Identification of vortices in quantum fluids: finite element algorithms and programs

Victor Kalt[1], Georges Sadaka[1], Ionut Danaila[1*], Frédéric Hecht[2]

[1]Univ Rouen Normandie, CNRS UMR 6085, Laboratoire de Mathématiques Raphaël Salem,
Avenue de l'Université, BP 12, F-76801 Saint-Étienne-du-Rouvray, France
[2]Sorbonne Université, CNRS UMR 7598, Laboratoire Jacques-Louis Lions, F-75005, Paris, France.
*Corresponding author. Tel.: (+33) 2 32 95 52 50

Abstract

We present finite-element numerical algorithms for the identification of vortices in quantum fluids described by a macroscopic complex wave function. Their implementation using the free software FreeFem++ is distributed with this paper as a post-processing toolbox that can be used to analyse numerical or experimental data. Applications for Bose-Einstein condensates (BEC) and superfluid helium flows are presented. Programs are tested and validated using either numerical data obtained by solving the Gross-Pitaevskii equation or experimental images of rotating BEC. Vortex positions are computed as topological defects (zeros) of the wave function when numerical data are used. For experimental images, we compute vortex positions as local minima of the atomic density, extracted after a simple image processing. Once vortex centers are identified, we use a fit with a Gaussian to precisely estimate vortex radius. For vortex lattices, the lattice parameter (inter-vortex distance) is also computed. The post-processing toolbox offers a complete description of vortex configurations in superfluids. Tests for two-dimensional (giant vortex in rotating BEC, Abrikosov vortex lattice in experimental BEC) and three-dimensional (vortex rings, Kelvin waves and quantum turbulence fields in superfluid helium) configurations show the robustness of the software. The communication with programs providing the numerical or experimental wave function field is simple and intuitive. The post-processing toolbox can be also applied for the identification of vortices in superconductors.

Email— victor.kalt@univ-rouen.fr, georges.sadaka@univ-rouen.fr, ionut.danaila@univ-rouen.fr, frederic.hecht@sorbonne-universite.fr

Programm summary
Program Title: FFEM_postproc_data_2D, FFEM_postproc_data_3D and FFEM_postproc_image
Catalogue identifier:
No. of lines in distributed program, including test data, etc.: 81429
No. of bytes in distributed program, including test data, etc.: 8 768 685
Distribution format: .zip
Programming language: FreeFem++ (v 4.12) free software (www.freefem.org)
Computer: PC, Mac, Super-computer.
Operating system: Mac OS, Linux, Windows.
Nature of problem: The software is scoped to the identification of quantized vortices in 2D or 3D configurations described by a complex wave function. Either simulation data obtained through numerical resolution of the Gross-Pitaevskii equation or 2D experimental images of Bose-Einstein condensates (BEC) can be used as input data. Examples of vortex identification in rotating BECs and superfluid helium are illustrated in the paper.
Solution method: We model the complex wave function using P1 (piece-wise linear) Galerkin triangular (in 2D) and tetrahedral (in 3D) finite elements. For simulation data, the zeros of the wave function are directly computed and the circulation of the velocity on triangles sides is assessed. Vortices are obtained when a zero is found in a triangle with non-zero circulation. In 3D, the algorithm is applied on tetrahedron faces and the zero points are linked using mesh connectivity to form vortex lines. For experimental images, vortices are identified as local minima of the atomic density and extracted using the image contrast.
Running time: From seconds to minutes depending on the mesh resolution of the simulation/experimental data.

1 Introduction

The nucleation of quantized vortices, with fixed (quantized) circulation and fixed core diameter is a striking feature of the two main known quantum fluids: superfluid helium and atomic Bose-Einstein condensate (BEC). Liquid helium below the critical (lambda) temperature $T_\lambda = 2.17K$ is also called He II and is generally described as a mixture of two fluids with independent velocity fields: a *normal* viscous fluid and an inviscid *superfluid*. In the superfluid component, a large number of quantized vortices of the atomic size are nucleated and their complex interactions lead to a Quantum Turbulence (QT) state. The vortex line density plays an important role in the statistical description of QT and is usually measured in experiments using second sound probes (Donnelly, 2009). Visualisations of vortices in superfluid helium are relatively recent and offer only a qualitative picture of vortex reconnections (Bewley et al., 2006). Therefore, considerable theoretical and numerical efforts (see dedicated reviews or volumes Skrbek and Sreenivasan (2012); Barenghi et al. (2014); Tsubota et al. (2017)) were devoted to unravel properties of vortex interactions in He II. The Gross-Pitaevskii (GP) model has been extensively used to explore properties of QT in an ideal setting containing only the superfluid (Nore et al., 1997; Abid et al., 2003; Kobayashi et al., 2021). Even though the GP equation offers only a partial description of the complexity of superfluid helium, quantum vortex interactions are accurately described without any additional phenomenological models.

The extraction of vortices from a GP numerical dataset providing values of the complex macroscopic wave function (order parameter) ψ is thus an important tool to study properties of QT turbulence in superfluid helium. The main difficulty in this setting is the existence of a very large number of vortices with tangled shapes resulting from multiple reconnections. The identification of vortices is however simplified by the fact that vortices are evolving into a uniform background quantum flow of constant atomic density $n = |\psi|^2 = n_0$. One the main applications of the algorithms and programs presented in this paper is the extraction of multiple vortex lines from a given complex wave function field ψ, resulting from numerical simulations of QT based on the GP model.

Quantized vortices have been extensively studied in BECs, since their existence provides an evidence of the superfluid nature of these systems. Different vortex generation methods were used in experimental BEC setups: based on the drag on an object moving through the condensates (Jackson et al., 1999), by rotating the trap confining the atoms (Madison et al., 2000; Haljan et al., 2001), or by phase-engineering of topological defects (Leanhardt et al., 2002). Experimental observations were systematically supplemented with numerical simulations based on the GP equation, which is in this case (dilute BEC at zero temperature) the standard mathematical model. Different types of vortices have been observed in experiments or numerical simulations of BEC: Abrikosov triangular vortex lattices (Engels et al., 2003), bent vortex lines of "U" or "S" shapes (Rosenbusch et al., 2002a; Aftalion and Danaila, 2003), giant vortices (Bretin et al., 2004; Aftalion and Danaila, 2004; Danaila, 2005) or more exotic shapes of vortices (double rings, hopfions, stars, etc.) (Bisset et al., 2015; Crasovan et al., 2004). Experimental images of quantized vortices in BEC are typically obtained by switching-off the magnetic trap and imaging (after a time of flight) the absorption of a resonant laser beam propagating along the z-axis. The obtained pictures, showing atomic density integrated along the z-axis, are then used to extract the size and the average position of the condensate (Rosenbusch et al., 2002b). Note that vortex cores are larger in BEC ($5 \cdot 10^{-7} m$) than in superfluid helium ($10^{-10} m$) and thus more easily observable. Note also that recent experimental and theoretical studies (Kwon et al., 2014; Navon et al., 2016) reported different possible routes to QT in BEC.

Another application of the algorithms and programs presented in this contribution is the possibility to accurately post-process BEC experimental images, as well as wave function fields provided by GP numerical simulations of BEC. The difficulty in the BEC case is that vortices are topological defects (zeros of the wave function) on the top of a non-constant atomic density that also vanishes outside the condensate. This background density depends on the BEC setting (trapping potential, rotation) and is taken into account in the algorithm extracting vortices. The post-processing thus separates the non-condensed cloud from the condensate, extracts the background atomic density to isolate vortices and finally fits each individual vortex with an anisotropic Gaussian to compute vortex radius.

Even though the present toolbox can be easily used with experimental images of quantum systems with vortices (superfluids, superconductors), the main intended application is the link with numerical systems providing a 2D or 3D field of the complex wave function ψ. Numerical results of vortices in superfluids are generally presented using contours of the atomic density in 2D simulations or iso-surfaces of low atomic density in 3D simulations. Precise methods for the tracking of vortices in complex fields were developed by Villois et al. (2016) using a Newton-Raphson method to determine the zeros of the wave function combined with Fourier interpolation for the evaluation of the wave function outside grid points. Caliari and Zuccher (2017) suggested another method adapted to spectral Fourier computations, based on non-equispaced Fast Fourier Transform to truncate Fourier series at arbitrary points and finally identify vortex line positions during vortex reconnection. A fast real-time visualisation technique of vortices in QT simulations of high resolution simulations was developed by Liu et al. (2020) using a graph-based method.

The advantage of using the present toolbox is to use generic unstructured grids with finite elements in 2D and 3D that can be subsequently refined around vortices. Adapted mesh provides accurate identification of vortex centers, vortex lines description and vortex radius computation. These quantities are important in assessing physical and mathematical theories on vortex dynamics or QT in BEC or superfluid helium. A great number of software packages for solving different forms (stationary, time-dependent, with non-local or high-order interactions) of the GP equation exist in the CPC Program Library. Spatial discretizations are generally based on spectral (Dion and Cancès, 2007; Caliari and Rainer, 2013; Antoine and Duboscq, 2014, 2015), finite-elements (Marojević et al., 2016; Vergez et al., 2016) or finite-difference (Muruganandam and Adhikari, 2009; Vudragović et al., 2012; Caplan, 2013; Kong et al., 2014; Hohenester, 2014; Kishor Kumar et al., 2019) methods. Provided programs are written in Fortran (Dion and Cancès, 2007; Muruganandam and Adhikari, 2009), C (Vudragović et al., 2012; Caplan, 2013), Matlab (Caliari and Rainer, 2013; Caplan, 2013; Antoine and Duboscq, 2014; Hohenester, 2014; Antoine and Duboscq, 2015), FreeFem++ (Vergez et al., 2016) or C and Fortran with OpenMP (Kishor Kumar et al., 2019). All these programs could benefit from the use of the present post-processing toolbox that can be easily interfaced with any type of space discretization by writing the output complex wave function fields in a standard finite-element format (examples of how to write data are provided in Sect. 7).

The organization of the paper is as follows. In Sect. 2 we introduce the GPE model and the characterisation of quantized vortices. Sections 3 and 4 present the main algorithms used for the identification of vortices in GP simulations of BECs for 2D and 3D configurations, respectively. Section 5 is dedicated to the identification of vortices in experimental images. Programs architecture and main parameters of the toolbox are described in Sect. 7. We present various 2D (Abrikosov vortex lattice, giant vortex) and 3D (bent vortex, vortex knot, Kelvin waves and vortices in QT) benchmarks used for the validation of our codes in Sect. 6. Finally, the main features of the toolbox and its possible extensions are summarized in Sect. 8.

2 The Gross-Pitaevskii model and quantized vortices

2.1 The Gross-Pitaevskii equation

The Gross-Pitaevskii equation describes the time-space evolution of the macroscopic wave function of a dilute gaseous BEC in the limit of zero temperature. For a BEC of N atoms confined in a trapping potential $V_{\text{trap}}(\boldsymbol{x})$ and rotating with angular velocity $\boldsymbol{\Omega}$, the GP equation is:

$$i\hbar\frac{\partial \psi}{\partial t} = -\frac{\hbar^2}{2m}\nabla^2\psi + V_{\text{trap}}\psi + g|\psi|^2\psi - \boldsymbol{\Omega}\cdot\boldsymbol{\mathcal{L}}(\psi), \tag{1}$$

where $\boldsymbol{\mathcal{L}}$ is the angular momentum, $\hbar$ the reduced Planck constant and m the atomic mass. The constant in front of the non-linear term depends on the scattering length a_s, as $g = \frac{4\pi\hbar^2 a_s}{m}$. The atomic density $n(\boldsymbol{x}) = |\psi(\boldsymbol{x})|^2$ vanishes outside the condensate and thus natural boundary conditions for (1) are homogeneous Dirichlet conditions ($\psi(\boldsymbol{x}) \to 0$ as $\boldsymbol{x} \to \infty$). When Eq. (1) is used to describe superfluid helium, the trapping potential and the angular velocity are set to zero and periodic boundary conditions are assumed. Considering rotations along the z-axis (*i.e.* $\boldsymbol{\Omega} = \Omega\,\boldsymbol{k}$) implies that only the z-component of the angular momentum appears in Eq. (1) and the rotation term can be presented as (Danaila and Kazemi, 2010; Danaila and Hecht, 2010; Danaila and Protas, 2017):

$$\boldsymbol{\Omega}\cdot\boldsymbol{\mathcal{L}}(\psi) = \Omega\mathcal{L}_z\psi = i\hbar\Omega\left(y\frac{\partial \psi}{\partial x} - x\frac{\partial \psi}{\partial y}\right) = i\hbar\Omega\,A^t\nabla\psi, \quad \text{with} \quad A^t = (y, -x, 0). \tag{2}$$

The following general *quadratic $\pm$ quartic* form of the trapping potential describes most of the existing theoretical and experimental studies of rotating BECs (Vergez et al., 2016):

$$V_{\text{trap}}(x, y, z) = \frac{m}{2}\left(\omega_x^2 x^2 + \omega_y^2 y^2 + \omega_z^2 z^2\right) + U_2\left(\frac{r}{w_2}\right)^2 + U_4\left(\frac{r}{w_4}\right)^4, r^2 = x^2 + y^2, \tag{3}$$

where $\omega_x, \omega_x, \omega_x$ are trapping frequencies of the classical harmonic ($\sim r^2$) trapping potential and U_2, U_4, w_2, w_4 are characteristics of the detuned laser beam adding the quartic part ($\sim r^4$) of the potential and a supplementary (positive or negative) quadratic term (Bretin et al., 2004). The conservation of the number of atoms N inside the condensate is expressed as:

$$\int_{\mathbb{R}^3} |\psi(\boldsymbol{x})|^2\,d\boldsymbol{x} = N. \tag{4}$$

2.2 The Thomas-Fermi approximation

The Thomas-Fermi regime is characterized by strong interactions (the kinetic energy is negligible compared to the interaction energy). In absence of rotation, the Thomas-Fermi density is classically (Pitaevskii and Stringari, 2003) obtained from Eq. (1) by considering stationary solutions $\psi(\boldsymbol{x}, t) = \phi(\boldsymbol{x}) \exp(-i\mu t/\hbar)$ and neglecting the Laplacian term (corresponding to kinetic energy):

$$V_{\text{trap}}(\boldsymbol{x}) + g|\phi|^2 = \mu. \tag{5}$$

The corresponding Thomas-Fermi atomic density then takes a simple form:

$$n_{\text{TF}}(\boldsymbol{x}) = |\phi_{\text{TF}}|^2 = \frac{\mu - V_{\text{trap}}(\boldsymbol{x})}{g}. \tag{6}$$

The constant μ is the chemical potential and its value can be calculated by combining Eqs. (6) and (4). If $\Omega \neq 0$, an equivalent Thomas-Fermi approximation can be used to approximate the density of the rotating condensate by considering (6), but with an effective trapping potential:

$$V_{\text{trap}}^{\text{eff}} = V_{\text{trap}} - \frac{1}{2}m\Omega^2 r^2, \quad r^2 = x^2 + y^2. \tag{7}$$

The effective potential is the original potential diminished by the centrifugal term (Stringari, 1999; Vergez et al., 2016). In the case of superfluid helium ($V_{\text{trap}} = \Omega = 0$) the Thomas-Fermi density is constant $n_{\text{TF}} = n_0$.

2.3 Quantized vortices

Using the Madelung transformation, the wave function can be also presented as:

$$\psi(\boldsymbol{x}, t) = \sqrt{n(\boldsymbol{x}, t)} e^{iS(\boldsymbol{x}, t)}, \tag{8}$$

with n the atomic density and S the phase. The mass density ρ and the velocity $\boldsymbol{v}$ of the superfluid are defined as:

$$\rho(\boldsymbol{x}, t) = m\, n(\boldsymbol{x}, t) = m\, |\psi(\boldsymbol{x}, t)|^2, \tag{9}$$

$$\boldsymbol{v}(\boldsymbol{x}, t) = \frac{\hbar}{m}\nabla S(\boldsymbol{x}, t) = \frac{\hbar}{\rho}\frac{\psi^*\nabla\psi - \psi\nabla\psi^*}{2i}. \tag{10}$$

For a flow of non-vanishing density ($\rho \neq 0$), we infer from (10) that the superfluid is irrotational:

$$\nabla \times \boldsymbol{v} = \frac{\hbar}{m}\nabla \times \nabla S = 0. \tag{11}$$

The lines along which $\rho = 0$ define topological defects, known as *quantized vortices*. It results from (10) that the velocity is singular at the vortex center. The Madelung transformation becomes singular when vortices are present in the flow. For a review of mathematical problems related to the Madelung transformation in presence of quantum vortices, see Carles et al. (2012).

The circulation Γ along a closed regular path C surrounding a simply connected domain S is:

$$\Gamma = \oint_C \boldsymbol{v} \cdot d\boldsymbol{l} = \frac{\hbar}{m}\oint_C \nabla S \cdot d\boldsymbol{l}. \tag{12}$$

If S is a simply connected domain without vortices, using property (11) and Stokes' theorem results in $\Gamma = 0$. If a vortex is present in the domain S, the curl of the velocity expressed by Eq. (11) becomes a Dirac function and a phase-defect line starting from the vortex center exists. Since ψ must be a single-valued function, the integrated change in the phase must be a multiple of 2π, which implies that:

$$\Gamma = \frac{\hbar}{m}2\pi\kappa = \frac{h}{m}\kappa, \tag{13}$$

where κ is an integer representing the winding number of the vortex. It is important to note that vortex solutions are not singular solutions to the GP equation (1). Indeed, a straight-line vortex solution Ψ_v can be obtained (Pitaevskii and Stringari, 2003) using cylindrical coordinates (z, r, φ) as:

$$\Psi_v = \sqrt{n_0}\, f(r)\, e^{i\kappa\varphi}. \tag{14}$$

The asymptotic behaviour of this solution near the origin ($r = 0$) is known (Neu, 1990):

$$f(r) \sim r^{|\kappa|} + \mathcal{O}(r^{|\kappa|+2}), \quad r \to 0. \tag{15}$$

The region near the vortex center where the density is varying in a significant way defines the vortex core. The radius of the vortex is of the order of the healing length ξ (Pitaevskii and Stringari, 2003):

$$\xi = \frac{\hbar}{\sqrt{2mg|\psi|^2}} = \frac{1}{\sqrt{8\pi a_s |\psi|^2}}. \tag{16}$$

This description of quantized vortices offers a very simple practical way to extract vortices from a GP wave function field. Vortices are thus identified in the present post-processing as points in 2D or lines in 3D characterized by the following two properties:

- The density at the vortex center is 0.
- The circulation of the velocity on a closed path around the center is non zero.

2.4 Scaling

Numerical simulations are usually based on the non-dimensional form of the GP equation (1). By setting

$$\boldsymbol{x} \to \frac{\boldsymbol{x}}{a_{\text{ho}}}, \quad u = \frac{\psi}{\sqrt{N}\, a_{\text{ho}}^{-d/2}}, \tag{17}$$

where $d = 2$ or 3 is the dimension of the space and $a_{\text{ho}} = \sqrt{\frac{\hbar}{m\omega_\perp}}$ the harmonic oscillator length defined with respect to a reference trapping frequency $\omega_\perp = \min(\omega_x, \omega_y)$, the dimensionless form of the GP equation becomes:

$$i\frac{\partial u}{\partial t} = -\frac{1}{2}\nabla^2 u + C_{\text{trap}}u + C_g|u|^2 u - iC_\Omega A^t \nabla u. \tag{18}$$

Coefficients in (18) are defined as:

$$C_\Omega = \left(\frac{\Omega}{\omega_\perp}\right), \quad C_g = \frac{4\pi N a_s}{a_{\text{ho}}} \quad \text{(in 3D)}, \quad C_g = \beta \quad \text{(given in 2D)}. \tag{19}$$

The dimensionless effective potential corresponding to (7) and (3) is :

$$V_{\text{trap}}^{\text{eff}} = \frac{1}{2}\left(a_x x^2 + a_y y^2 + a_z z^2 + a_4(x^2 + y^2)^2\right), \tag{20}$$

with dimensionless coefficients (Vergez et al., 2016):

$$\begin{cases} a_x = \left(\dfrac{\omega_x}{\omega_\perp}\right)^2 - \left(\dfrac{\Omega}{\omega_\perp}\right)^2 + 2\left(\dfrac{U_2}{m\omega_\perp^2 w_2^2}\right), \\[2mm] a_y = \left(\dfrac{\omega_y}{\omega_\perp}\right)^2 - \left(\dfrac{\Omega}{\omega_\perp}\right)^2 + 2\left(\dfrac{U_2}{m\omega_\perp^2 w_2^2}\right), \\[2mm] a_z = \left(\dfrac{\omega_z}{\omega_\perp}\right)^2, \quad a_4 = 2\left(\dfrac{U_4 a_{\text{ho}}^2}{m\omega_\perp^2 w_4^4}\right). \end{cases} \tag{21}$$

Non-dimensional forms (20) and (21) are used to obtain analytical forms for the Thomas-Fermi approximation (6):

$$\rho_{\text{TF}} = \mod u^2 = \left(\frac{\rho_0 - 2V_{\text{trap}}^{\text{eff}}}{2C_g}\right)_+. \tag{22}$$

Analytical expressions for ρ_0 are presented in detail by Vergez et al. (2016) for trapping potentials of the form *quartic $\pm$ quadratic*. Note that the non-dimensional healing length (16) becomes in the Thomas-Fermi approximation:

$$\frac{\xi}{a_{\text{ho}}} = \frac{1}{\sqrt{2C_g|u|^2}} \implies \frac{\xi_{\min}}{a_{\text{ho}}} = \frac{1}{\sqrt{\rho_0}}. \tag{23}$$

3 Vortex identification in 2D BECs

The vortex identification process in 2D is decomposed into two parts: the localization of the vortex center and the computation of vortex characteristics. For the latter step, the vortex core radius (obtained through a fitting procedure) and the parameter of the Abrikosov vortex lattice (when identified) are computed. The initial data is the complex wave function u obtained from numerical simulations and represented as a P1 (piece-wise linear) finite-element function on a triangular mesh Th. We illustrate the algorithms using numerical data reported by Vergez et al. (2016) for a fast rotating BEC with a *quartic-minus-quadratic* trapping potential. The initial data is shown in Fig. 1(a). A giant vortex is formed in the middle of the condensate, surrounded by a lattice of individual vortices. The purpose of the post-processing is to identify individual vortices and compute their characteristics.

3.1 Vortex localization

The main difficulty introduced by this case is that the wave function is vanishing inside the giant vortex and also outside the condensate. These zones have to be removed before searching the zeros of the wave function corresponding to vortices. There are two possibilities depending on the availability of numerical parameters used in simulations.

- If parameters (21) of the trapping potential are known, we use formulae presented by Vergez et al. (2016) to compute ρ_0 necessary to estimate the background density following the Thomas-Fermi approximation (22). For the case presented in this section the trapping potential (20) is of the form $V_{\text{trap}}^{\text{eff}} = \frac{1}{2}\left(a_2 r^2 + a_4 r^4\right)$ and we use the following algorithm:

$$
\begin{array}{l}
--> \text{compute } \rho_0 \\
\left[
\begin{array}{l}
\text{Compute} \quad C_{\text{S}} = 2C_g, \\[2mm]
\bullet \text{ if } a_2 < 0 \text{ and } a_4 < \sqrt{\dfrac{\pi a_2^3}{6C_{\text{S}}}} \\[3mm]
\rho_0 = \dfrac{1}{4a_4}\left[\left(\dfrac{6a_4^2}{\pi}C_{\text{S}}\right)^{2/3} - a_2^2\right] \\[4mm]
\bullet \text{ else} \\[2mm]
\text{calculate the root } \eta > 0 \text{ of:} \\[2mm]
f(\eta) = 4a_4\eta^3 + 3a_2\eta^2 - \dfrac{6}{\pi}\left(C_{\text{S}}\right) = 0, \\[3mm]
\text{and then calculate} \\[2mm]
\rho_0 = a_2 R^2 + a_4 R^4
\end{array}
\right. \\[2mm]
--> \text{compute the maximum radius of the condensate} \\[2mm]
R_+ = \left(\dfrac{-a_2 + \sqrt{a_2^2 + 4\rho_0 a_4}}{2a_4}\right)^{1/2}.
\end{array}
\tag{24}
$$

  At the end of this step, mesh triangles where $\rho_{\text{TF}} < 0$ are removed.

- If the analytical expression of the Thomas-Fermi approximation is not known, we use to identify the border of the condensate an iso-value line of very low density $\rho_{\min}$. A typical value is $\rho_{\min} = 5 \cdot 10^{-5}$, but this threshold can be adjusted depending on the type of the scaling used in the numerical simulation. Triangles where $\rho < \rho_{\min}$ are removed at the end of this step.

We can proceed now with vortex center localisation. The location of vortices can be obtained through a direct application of their defining properties mentioned at the end of Sect. 2.3. For each triangle T of the mesh, we search whether it contains a zero of the wave function or not. We denote by $u_r = \Re(u)$ and $u_i = \Im(u)$ the real and imaginary part of the wave function, respectively. Let P_0, P_1, P_2 be the vertices of the triangle. A necessary condition for the presence of a zero of a P1 complex function inside a triangle is that both real and imaginary part must change sign inside the triangle:
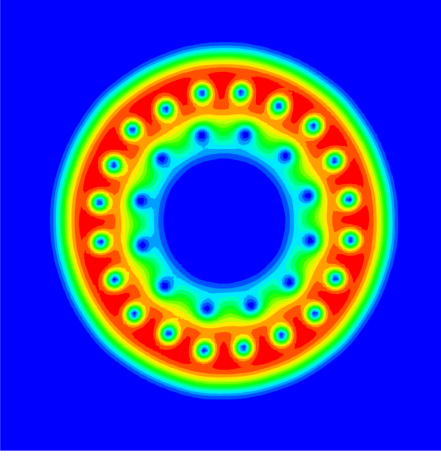
$$\min(u_r(P_0), u_r(P_1), u_r(P_2)) < 0, \tag{25}$$

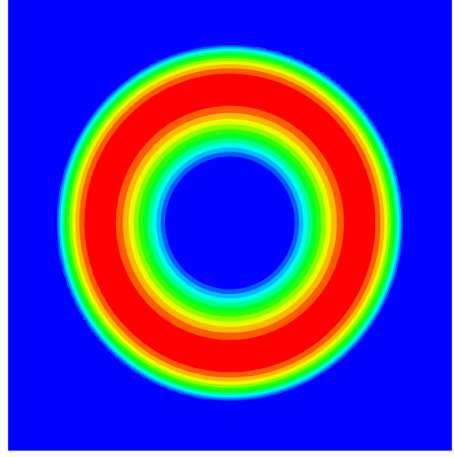$$\min(u_i(P_0), u_i(P_1), u_i(P_2)) < 0, \tag{26}$$

$$\max(u_r(P_0), u_r(P_1), u_r(P_2)) > 0, \tag{27}$$

$$\max(u_i(P_0), u_i(P_1), u_i(P_2)) > 0. \tag{28}$$

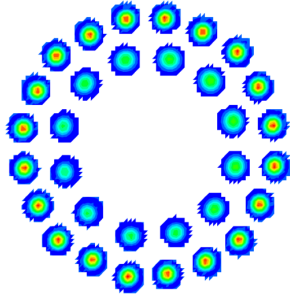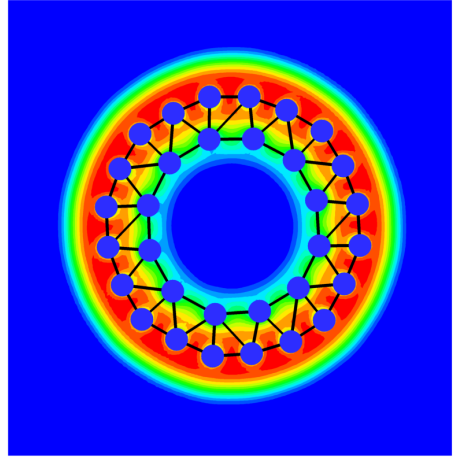*Figure 1: Illustration of vortex identification steps in 2D. Data from numerical simulation of a fast rotating BEC with a* quartic-minus-quadratic *trapping potential (Vergez et al., 2016). a) Initial density ρ, b) Thomas-Fermi density $ρ_{TF}$, c) identified vortex zones used to localize vortex centers and for the fitting with a Gaussian, d) mesh of the identified vortex lattice (blue points) on top of the initial data.*

The coordinates of the zero in barycentric coordinates are then computed as:

$$x_v = \frac{\det(u(P_0), 0, u(P_2))}{\det(u(P_0), u(P_1), u(P_2))}, \tag{29}$$

$$y_v = \frac{\det(u(P_0), u(P_1), 0)}{\det(u(P_0), u(P_1), u(P_2))}. \tag{30}$$

If a zero is found inside the triangle, we compute the circulation of the velocity on triangle sides to assess whether a vortex is present. After discretizing Eq. (12), we use the following formula to compute circulation and corresponding winding number:

$$\kappa = \frac{1}{2\pi} \Im \left(\log \left(\frac{u(P_1)}{u(P_0)} \right) + \log \left(\frac{u(P_2)}{u(P_1)} \right) + \log \left(\frac{u(P_0)}{u(P_2)} \right) \right), \tag{31}$$

where $\Im$ denotes the imaginary part. A winding number $\kappa = \pm 1$ indicates the presence of a vortex inside the triangle at position (x_v, y_v). Divisions by zero in Eq. (31) are avoided in programs by first checking the value of the wave function on the vertices of the triangle. This enables the algorithm to also capture vortices passing exactly through a vertex of the mesh.

3.2   Vortex radius estimation

The difficulty of this step is the separation of the vortex density from the background (Thomas-Fermi) density. We use the decomposition:

$$\rho = \rho_v \rho_b \tag{32}$$

with ρ_b is the background density corresponding to the shape of the condensate and ρ_v the vortex density varying between 0 and 1. The computation of ρ_b is done using two methods.

- If the parameters defining the trapping potential are known, we use again the Thomas-Fermi approximation, as in Sect. 3.1. For this case (*quadratic-minus-quartic* potential) we approximate $\rho_b = \rho_{\text{TF}}$ using (24).

- If the Thomas-Fermi expression cannot be reconstructed analytically, we assume the general formula:

$$\rho_b = \rho_0^{\text{fit}} - \frac{1}{2} \left(a_x^{\text{fit}} x^2 + a_y^{\text{fit}} y^2 + a_4^{\text{fit}} (x^2 + y^2)^2 \right), \tag{33}$$

and use a least-square procedure to find coefficients $C = \left\{ \rho_0^{\text{fit}}, a_x^{\text{fit}}, a_y^{\text{fit}}, a_4^{\text{fit}} \right\}$ minimizing the functional

$$J(C) = \frac{1}{2} ||\rho(x, y) - \rho_b(C, x, y)||_{Th_{\text{trunc}}}^2. \tag{34}$$

Note that the minimization is performed on the domain Th_{trunc} representing the condensate after removing the regions corresponding to vortex domains (where the density has a sharp decrease to zero). A minimization on the entire domain Th resulted in a poor approximation of the Thomas-Fermi density, when tested against analytical forms. Several algorithms defining Th_{trunc} were tested and finally the most reliable and fast algorithm is based on removing individual vortex domains which are disks of radius $0.1d_{\text{min}}$, where d_{min} is the minimum distance between vortex centers. The minimization of the functional (34) is easily computed using the interface of FreeFem++ with the `Ipopt` library based on the interior point minimisation method (Wächter and Biegler, 2006).

The obtained background density ρ_b is plotted in Fig. 1(b). It is now possible to isolate vortex density and fit it with a Gaussian, following

$$\rho_b - \rho = \rho_b(1 - \rho_v) = A \exp \left[-\frac{1}{2} \frac{(x - x_0)^2}{r_x{}^2} - \frac{1}{2} \frac{(y - y_0)^2}{r_y{}^2} \right]. \tag{35}$$

The amplitude A is used to define the vortex contrast (as in experiments, Coddington et al. (2004)), since $A/\rho_b(x_0, y_0)$ is the ratio between the "missing" column density at vortex center and the corresponding TF value (see also Danaila (2005)). Only vortices with a contrast greater than 0.3 are considered to compute core radii $r_v = (r_x + r_y)/2$. Note that we used for the vortex fit (35) an anisotropic Gaussian. In practice $r_x \approx r_y$. As the vortex center position has been already computed, the fit is simplified by setting $x_0 = x_v, y_0 = y_v$. When fitting the vortex center position during tests, we obtained the expected behaviour: $x_0 \approx x_v$ and $y_0 \approx y_v$. The fit following Eq. (35) is computed on individual vortex domains which are disks of radius $0.45d_{\text{min}}$ centered on vortex center, as shown in Fig. 1(c). The `Ipopt` library is again used for the least-square fit, which ensures fast and reliable results.

The program also computes the characteristics of the vortex lattice (vortex arrangement and vortex lattice parameter). A mesh is built using vortex points as vertices, as shown in Fig. 1(d), on the top of the original density plot. This representation is useful to better distinguish the arrangement of vortices in different types of vortex lattices (circular, triangular, squared) than commonly used plots of contours of density. For the case of dense Abrikosov lattices (as shown in Sect. 6.1), we select vortices with 6 neighbours and compute the vortex lattice parameter as the average distance to these 6 neighbours.

4 Vortex identification in 3D

Vortex identification in 3D combines the localization algorithm described in Sect. 3.1 with the graph-based method suggested by Liu et al. (2020). The wave function is represented as a P1 finite element function on mesh elements which are now tetrahedrons. Vortex lines are described by a series of points identified as the intersection of vortex lines with faces of a mesh element. The algorithm is illustrated in Fig. 2 using numerical data corresponding to a superfluid helium field with vortex rings (Kobayashi et al., 2021). A low-density isosurface of the initial data is presented in Fig. 2(a). An important difference when studying vortex lines instead of points is the presence of reconnections (vortex line intersection) and closed loops.

The algorithm used for the localization of vortices in 2D is directly adapted to 3D by applying Eqs. (25), (29) and (31) on the faces of each tetrahedron. The result is the detection of faces crossed by vortex lines. The position of the zero density point inside those faces is computed and stored. Tetrahedrons crossed by vortex lines are plotted in Fig. 2(b). Following the method suggested by Liu et al. (2020), a graph is then built by linking the points belonging to a same mesh element and using the fact that a given point is inside two different tetrahedrons. The connectivity c of the graph allows an easy separation of vortex lines through the following algorithm:

1. Choose a point of connectivity $c = 1$ and follow the graph while keeping an ordered list of visited points. Stop when reaching a point where $c = 1$ (end of a line) or $c > 2$ (reconnection point). Remove all visited points from the graph except if the end point is a reconnection point; in this case remove the link to the penultimate point and actualize connectivity. Repeat this step until there are no points with $c = 1$ left. The ordered lists of points represent vortex lines.

2. The only remaining lines are now loops. Start by removing the ones that reconnect through the same process as before, excepting that we start and stop at points where $c > 2$.

3. Only points with $c = 2$ remain. Starting from one such point and stopping when reaching the same point ends the identification of a vortex loop. All lines have been extracted from the graph when all points have been extracted.

As the points are constrained to be on faces of tetrahedrons, the vortex line is not smooth, as illustrated in Fig. 2(c). Two possibilities exist to smooth the vortex line. For a precise study of vortex lines, it is necessary to locate more accurately its points through the interpolation of the wave function and a local minimization of the density to identify its zero. This local minimization can be done using a Newton method (Villois et al., 2016) or by combining a pseudo-vorticity method and a gradient descent minimization (Liu et al., 2020). The drawback of these methods is that they are time-consuming and directly related to the spectral representation of the solution. We privileged a second possibility to smooth the vortex line, using a simple five-point moving average along the line. This methodology proved to be very fast and sufficiently precise to identify 3D vortex lines. Since this is a low-cost methodology, the results thus obtained could be eventually used as initial (guess) data in more complex procedures, as those suggested by Villois et al. (2016); Liu et al. (2020), to identify vortex lines with spectral accuracy.

For a line l, with points l_i, $1 \leq i \leq n$, we compute new points m_i, as follows:

$$m_i = \frac{0.8}{5} \left(l_{i-2} + l_{i-1} + l_i + l_{i+1} + l_{i+2} \right) + 0.2 l_i. \tag{36}$$

The ends of the line are kept fixed and a three point window is used for the second and penultimate points. The process is repeated, with the number of iterations being an adjustable parameter. In practice, 10 iterations are usually enough. To obtain a smoother result, it is possible to use a linear interpolation on line segments to increase the number of points before applying the moving average. Figure 2(c) shows a comparison between the initial line and the smoothed result for one of the vortex lines (located at the bottom left of the initial solution). The y axis of the figure is enlarged for the clarity of the visualisation. All lines identified in the original field are plotted in Fig. 2(d).

An optional quantity that can be computed is the curvature γ of the vortex line. At a given point m_i, we estimate $\gamma(m_i) = \alpha / \Delta s$, where Δs is the arc-length variation between points m_{i-1} and m_{i+1} and α is the angle between line segments linking m_{i-1} to m_i and m_i to m_{i+1}.
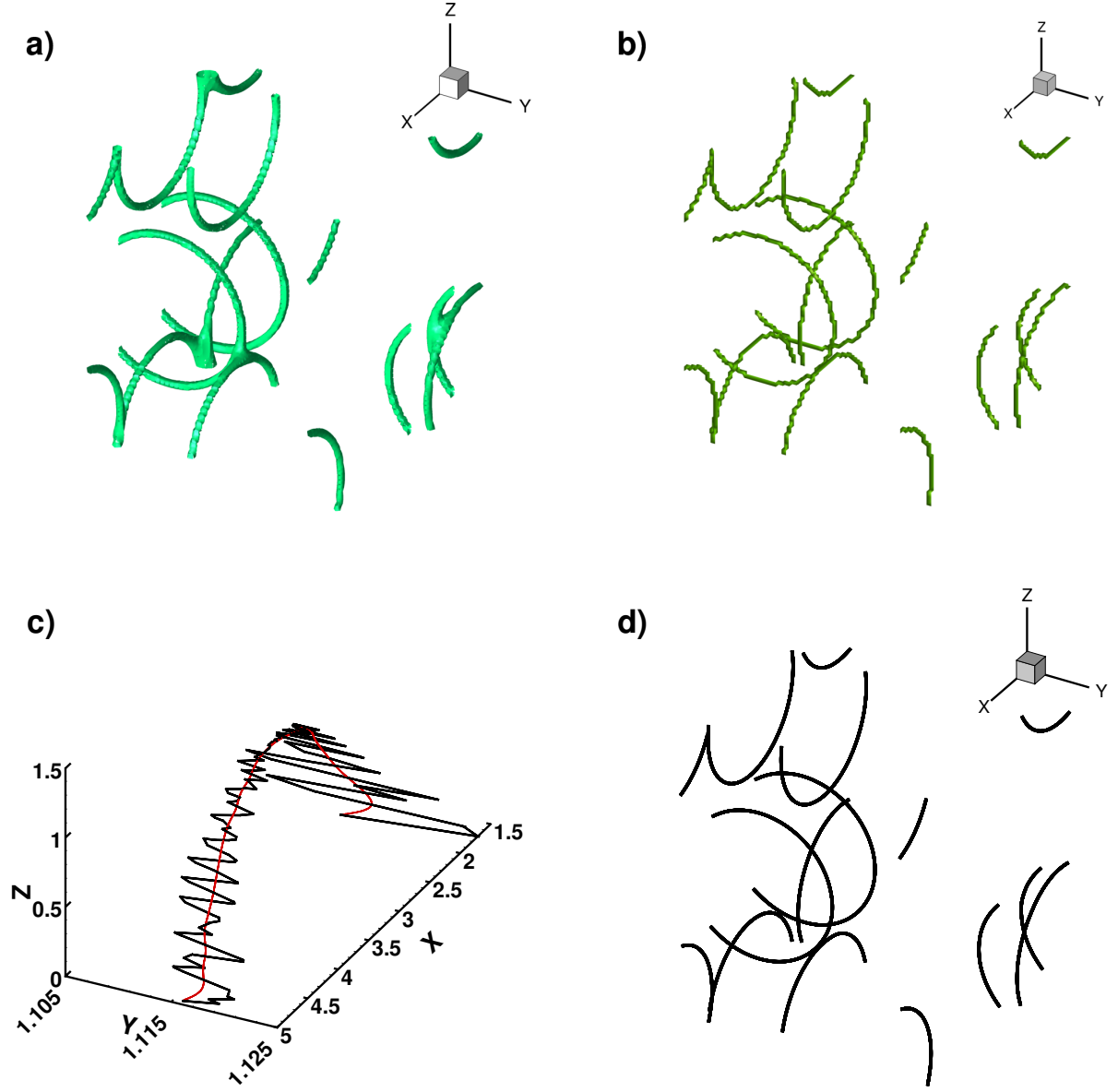
Figure 2: Illustration of the vortex identification algorithm in 3D. Test case with vortex rings in superfluid helium (Kobayashi et al., 2021) simulated inside a cubic domain (spectral method). a) Isosurface of low density ρ, b) tetrahedrons crossed by vortex lines, c) example of extracted vortex line (black) and smoothed vortex line (red), d) final vortex lines reconstructed in the full domain.

5 Vortex identification in experimental images

The toolbox distributed with this paper can be also used to analyse experimental images of BEC with vortices. Such images are obtained by switching-off the magnetic trap and imaging the absorption of a resonant laser beam along the vertical axis. The contrast in experimental pictures thus represents the atomic density integrated along the z-axis and vortices are visible as very dark regular spots (see Fig. 3). Since the information concerning the phase of the wave function is not available, the algorithms presented in Sect. 3 have to be adapted to analyse experimental images.

We illustrate the adapted algorithm by extracting vortex characteristics from an experimental image presented by Coddington et al. (2004). It corresponds to a slowly rotating atomic BEC in an harmonic trap. The original image, shown in Fig. 3(a), is transformed in the pgm format and then loaded in FreeFem++ as a rectangular array of grey values between 0 (dark) and 1 (white). A triangular finite-element mesh is then constructed from the rectangular grid and the values of grey (representing the density contrast) are represented as a P1 (piece-wise linear) function on this mesh.

The first step of the algorithm consists in separating vortex regions from the rest of the domain. We define a new variable $\rho_{iso} = 1\!\!\!1_{\rho > c_{iso}} - 1\!\!\!1_{\rho \leq c_{iso}}$, where $1\!\!\!1$ is the characteristic (indicator) function and c_{iso} is an adjustable parameter. If it is necessary to remove noisy artefacts on the border of the condensate, ρ_{iso} can be smoothed using algorithms inspired from image processing techniques. We implemented a smoothing algorithm that balances between the accuracy of the fit of the contour and the preservation of the length of the contour, but any other image smoothing algorithm can be tested. Figure 3(b) shows that ρ_{iso} separates the mesh into three subdomains that can be identified as the condensate, the vortices domain (holes) and the exterior of the condensate. If some vortices are not correctly separated from the background contrast, the value c_{iso} can be adjusted by trials and the above algorithm repeated.

For each vortex region, the vortex center (x_v, y_v) is computed as the minimum of the density in the region. Following this point, all the information needed to apply the algorithms presented in Sect. 3.2 are available.

- A mesh Th_{trunc} is created by removing circles of radius $0.1 d_{min}$ around each vortex point, where d_{min} is the minimum distance between vortices.

- The background density ρ_b is then estimated following Eqs. (33) and (34). This density is plotted in Fig. 3(c).

- The vortex density ρ_v, presented in Fig. 3(d), is fitted with the Gaussian (35) to estimate vortex radius r_v. To obtain a better estimation of the position of the vortex center, x_0 and y_0 are also parameters of the fit (which was not the case for the fit of 2D numerical data). The mesh of identified vortices is plotted in Fig. 3(e).

6 Examples and benchmarks provided with the toolbox

Several other examples and benchmarks used to validate our numerical codes are provided with the toolbox. To reduce the size of the files distributed with the 3D code, solutions have been truncated to remove tetrahedrons where $\rho > \rho_{threshold}$, with $\rho_{threshold}$ being adapted to each simulation to ensure that a high density area is still surrounding each vortex. Table 1 presents the computational (CPU) time and the number of mesh elements for the different test cases. The first test cases in each category correspond to examples used to explain algorithms in Sects. 3, 4 and 5.

	Test case	Figure	CPU time (s)	Number of triangles
2D data	Quartic - quadratic potential	1	2.5 (14.7)	32258
	Dense Abrikosov lattice	4	22.6 (79.6)	130050
3D data	Vortex rings	2	0.29	16574 (4368)
	Bose-Einstein condensate	5	1.4	267963 (84562)
	Vortex knot	6	0.5	95004 (3479)
	k^{-3} Kelvin wave	7	1.08	273733 (912)
	k^{-6} Kelvin wave	7	1.09	273673 (834)
	128^3 quantum turbulence	8	3.9	101189 (37756)
	256^3 quantum turbulence	8	774.7	99488250 (345541)
experimental images	Small Abrikosov lattice	3	13.82	91574
	Dense Abrikosov lattice	9	27.66	114720

Table 1: Computational (CPU) time and number of mesh elements for different test cases. For 2D data, the indicated time corresponds to the computation of ρ_b using numerical parameters. The computational time when using a fit to estimate ρ_b is indicated in parenthesis. For 3D cases, the number of tetrahedrons crossed by a vortex line is indicated in parenthesis.
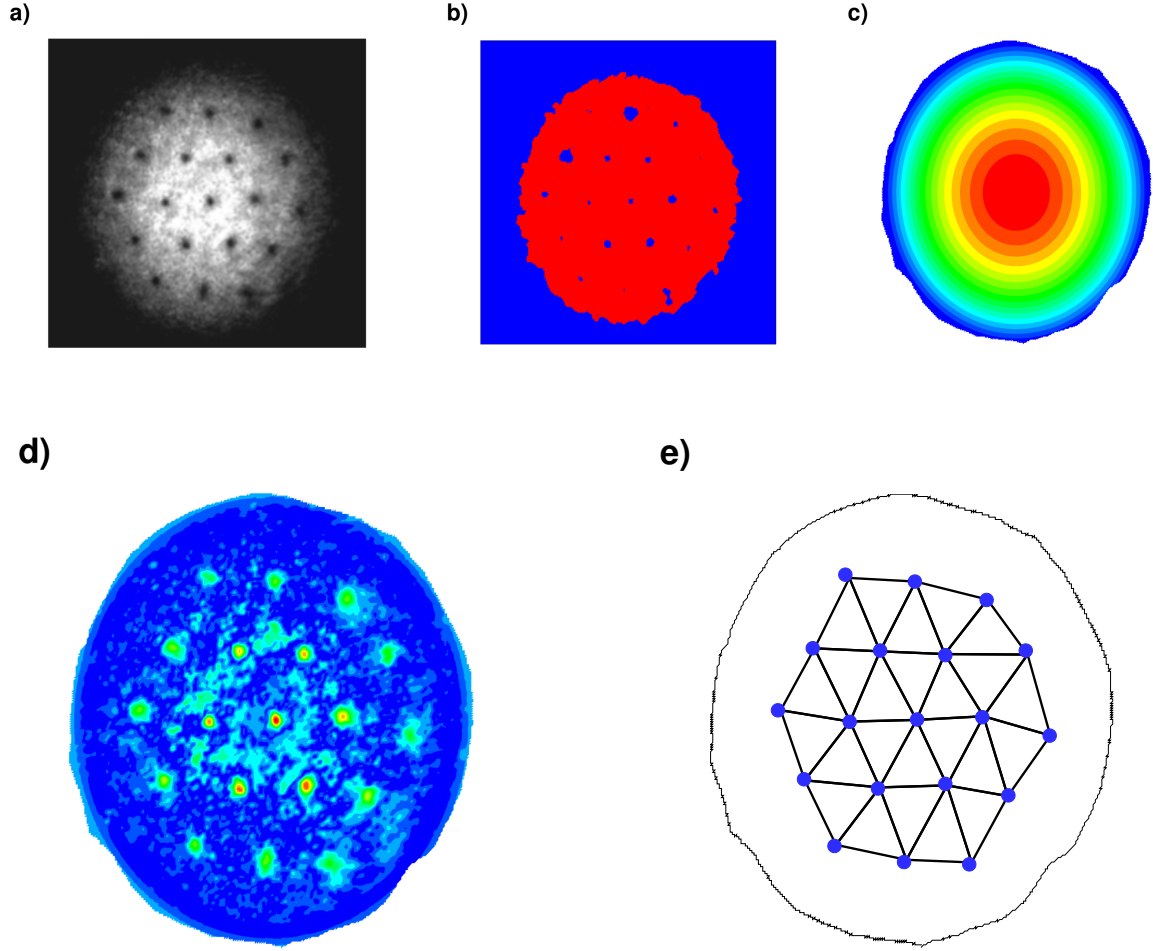
Figure 3: Illustration of vortex identification steps in experimental images. Case of a vortex lattice in a slowly rotating BEC (Coddington et al., 2004). a) Experimental image of atomic density ρ, b) region separated using a density contour ρ_{iso}, c) fitted background (Thomas-Fermi) density ρ_{TF}, d) vortex density ρ_v, e) mesh of identified vortices (blue points) and condensate border.

6.1 Dense 2D Abrikosov vortex lattice

This case corresponds to a 2D Abrikosov lattice in a fast rotating BEC. The wave function field was obtained using a spectral solver (Parnaudeau et al., 2015) for the scaled Gross-Pitaevskii equation (18) with parameters: $C_g = 1000$, $C_\Omega = 0.95$, $a_x = a_y = 1$. The background density ρ_b is obtained by computing the Thomas-Fermi density directly from simulation parameters. The initial data is shown in Fig. 4(a) and the identified vortices are plotted in Fig. 4(b). Characteristics of the Abrikosov lattice are plotted in next panels: the vortex radius r_v as a function of the distance to the condensate center r/R_{TF} in Fig. 4(c) and the inter-vortex distance s_v in Fig. 4(d).
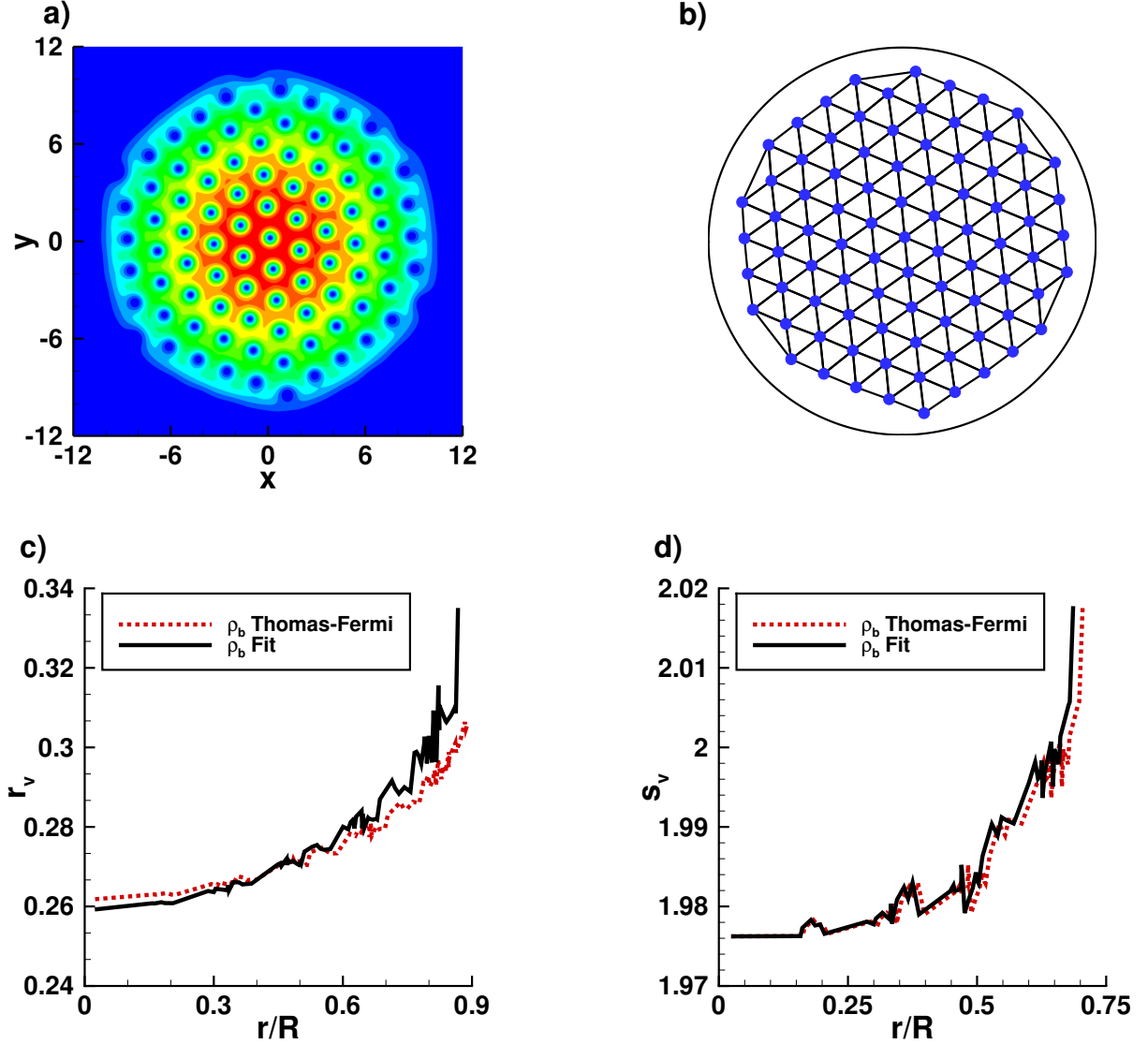

Figure 4: *Illustration of the vortex identification in 2D for a dense Abrikosov lattice in fast rotating BEC. Wave function field obtained with a spectral GP code (Parnaudeau et al., 2015). a) Initial density ρ, b) mesh of identified vortices, c) vortex radius r_v as a function of the normalized distance from the vortex to the center of the condensate r/R_{TF}, d) inter-vortex distance s_v.*

6.2 Vortices in 3D Bose-Einstein condensates

To test the toolbox on an unstructured mesh, we simulated a 3D BEC with the GP finite-element toolbox distributed by Vergez et al. (2016). A cut of the initial data with 12 vortices is shown in Fig. 5(a). The extracted lines are plotted in Fig. 5(b).
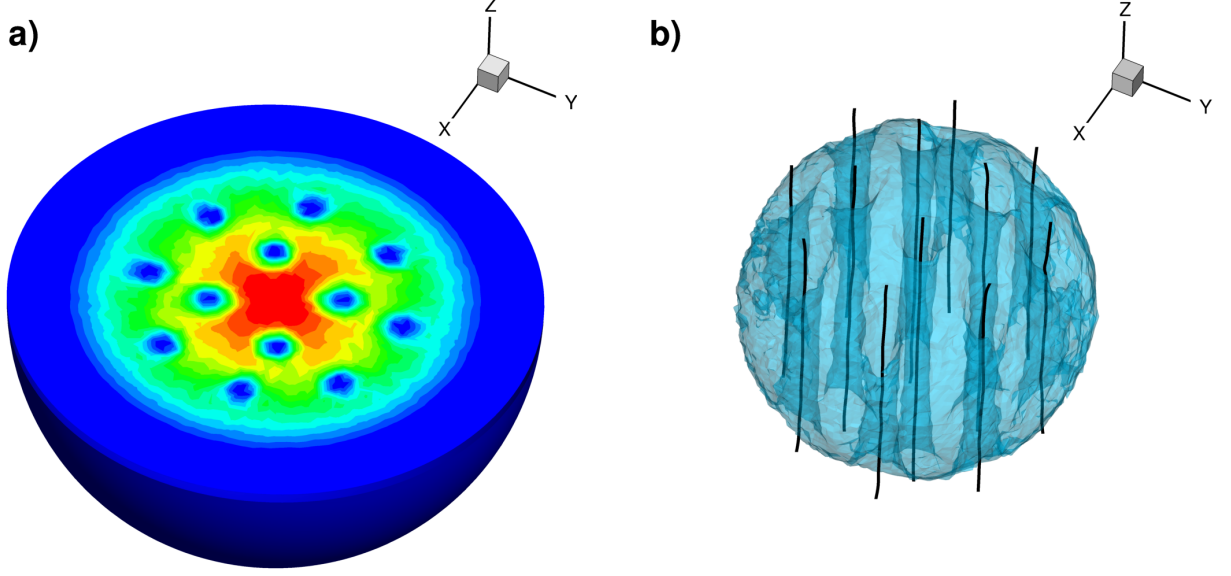

Figure 5: Illustration of the vortex line identification in 3D for a rotating BEC. Data obtained on an unstructured mesh using a finite-element solver (Vergez et al., 2016). a) Cut of the initial density ρ, b) extracted vortex lines.

6.3 3D vortex knot

For a more quantitative assessment of the results provided by the toolbox, we tested the programs for a case with known analytical description of the vortex line. This is the vortex knot benchmark presented by Villois et al. (2016). The vortex line is defined as $s(\sigma) = (x_p^{-1}(s_x(\sigma)), y_p^{-1}(s_y(\sigma)), z_p^{-1}(s_z(\sigma)))$ with:

$$s_x(\sigma) = [R_0 + R_1 \cos(q\sigma)] \cos(p\sigma), \tag{37}$$

$$s_y(\sigma) = [R_0 + R_1 \cos(q\sigma)] \sin(p\sigma), \tag{38}$$

$$s_z(\sigma) = R_1 \sin(q\sigma). \tag{39}$$

Functions x_p, y_p and z_p are introduced during the computation of the wave function to ensure periodic boundary conditions:

$$x_p = -\frac{L}{\pi} \cos\left(\frac{\pi}{L}x\right), \tag{40}$$

$$y_p = -\frac{L}{\pi} \cos\left(\frac{\pi}{L}y\right), \tag{41}$$

$$z_p = -\frac{L}{2\pi} \cos\left(\frac{2\pi}{L}z\right). \tag{42}$$

A comparison between the vortex line described analytically by Eq. (37) and the result obtained with the toolbox is shown in Fig. 6(a). The wave function was computed with a 128^3 grid, in a cubic domain of side 2π. The knot parameters are: $R_0 = 8\xi$, $r_1 = \xi/2$, $p = 2$ and $q = 5$ with $\xi = 0.5\Delta x$. The curvature γ of the vortex line was computed as an additional test and compared in Fig. 6(b) to analytical values. A fairly good approximation is observed. To reach the precision obtained by Villois et al. (2016), a higher resolution and a better smoothing method are needed (with the drawback of considerably increasing the computational time). The script used to create the initial wave function for this case is included in the toolbox.
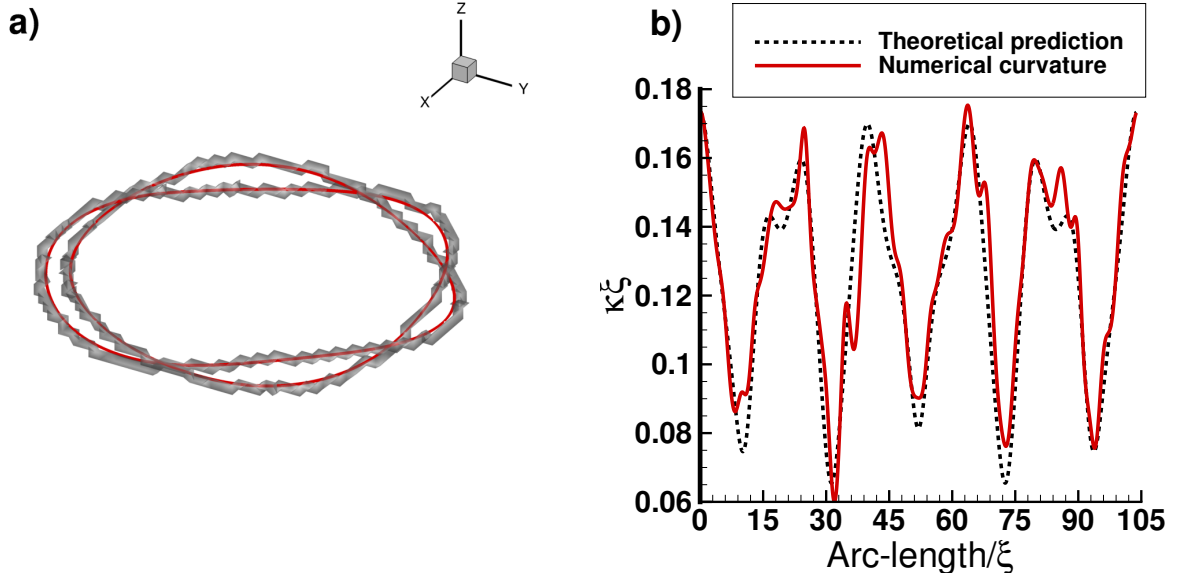
Figure 6: Illustration of the vortex line identification in 3D for a vortex knot. Test case suggested by Villois et al. (2016). a) Vortex line (in red) inside the mesh elements identified of circulation $\kappa = 1$, b) comparison of the vortex line curvature with analytical values.

6.4 Kelvin waves

Another validation against analytical forms of vortex lines considered Kelvin waves with a k^{-3} and k^{-6} energy spectrum. Oscillations of the vortex line were generated following the method suggested by Proment et al. (2013). A Padé approximation of the vortex profile (Berloff, 2004) was used in each xy plane. The vortex center in each plane was placed along a line $(X(z), Y(z), z)$ defined by:

$$X(z) + iY(z) = \mathcal{F}^{-1}\left(\sqrt{n_{th}(k)}e^{i\phi(k)}\right), \tag{43}$$

where $n_{th}(k)$ is the imposed spectrum (here k^{-3} or k^{-6}) and $\phi(k)$ is randomly distributed in $[0, 2\pi[$. $\mathcal{F}^{-1}$ denotes the inverse Fourier transform.

To compute the spectrum, we start by remeshing the line on the interval $[0, L_z]$ where L_z is the simulation box size. The line is expressed as $L(z) = (l_x(z), l_y(z), z)$ and the spectrum is:

$$n_{kw} = |\hat{R}(k)|^2, \tag{44}$$

where $r(z) = l_x(z) + il_y(z)$ and $\hat{R}$ is the Fourier transform of r. Figure 7 shows that the numerical spectrum is close to the imposed power law $n_{th}(k)$ in both cases. The script used to create the initial wave function for this case is included in the toolbox.
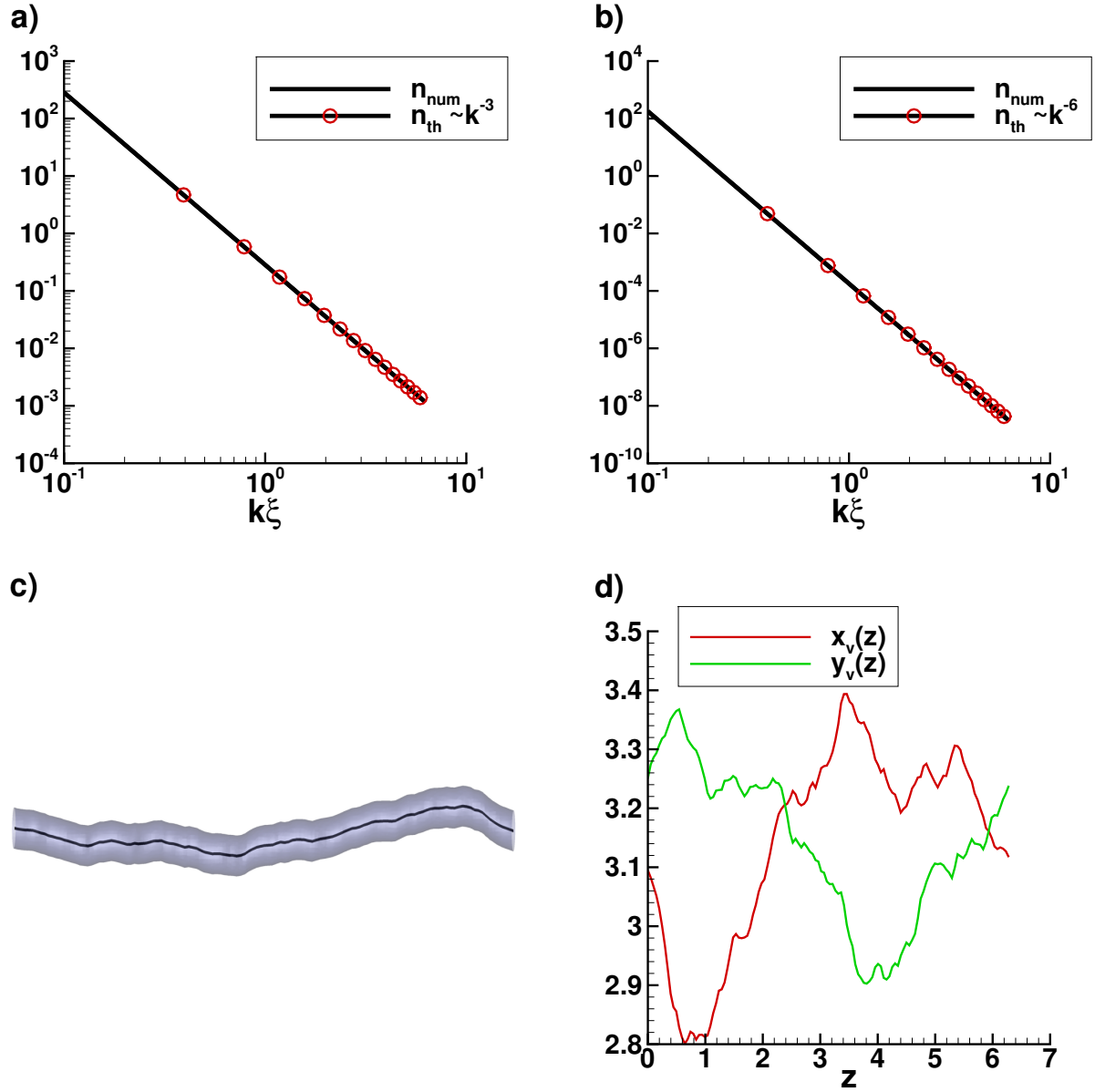
Figure 7: Kelvin wave spectrum for a vortex line scaling as k^{-3} (a) and k^{-6} (b). Test case suggested by Proment et al. (2013). Numerical spectrum in black and theoretical spectrum in red. (c) Vortex line and density iso-surface for the k^{-3} case, (d) vortex coordinates $x_v(z)$ and $y_v(z)$ for the same case.

6.5   3D quantum turbulence

The final test case for the 3D algorithm considered a quantum turbulence field with a large number of vortices. The wave function field was obtained from a GP spectral simulation with an initial condition containing Taylor-Green vortices (Kobayashi et al., 2021). Results are presented in Fig. 8: 155 vortex lines were identified from a 128^3-grid simulation and 640 lines were found in a 256^3-grid simulation. The smaller vortex loops have a radius of the order of the mesh resolution. Vortex identification is fast, as shown by computational times presented in Tab. 1, despite the large number of tetrahedrons involved in the process. Villois et al. (2016) reported, for a similar test case, a CPU time of 6 hours on a 64 core cluster using MPI. As our method is local, higher resolutions can be treated by separating the simulation in small subdomains that can be analysed separately. The vortex lines can then be linked together before the smoothing.
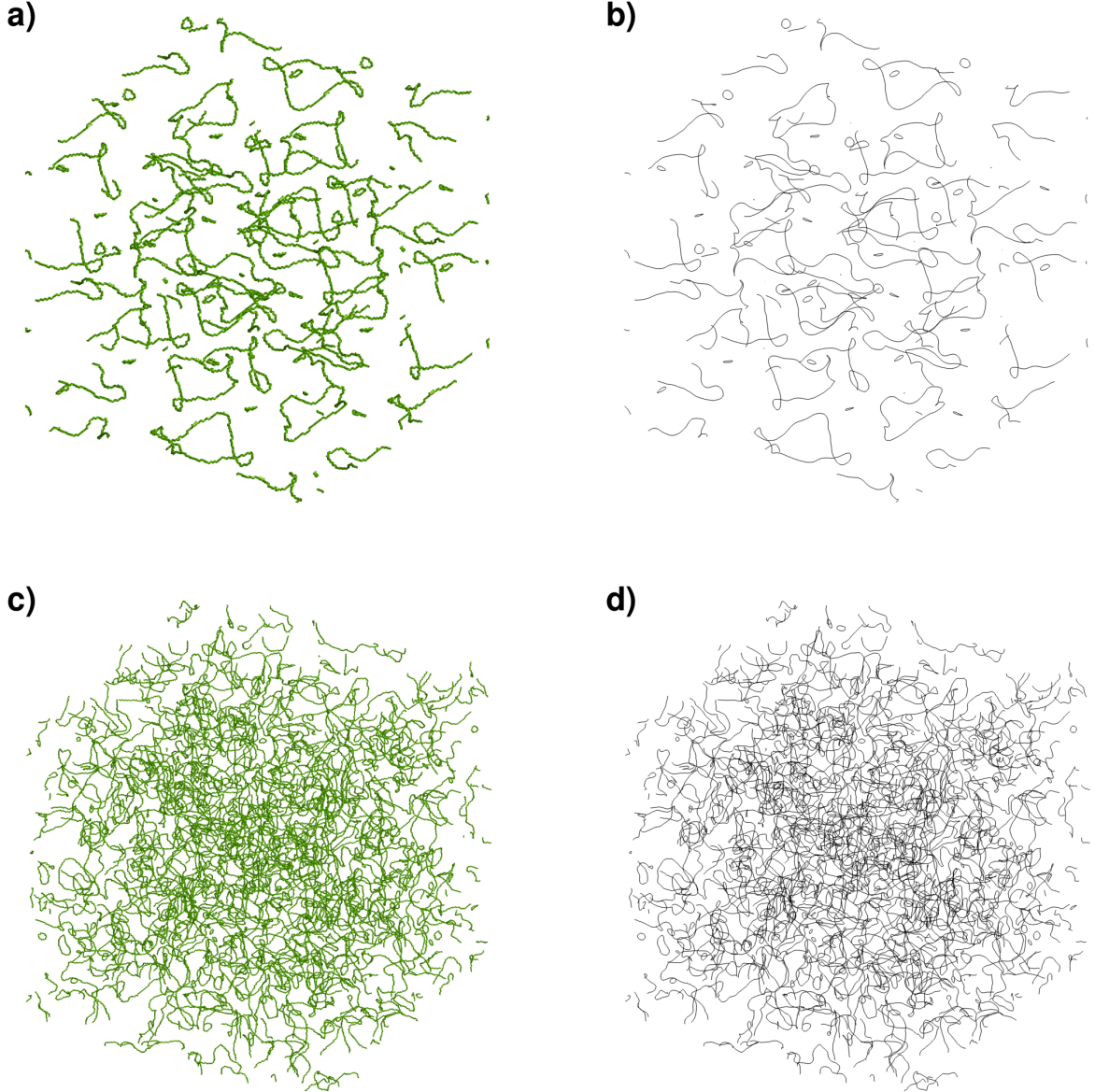

Figure 8: Illustration of the vortex identification in 3D for quantum turbulence in superfluid helium. Spectral numerical simulation of the GP equation (Kobayashi et al., 2021). Tetrahedrons with non-zero circulation on the left and identified vortex lines on the right for two resolutions: 128^3 (a,b) and 256^3 (c,d).

17

6.6 Experimental dense Abrikosov lattice image

To complete the test presented in Sect. 5, we considered the more challenging test of an experimental image with a dense Abrikosov vortex lattice reported by Coddington et al. (2004). The result is shown in Fig. 9. Vortices on the side of the condensate are not identified as they are in a very low contrast regions. The vortex lattice properties were also computed. We notice in Fig. 9(d) that the vortex radius r_v increases for vortices away from the BEC center, which is the expected behaviour because of the low local atomic density. As the length scale information is not available with images, all quantities are expressed in pixel size.
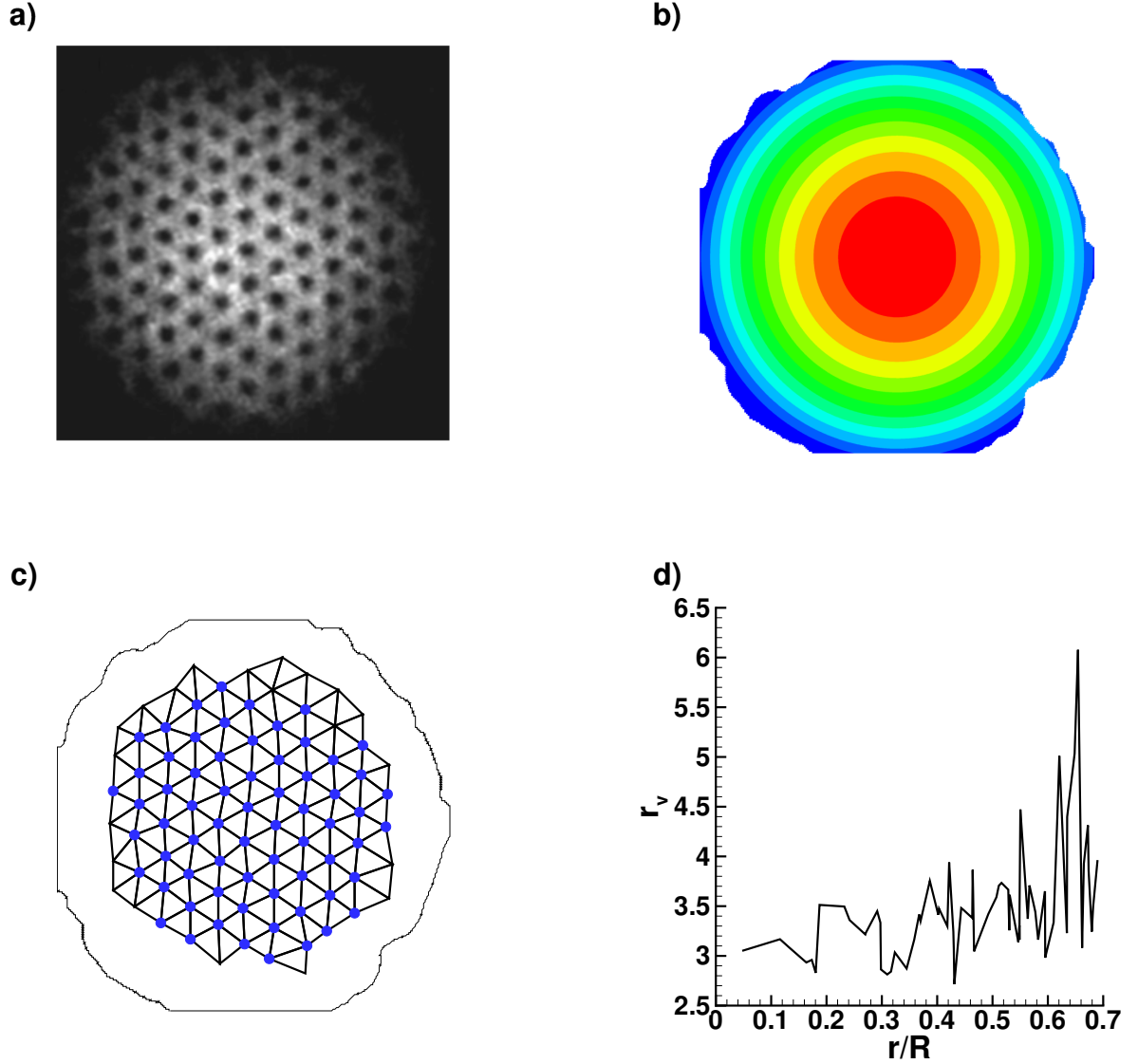
a)

b)

c)

d)

Figure 9: Illustration of the vortex identification in an experimental image with dense Abrikosov vortex lattice (Coddington et al., 2004). a) Initial image, b) background density ρ_b, c) mesh of identified vortex points and Thomas-Fermi border, d) vortex radius r_v as a function of the distance to the center of the BEC. In panel (c), only vortices with high enough density contrast (and thus fitted with a Gaussian) are represented by blue circles.

7 Description of the programs

In this section, we first describe the architecture of the programs and the organisation of the provided files. Then we describe the input parameters and the structure of the output files for the different codes.

7.1 Program architecture

The `Postproc_toolbox` directory is organized around three main subdirectories, `postproc_data_2D`, `postproc_data_3D` and `postproc_image_2D`, corresponding to the three algorithms described above: the case of 2D simulation data, that of 3D simulation data and the case of experimental images. Each of these subdirectory contains a file: *FFEM_postproc_$case.edp* which is the main FreeFem++ script file and a folder `A_macro` containing macros used in main code. For example, to run the vortex identification in 3D simulation data, the user can use the command `FreeFem++ FFEM_postproc_data_3D.edp`. Initial data fields or images for the examples presented in this paper are stored in the `INIT` folder.

The obtained solutions are saved in the folder `dircase`. Depending on the output format selected by the user, data files are generated in specific folders for visualization with Tecplot, Paraview or Gnuplot. We also provide in the folder `Figures` ready-made layouts for Tecplot. 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.

Before running the main scripts, it could be useful for the reader to get familiar with the initial data fields. This will help with the use of the toolbox in different programming frameworks. For this purpose, we included some auxiliary (optional) scripts that could be used to generate again data files for the examples presented in this paper (folder `extra_codes_for_examples`) or to have an example of how to link an external solver (GP or similar) to the toolbox (folder `data_example`). More in detail:

1. The directory `extra_codes_for_examples` contains codes generating the wave function and computing vortex line properties for the examples described in Sects. 6.2 to 6.4:
   - *generate_vortex_knot.edp*: generates the vortex knot wave function (Sect. 6.3).
   - *compute_vortex_knot_curvature.edp*: computes numerical and theoretical curvatures for the vortex knot and compares them.
   - *generate_sphere_mesh.edp*: generates a spherical mesh for the BEC example with 12 vortices (Sect. 6.2).
   - *generate_Kelvin_wave.edp*: generates the wave function of a vortex line deformed by a Kelvin wave (Sect. 6.4).
   - *compute_Kelvin_spectrum.edp*: computes the Kelvin wave spectrum of the vortex line identified in the field file generated by the previous script.

2. The directory `data_example` contains example codes showing how to save numerical data and parameters in the format used in the toolbox:
   - *data_example_2D.f90*: Fortran code computing the wave function of a 2D BEC with three vortices.
   - *data_example_3D.f90*: Fortran code computing the wave function of a 3D BEC with a central vortex line.
   - *convert_data_2D.edp*: FreeFem++ script converting the data created by the 2D Fortran code to the FreeFem++ format used in the toolbox.
   - *convert_data_3D.edp*: FreeFem++ script converting the data created by the 3D Fortran code to the FreeFem++ format used in the toolbox.

Going back to main scripts, the complete architecture of the `Postproc_toolbox` main directory is the following:

1. *FFEM_postproc_generate_files.edp* is a FreeFem++ script used to automatically run preliminary or maintenance operations and generate data and mesh used in the presented examples. It is useful to run this script before testing different examples. A *clean* option was also added to help managing data files and results of compilation.

2. The directory `Common_Macros` contains one file:
   - *Macro_common.idp*: macros used to create directories and save the results,

3. The directory `postproc_data_2D` contains codes dedicated to vortex identification in 2D numerical data with the algorithms presented in Sect. 3:
   • *FFEM_postproc_data_2D.edp*: the main FreeFem++ script.
   • `INIT`: directory containing data and parameter files for the Abrikosov lattice and quartic potential examples described in Sects. 3 and 6.1.
   • `Figures`: directory containing Tecplot layouts used to replot the figures shown in this article. The main code must be run with the associated example before opening the layout to replot the figure.
   • `A_macro`: directory containing macros used for the vortex identification in 2D complex data.

4. The directory `postproc_data_3D` contains codes dedicated to vortex identification in 3D numerical data with the algorithms presented in Sect. 4:
   • *FFEM_postproc_data_3D.edp*: the main FreeFem++ script.
   • *FFEM_files_for_Kelvin_wave_vortex_knot_lay.edp*: script used to generate the files necessary to replot the figures in Sects. 6.3 and 6.4.
   • `INIT`: directory containing the data and parameter files for the examples presented in Sects. 4, 6.2 and 6.5: vortex rings in superfluid helium, Bose-Einstein condensate and quantum turbulence. The scripts in the `extra_codes_for_examples` directory must be used to obtain the initial data for the examples described in Sects. 6.3 and 6.4.
   • `Figures`: directory containing Tecplot layouts used to replot the figures shown in this article. The main code must be run with the associated example before opening the layout to replot the figure. To obtain the curvature and spectrum shown in Figs. 6 and 7, it is necessary to use the scripts in the `extra_codes_for_examples` directory before opening the layout.
   • `A_macro`: directory containing macros used for the vortex identification in 3D complex data.

5. The directory `postproc_image_2D` contains codes dedicated to vortex identification in 2D experimental images with the algorithms presented in Sect. 5:
   • *FFEM_postproc_image_2D.edp*: the main FreeFem++ script.
   • `INIT`: directory containing the data files for the vortex lattice presented in Sects. 5 and 6.6.
   • `Figures`: directory containing Tecplot layouts used to replot the figures shown in this article. The main code must be run with the associated example before opening the layout to replot the figure.
   • `A_macro`: directory containing macros used for the vortex identification in 2D experimental images.

7.2 Macros and functions

The different macros and functions used in the toolbox are:

1. In `postproc_data_2D`, the directory `A_macro` contains:
   • *M_bec_compute_TF.edp*: macro computing the Thomas-Fermi density ρ_{TF} and condensate radius R_{TF}.
   • *M_bec_fit_gaussian.edp*: macro using Ipopt to fit the vortex density ρ_v with a Gaussian function (see Eq. (35)).
   • *M_bec_fit_TF.edp*: macro using Ipopt to fit the Thomas-Fermi density (see Eq. (34)).
   • *M_bec_lattice_stats.edp*: macro computing characteristics of the vortex lattice (radius and inter-vortex spacing as a function of the distance to the center of the BEC).
   • *M_bec_read_data.edp*: macro used to read the initial data (wave function field) and parameter files.
   • *M_bec_sort_save_results.edp*: macro used to save the results.
   • *M_bec_vortex_lattice.edp*: macro building a mesh corresponding to the vortex lattice.
   • *M_bec_vortex_regions.edp*: macro computing the vortex position (Eqs. (29) and (31)) and building the mesh Th_{trunc}.

2. In `postproc_data_3D`, the directory `A_macro` contains:
   • *M_bec_build_line.edp*: macro building a vortex line following the method described in Sect. 4.
   • *M_bec_compute_curvature.edp*: macro computing the curvature of a vortex line.
   • *M_bec_compute_TF.edp*: macro computing the Thomas-Fermi density ρ_{TF} and condensate radius R_{TF}.
   • *M_bec_read_data.edp*: macro used to read the initial data (wave function field) and parameter files.
   • *M_bec_smooth_line.edp*: macro smoothing a vortex line using a 5-point moving average (see Eq. (36)).

3. In `postproc_image_2D`, the directory `A_macro` contains:
   - *M_bec_fit_gaussian.edp*: macro using Ipopt to fit the vortex density ρ_v with a Gaussian function (see Eq. (35)).
   - *M_bec_fit_TF.edp*: macro using Ipopt to fit the Thomas-Fermi density (see Eq. (34)).
   - *M_bec_lattice_stats.edp*: macro computing characteristics of the vortex lattice (radius and inter-vortex spacing as a function of the distance to the center of the BEC).
   - *M_bec_reduceres_smooth.edp*: macro used to smooth and lower the resolution of the initial image.
   - *M_bec_regularize_isoline.edp*: macro used to smooth the region borders in ρ_{iso}.
   - *M_bec_sort_save_results.edp*: macro used to save the results.
   - *M_bec_vortex_lattice.edp*: macro building a mesh corresponding to the vortex lattice.
   - *M_bec_vortex_regions.edp*: macro computing the vortex position and building the mesh Th_{trunc}.

4. The *vortextools.cpp* file is included in the `FreeFem-sources/plugin/sec` directory for FreeFem++ versions above 4.12 and contains functions used by the previously described scripts and macros:
   - *smoothCurve*: function smoothing a vortex line using a 5-point moving average (see Eq. (36)).
   - *uZero*: function computing vortices in a 3D complex field (see Eqs. (29) and (31)); it returns a P0 function of value 1 if the tetrahedron is crossed by a vortex line and 0 otherwise.
   - *uZero2D*: function computing vortices in a 2D complex field (see Eqs. (29) and (31)), returns a P0 function of value 1 if the tetrahedron is crossed by a vortex line and 0 otherwise, an array containing vortex positions and $d_{\min}$.
   - *ZeroLines*: function used to identify the vortex lines through the graph described in Sect. 4.
   - *curvatureL*: function computing the curvature of a curve described by a line 1D-mesh.
   - *abscisses*: function computing the arc-length along a curve.
   - *interpol*: function using linear interpolation to transform data from an irregular discretization to a uniform one.

7.3 Input parameters

Parameters for the identification process must be adapted in the main code file. They differ depending on the type of data and the space dimension (2D/3D).
(1) In the file `FFEM_postproc_data_2D.edp`, the parameters are:

- **paramread**: defines the method to compute the background density: computes ρ_{TF} from the Gross-Pitaevskii parameters (`true`) or uses the fit presented in Eq. (34) (`false`). If `true`, a parameter file must be provided.

- **displayplot**: controls the output information to plot. Possible values from 0 (no plots), to 2 (plots data for all vortices during the fit).

- **save**: if `true`, data used in the Tecplot layouts will be saved.

- **iwait**: a Boolean indicating if the code must wait for user input when a plot is shown (`true`) or if it can continue (`false`).

- **hole**: a Boolean indicating whether the condensate has a central hole as in the quartic-quadratic potential test case.

- **fcase**: the name of the input file.

- **dirInput**: the name of the directory where the input file is stored.

- **dircase**: the name of the directory where the results will be stored.

(2) In the file `FFEM_postproc_data_3D.edp`, the parameters are:

- **BEC**: if `true`, the tetrahedrons outside the Thomas-Fermi zone will be removed. In this case, a parameter file must be given.

- **displayplot**: controls the output information to plot. Possible values from 0 (no plots), to 2 (plots data for all vortices during the fit).

- **save**: if `true`, data will be saved in the `Tecplot` folder.

- **iwait**: a Boolean indicating if the code must wait for user input when a plot is shown (`true`) or if it can continue (`false`).

- **curv**: the vortex line curvature is computed if `true`.

- **npadd**: the number of points to add on each segment of the line before smoothing. The minimum is 1 (the extremity of the segment).

- **smooth**: a Boolean indicating whether to smooth the line or not.

- **nsmooth**: the number of smoothing iterations.

- **fcase**: the name of the input file.

- **dirInput**: the name of the directory where the input file is stored.

- **dircase**: the name of the directory where the results will be stored.

(3) In the file `FFEM_postproc_image.edp`, the parameters are:

- **displayplot**: controls the output information to plot. Possible values from 0 (no plots), to 2 (plots data for all vortices during the fit).

- **save**: if `true`, data will be saved in the `Tecplot` folder.

- **iwait**: a Boolean indicating if the code must wait for user input when a plot is shown (`true`) or if it can continue (`false`).

- **reduceres**: a Boolean, the image resolution is reduced when the value is `true`.

- **fcase**: the name of the input file.

- **dirInput**: the name of the directory where the input file is stored.

- **dircase**: the name of the directory where the results will be stored.

- **cutisoval**: value c_{iso} used when separating the regions.

- **cutisoval2**: a second value $c_{iso,2}$ used when separating the regions.

7.4 Outputs

When a computation starts, the `OUTPUT_$case` directory is created. It contains five folders. The `RUNPARAM` directory contains a copy of the code and data files, allowing an easy identification of each case and preparing an eventual rerun of the same case. The other folders contains different output format files of the computed solution, to be visualised with Tecplot, Paraview, Gnuplot or FreeFem++ . The content of those subfolders depends on the case and on the computation parameters:

- For 2D data, the `Gnuplot` folder contains multiple files. Computed quantities are stored in the `vortex_coord_$case.dat` file. The column are in order: the vortex number, the vortex coordinates x_v and y_v, the radius, the distance from the center of the BEC and a boolean indicating if Ipopt has converged (value 0) or not (value 1) when fitting the vortex. The `sorted_radius.dat` and `stats_lattice.dat` contain respectively the core radius and the inter-vortex distance (second column) sorted by increasing distance to the center of the condensate (first column). The `Tecplot` folder contains files for the initial density (`rhoinit.dat`), the background density (`rhoTF.dat`), the vortex wave function on the whole domain (`rhov.dat`) and only on vortex regions (`rhovVortex.dat`), as well as meshes for the Thomas-Fermi border of the condensate (`TF_meshL.dat`) and the estimated vortex points (`vortex_mesh.dat`).

- For 3D data: The smoothed vortex lines are stored as *.vtk* files in the `Paraview` folder, as FreeFem++ *.meshb* files in the `meshL` folder and as *.dat* files in the `Gnuplot folder` (as a succession of point coordinates x, y, z). The file names are `line_` followed by **fcase** and the line number. Files starting by `zeroline` in the `Gnuplot` and `Paraview` folders are the vortex lines before smoothing. The `Tecplot` folder contains the initial density (`rho_init.dat`), a mesh of the tetrahedrons crossed by vortex lines (`vortex_mesh.dat`) and if it is computed, the Thomas-Fermi density (`rho_TF.dat`). If the curvature is computed, it is saved in the Gnuplot folder as `curv_` followed by the line number.

- For experimental images, the `Gnuplot` and `Tecplot` folders are almost the same as in the 2D data case. The only difference is that there are two meshes built from vortex points: the first one uses the points estimated from the vortex regions (`vortex_mesh_estimated.dat`) and the second one uses vortex positions obtained from the fit with a Gaussian (`vortex_mesh_fitted.dat`).

8 Summary and conclusions

The main advantage of the toolbox distributed with this paper is its ability to give fast and accurate results for vortex identification in different configurations: 2D, 3D and experimental images. While more precise numerical methods for the identification of vortices could be set by combining local minimization and Fourier interpolation (Villois et al., 2016; Caliari and Zuccher, 2017), we showed that the the computation of the zeros of a P1 finite element function is a reliable method to obtain a good degree of precision while keeping the computational cost low. This approach is well suited for fields with a large number of vortices, as obtained in high resolution simulation of quantum turbulence, where local minimization would be too costly. Another advantage of the toolbox is to be adaptable to any computational framework (finite elements, spectral, finite differences) providing the wave function field. Examples how to interface the toolbox with other software are also provided.

The toolbox was created with FreeFem++ , a free and open-source software for the resolution of partial differential equations and the manipulation of finite elements. The main steps of the method are (i) the search of the zeros of the wave function (ii) the delimitation of the vortex region (in 2D) or the vortex line (in 3D) (iii) the computation of vortex characteristics. The numerical code was validated against a wide range of test cases available in the literature for both 2D and 3D configurations.

To facilitate its use, the toolbox is provided with separate folders containing all the necessary files (parameters, restart files) to run all the cases described in the paper. Ready-made scripts and layouts allow the user to generate the figures presented in this paper with newly generated data after running the programs. As a consequence,the present toolbox can be easily tested and modified for the computation of other diagnostics or the identification of vortices in other types of systems (*e. g.* vortices in superconductors described by a Ginzburg-Landau wave function field).

Acknowledgements

Bibliography

References

Abid, M., Huepe, C., Metens, S., Nore, C., Pham, C., Tuckerman, L., Brachet, M. E., 2003. Gross-Pitaevskii dynamics of Bose-Einstein condensates and superfluid turbulence. Fluid Dynamics Research 33 (5-6), 509–544.

Aftalion, A., Danaila, I., 2003. Three-dimensional vortex configurations in a rotating Bose-Einstein condensate. Physical Review A 68, 023603.

Aftalion, A., Danaila, I., 2004. Giant vortices in combined harmonic and quartic traps. Physical Review A 69, 033608(1–6).

Antoine, X., Duboscq, R., 2014. GPELab, a Matlab toolbox to solve Gross-Pitaevskii equations I: Computation of stationary solutions. Comput. Phys. Comm. 185 (11), 2969–2991.

Antoine, X., Duboscq, R., 2015. GPELab, a Matlab toolbox to solve Gross-Pitaevskii equations II: dynamics and stochastic simulations. Comput. Phys. Comm. 193, 95–117.

Barenghi, C. F., Skrbek, L., Sreenivasan, K. R., 2014. Introduction to quantum turbulence. Proceedings of the National Academy of Sciences of the United States of America 111 (SUPPL. 1), 4647–4652.

Berloff, N. G., 2004. Padé approximations of solitary wave solutions of the Gross–Pitaevskii equation. Journal of Physics A: Mathematical and General 37 (5), 1617.

Bewley, G. P., Lathrop, D. P., Sreenivasan, K. R., 2006. Visualization of quantized vortices. Nature 441, 588.

Bisset, R. N., Wang, W., Ticknor, C., Carretero-González, R., Frantzeskakis, D. J., Collins, L. A., Kevrekidis, P. G., 2015. Robust vortex lines, vortex rings, and hopfions in three-dimensional Bose-Einstein condensates. Phys. Rev. A 92, 063611.

Bretin, V., Stock, S., Seurin, Y., Dalibard, J., 2004. Fast rotation of a Bose-Einstein condensate. Phys. Rev. Lett. 92, 050403.

Caliari, M., Rainer, S., 2013. GSGPEs: A Matlab code for computing the ground state of systems of Gross-Pitaevskii equations. Comput. Phys. Comm. 184 (3), 812 – 823.

Caliari, M., Zuccher, S., 2017. Infftm: Fast evaluation of 3d Fourier series in MATLAB with an application to quantum vortex reconnections. Computer Physics Communications 213, 197–207.

Caplan, R., 2013. NLSEmagic: Nonlinear Schrödinger equation multi-dimensional Matlab-based GPU-accelerated integrators using compact high-order schemes. Comput. Phys. Comm. 184 (4), 1250–1271.

Carles, R., Danchin, R., Saut, J.-C., 2012. Madelung, Gross–Pitaevskii and Korteweg. Nonlinearity 25 (10), 2843–2873.

Coddington, I., Haljan, P. C., Engels, P., Schweikhard, V., Tung, S., Cornell, E. A., 2004. Experimental studies of equilibrium vortex properties in a Bose-condensed gas. Phys. Rev. A 70, 063607.

Crasovan, L.-C., Pérez-Garcìa, V. M., Danaila, I., Mihalache, D., Torner, L., 2004. Three–dimensional parallel vortex rings in Bose–Einstein condensates. Physical Review A 70, 033605(1–5).

Danaila, I., 2005. Three-dimensional vortex structure of a fast rotating Bose–Einstein condensate with harmonic-plus-quartic confinement. Phys. Review A 72, 013605(1–6).

Danaila, I., Hecht, F., 2010. A finite element method with mesh adaptivity for computing vortex states in fast-rotating Bose-Einstein condensates. J. Comput. Physics 229, 6946–6960.

Danaila, I., Kazemi, P., 2010. A new Sobolev gradient method for direct minimization of the Gross–Pitaevskii energy with rotation. SIAM J. Sci. Computing 32, 2447–2467.

Danaila, I., Protas, B., 2017. Computation of ground states of the Gross-Pitaevskii functional via Riemannian optimization. SIAM J. Sci. Computing 36, B1102–B1129.

Dion, C. M., Cancès, E., 2007. Ground state of the time-independent Gross-Pitaevskii equation. Comput. Phys. Comm. 177, 787–798.

Donnelly, R., 2009. The two-fluid theory and second sound in liquid helium. Physics Today 62 (10), 34.

Engels, P., Coddington, I., Haljan, P., Schweikhard, V., Cornell, E., 2003. Observation of long-lived vortex aggregates in rapidly rotating Bose-Einstein condensates. Phys. Rev. Lett. 90, 170405.

Haljan, P. C., Coddington, I., Engels, P., Cornell, E. A., 2001. Driving Bose-Einstein condensate vorticity with a rotating normal cloud. Phys. Rev. Lett. 87, 210403–210407.

Hohenester, U., 2014. OCTBEC a Matlab toolbox for optimal quantum control of Bose-Einstein condensates. Comput. Phys. Comm. 185 (1), 194–216.

Jackson, B., McCann, J. F., Adams, C. S., 1999. Vortex rings and mutual drag in trapped Bose-Einstein condensates. Phys. Rev. A 60, 4882–4885.

Kishor Kumar, R., Lončar, V., Muruganandam, P., Adhikari, S. K., Balaž, A., 2019. C and Fortran OpenMP programs for rotating Bose-Einstein condensates. Computer Physics Communications 240, 74 – 82.

Kobayashi, M., Parnaudeau, P., Luddens, F., Lothodé, C., Danaila, L., Brachet, M., Danaila, I., 2021. Quantum turbulence simulations using the Gross–Pitaevskii equation: High-performance computing and new numerical benchmarks. Computer Physics Communications 258, 107579.

Kong, L., Hong, J., Zhang, J., 2014. LOD-ms for Gross-Pitaevskii equation in Bose-Einstein condensates. Communications in Computational Physics 14 (1), 219–241.

Kwon, J. K., Moon, G., Choi, J.-Y., Seo, S. W., Shin, Y.-I., 2014. Relaxation of superfluid turbulence in highly oblate Bose-Einstein condensates. Phys. Rev. A 90, 063627.

Leanhardt, A. E., Görlitz, A., Chikkatur, A. P., Kielpinski, D., Shin, Y., Pritchard, D. E., Ketterle, W., 2002. Imprinting vortices in a Bose-Einstein condensate using topological phases. Phys. Rev. Lett. 89, 190403.

Liu, D., Xiong, C., Liu, X., 2020. Vectorizing quantum turbulence vortex-core lines for real-time visualization. IEEE Transactions on Visualization and Computer Graphics.

Madison, K. W., Chevy, F., Wohlleben, W., Dalibard, J., 2000. Vortex formation in a stirred Bose-Einstein condensate. Phys. Rev. Lett. 84, 806.

Marojević, Z., Göklü, E., Lämmerzahl, C., 2016. ATUS-PRO: A FEM-based solver for the time-dependent and stationary Gross-Pitaevskii equation. Computer Physics Communications 202, 216 – 232.

Muruganandam, P., Adhikari, S., 2009. Fortran programs for the time-dependent Gross-Pitaevskii equation in a fully anisotropic trap. Comput. Phys. Comm. 180 (10), 1888–1912.

Navon, N., Gaunt, A. L., Smith, R. P., Hadzibabic, Z., 2016. Emergence of a turbulent cascade in a quantum gas. Nature 539, 72.

Neu, J. C., 1990. Vortices in complex scalar fields. Physica D 43, 385–406.

Nore, C., Abid, M., Brachet, M. E., 1997. Decaying Kolmogorov turbulence in a model of superflow. Physics of Fluids 9 (9), 2644–2669.

Parnaudeau, P., Suzuki, A., Sac-Epée, J.-M., 2015. GPS: An efficient & spectrally accurate code for computing Gross-Pitaevskii equation. ISC-2015, Research Posters Session, July 12 – 16, 2015, Germany.

Pitaevskii, L. P., Stringari, S., 2003. Bose-Einstein condensation. Clarendon Press, Oxford.

Proment, D., Barenghi, C. F., Onorato, M., 2013. Interaction and decay of Kelvin waves in the Gross–Pitaevskii model. arXiv preprint arXiv:1308.0852.

Rosenbusch, P., Bretin, V., Dalibard, J., 2002a. Dynamics of a single vortex line in a Bose-Einstein condensate. Phys. Rev. Lett. 89, 200403.

Rosenbusch, P., Petrov, D. S., Sinha, S., Chevy, F., andY. Castin, V. B., Shlyapnikov, G., Dalibard, J., 2002b. Critical rotation of a harmonically trapped Bose gas. Phys. Rev. Lett. 88, 250403.

Skrbek, L., Sreenivasan, K. R., 2012. How similar is quantum turbulence to classical turbulence? In: Davidson, P. A., Kaneda, Y., Sreenivasan, K. R. (Eds.), Ten Chapters in Turbulence. Cambridge University Press, p. 405.

Stringari, S., 1999. Phase diagram of quantized vortices in a trapped Bose-Einstein condensed gas. Phys. Rev. Lett. 82, 4371.

Tsubota, M., Fujimoto, K., Yui, S., 2017. Numerical studies of quantum turbulence. J. of Low Temperature Physics 188, 119–189.

Vergez, G., Danaila, I., Auliac, S., Hecht, F., 2016. A finite-element toolbox for the stationary Gross–Pitaevskii equation with rotation. Comput. Phys. Comm. 209, 144–162.

Villois, A., Krstulovic, G., Proment, D., Salman, H., 2016. A vortex filament tracking method for the Gross–Pitaevskii model of a superfluid. Journal of Physics A: Mathematical and Theoretical 49 (41), 415502.

Vudragović, D., Vidanović, I., Balaz, A., Muruganandam, P., Adhikari, S. K., 2012. C programs for solving the time-dependent Gross-Pitaevskii equation in a fully anisotropic trap. Comput. Phys. Comm. 183 (9), 2021 – 2025.

Wächter, A., Biegler, L. T., 2006. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. Mathematical Programming 106, 25–57.