and Tuckerman (2000); Seydel (2010), asymptotic numerical method (ANM) Ventura et al. (2020), and deflation-based techniques (Charalampidis et al., 2018, 2020; Boullé et al., 2020, 2023), among many others. Another possibility concerns about the interfacing of other libraries for eigenvalue computations, including the FEAST eigenvalue solver (Polizzi, 2009) which enjoys multiple levels of parallelization (Polizzi, 2023). Finally, with the recent experimental developments on spinor condensates (Xiao et al., 2021, 2022) described by more than two GP equations (see, *e.g.* Mithun et al. (2022) 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.

Acknowledgments

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. 6.3. They also thank Dr. V. Kalt for his insights and earlier collaboration.

References

- 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.
- 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.
- C. Pethick, H. Smith, *Bose-Einstein condensation in Dilute Gases*, Cambridge University Press, 2011.
- L. P. Pitaevskii, S. Stringari, *Bose-Einstein condensation and Superfluidity*, Oxford University Press, 2015.
- A. L. Fetter, A. A. Svidzinsky, Vortices in a trapped dilute bose-einstein condensate, *Journal of Physics: Condensed Matter* 13 (2001) R135.
- 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.
- 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.
- A. L. Fetter, Rotating trapped bose-einstein condensates, *Rev. Mod. Phys.* 81 (2009) 647–691.
- 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.

- 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.
- 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, *Nature Physics* 4 (2008) 496–501.
- K. W. Madison, F. Chevy, W. Wohlleben, J. Dalibard, Vortex formation in a stirred bose-einstein condensate, *Phys. Rev. Lett.* 84 (2000) 806–809.
- 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) 210403–210407.
- 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.
- 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.
- 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.
- 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.
- A. Aftalion, I. Danaila, Three-dimensional vortex configurations in a rotating Bose-Einstein condensate, *Physical Review A* 68 (2003) 023603.
- 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.
- B. A. Malomed, (invited) vortex solitons: Old results and new perspectives, *Physica D: Nonlinear Phenomena* 399 (2019) 108–137.
- L.-C. Crasovan, V. M. Pérez-García, I. Danaila, D. Mihalache, L. Torner, Three-dimensional parallel vortex rings in Bose-Einstein condensates, *Physical Review A* 70 (2004) 033605(1–5).
- 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.

- 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.
- F. H. V. Kalt, G. SADAHA, I. Danaila, Identification of vortices in quantum fluids: Finite element, *Computer Physics Communications* 284 (2023) 108606.
- 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.
- 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.
- 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.
- I. Danaila, M. A. Khamehchi, V. Gokhroo, P. Engels, P. G. Kevrekidis, Vector dark-antidark solitary waves in multicomponent Bose-Einstein condensates, *Physical Review A* 94 (2016) 053617.
- W. Wang, P. G. Kevrekidis, Two-component dark-bright solitons in three-dimensional atomic Bose-Einstein condensates, *Phys. Rev. E* 95 (2017) 032201.
- J. Ruostekoski, J. R. Anglin, Creating vortex rings and three-dimensional skyrmions in bose-einstein condensates, *Phys. Rev. Lett.* 86 (2001) 3934–3937.
- R. A. Battye, N. R. Cooper, P. M. Sutcliffe, Stable skyrmions in two-component bose-einstein condensates, *Phys. Rev. Lett.* 88 (2002) 080401.
- J. Ruostekoski, J. R. Anglin, Monopole core instability and alice rings in spinor bose-einstein condensates, *Phys. Rev. Lett.* 91 (2003) 190402.
- 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.
- E. Charalampidis, N. Boullé, P. Farrell, P. Kevrekidis, Bifurcation analysis of stationary solutions of two-dimensional coupled Gross-Pitaevskii equations using deflated continuation, *Communications in Nonlinear Science and Numerical Simulation* 87 (2020) 105255.
- 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.

- T. Kapitula, K. Promislow, Spectral and Dynamical Stability of Nonlinear Waves, Springer, 2013.
- N. N. Bogolyubov, On the theory of superfluidity, J. Phys. (USSR) 11 (1947) 23–32.
- P. G. De Gennes, Superconductivity of Metals and Alloys, CRC Press, 1966.
- C. M. Dion, E. Cancès, Ground state of the time-independent Gross-Pitaevskii equation, Comput. Phys. Comm. 177 (2007) 787–798.
- M. Caliori, S. Rainer, GSGPEs: A Matlab code for computing the ground state of systems of Gross-Pitaevskii equations, Comput. Phys. Comm. 184 (2013) 812 – 823.
- X. Antoine, R. Duboscq, GPELab, a Matlab toolbox to solve Gross-Pitaevskii equations I: Computation of stationary solutions, Comput. Phys. Comm. 185 (2014) 2969–2991.
- Z. Marojević, E. Göklü, C. Lämmerzahl, ATUS-PRO: A FEM-based solver for the time-dependent and stationary Gross-Pitaevskii equation, Computer Physics Communications 202 (2016) 216 – 232.
- G. Vergez, I. Danaila, S. Auliac, F. Hecht, A finite-element toolbox for the stationary Gross-Pitaevskii equation with rotation, Comput. Phys. Comm. 209 (2016) 144–162.
- H. Uecker, D. Wetzel, J. Rademacher, pde2path - A Matlab Package for Continuation and Bifurcation in 2D Elliptic Systems, Num. Math.: Theory, Methods and Applications 7 (2014) 58–106.
- P. Muruganandam, S. Adhikari, Fortran programs for the time-dependent Gross-Pitaevskii equation in a fully anisotropic trap, Comput. Phys. Comm. 180 (2009) 1888–1912.
- 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. Comm. 183 (2012) 2021 – 2025.
- L. Kong, J. Hong, J. Zhang, LOD-ms for Gross-Pitaevskii equation in Bose-Einstein condensates, Communications in Computational Physics 14 (2014) 219–241.
- U. Hohenester, OCTBEC a Matlab toolbox for optimal quantum control of Bose-Einstein condensates, Comput. Phys. Comm. 185 (2014) 194–216.
- R. Kishor Kumar, V. Lončar, P. Muruganandam, S. K. Adhikari, A. Balaž, C and Fortran OpenMP programs for rotating Bose-Einstein condensates, Computer Physics Communications 240 (2019) 74 – 82.

- E. L. Allgower, K. Georg, Spectral and Dynamical Stability of Nonlinear Waves, Springer-Verlag, 1990.
- C. T. Kelley, Solving Nonlinear Equations with Newton's Method, Society of Industrial and Applied Mathematics, Philadelphia, 2003.
- R. Carretero-González, P. Kevrekidis, T. Kolokolnikov, Vortex nucleation in a dissipative variant of the nonlinear Schrödinger equation under rotation, *Physica D: Nonlinear Phenomena* 317 (2016) 1–14.
- 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.
- A. Roy, S. Pal, S. Gautam, D. Angom, P. Muruganandam, FACT: FORTRAN toolbox for calculating fluctuations in atomic condensates, *Computer Physics Communications* 256 (2020) 107288.
- G. SADAKA, V. Kalt, F. Hecht, I. Danaila, A finite element toolbox for the bogoliubov-de gennes stability analysis of bose-einstein condensates, *Computer Physics Communications* 294 (2024) 108948.
- F. Hecht, New developments in Freefem++, *Journal of Numerical Mathematics* 20 (2012) 251–266.
- S. Balay, et. al., PETSc/TAO Users Manual, Argonne National Laboratory-21/39, <https://petsc.org/release/docs/manual/manual.pdf>, 2022.
- P. Jolivet, Introduction to FreeFEM with an emphasis on parallel computing, <https://joliv.et/FreeFem-tutorial/main.pdf>, 2023.
- 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.
- P.-H. Tournier, P. Jolivet, F. Nataf, FFDDM: FreeFem Domain Decomposition Method, <https://doc.freefem.org/documentation/ffddm/index.html>, 2019.
- 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.
- D. J. Frantzeskakis, Dark solitons in atomic Bose-Einstein condensates: from theory to experiments, *Journal of Physics A: Mathematical and Theoretical* 43 (2010) 213001.

- W. Bao, Y. Cai, Mathematical theory and numerical methods for Bose-Einstein condensation, *Kinetic and Related Models* 6 (2013) 1–135.
- R. A. Adams, J. J. F. Fournier, *Sobolev Spaces*, Academic Press, 2003.
- 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, Mississippi State Univ., 1996.
- P. J. Frey, P. L. George, *Maillages*, Hermès, Paris, 1999.
- B. Mohammadi, O. Pironneau, *Applied Shape Design for Fluids*, Oxford Univ. Press, 2000.
- C. Dapogny, C. Dobrzynski, P. Frey, Three-dimensional adaptive domain remeshing, implicit domain meshing, and applications to free and moving boundary problems, *Journal of Computational Physics* 262 (2014) 358–378.
- I. Danaila, F. Hecht, A finite element method with mesh adaptivity for computing vortex states in fast-rotating Bose-Einstein condensates, *J. Comput. Physics* 229 (2010) 6946–6960.
- 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.
- E. Charalampidis, P. Kevrekidis, P. Farrell, Computing stationary solutions of the two-dimensional gross-pitaevskii equation with deflated continuation, *Communications in Nonlinear Science and Numerical Simulation* 54 (2018) 482–499.
- 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.
- 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.
- V. Bretin, S. Stock, Y. Seurin, J. Dalibard, Fast rotation of a Bose-Einstein condensate, *Phys. Rev. Lett.* 92 (2004) 050403.
- 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 Journal on Scientific Computing* 44 (2022) B100–B121.
- 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.

- R. Seydel, Practical Bifurcation and Stability Analysis, volume 5, Interdisciplinary Applied Mathematics, Springer-Verlag (New York), 2010.
- P. Ventura, M. P. Ferry, H. Zahrouni, A secure version of asymptotic numerical method via convergence acceleration, *Compt. Rend. Mecan.* 348 (2020) 361–374.
- E. Polizzi, Density-matrix-based algorithm for solving eigenvalue problems, *Phys. Rev. B* 79 (2009) 115112.
- E. Polizzi, FEAST Eigenvalue Solver, <https://www.feast-solver.org/index.htm>, 2023.
- 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, *Communications Physics* 4 (2021).
- Y. Xiao, M. O. Borgh, A. A. Blinova, T. Ollikainen, J. Ruostekoski, D. S. Hall, Topological superfluid defects with discrete point group symmetries, *Nature Communications* 13 (2022).