SIAM J. Sci. Comput.
Vol. 39, No. 6, pp. B1102–B1129

COMPUTATION OF GROUND STATES OF THE GROSS–PITAEVSKII FUNCTIONAL VIA RIEMANNIAN OPTIMIZATION*

IONUT DANAILA† AND BARTOSZ PROTAS‡

Abstract. In this paper we combine concepts from Riemannian optimization [P.-A. Absil, R. Mahony, and R. Sepulchre, *Optimization Algorithms on Matrix Manifolds,* Princeton University Press, 2008] and the theory of Sobolev gradients [J. W. Neuberger, *Sobolev Gradients and Differential Equation,* 2nd ed., Springer, 2010] to derive a new conjugate gradient method for direct minimization of the Gross–Pitaevskii energy functional with rotation. The conservation of the number of particles in the system constrains the minimizers to lie on a manifold corresponding to the unit L^2 norm. The idea developed here is to transform the original constrained optimization problem into an unconstrained problem on this (spherical) Riemannian manifold, so that fast minimization algorithms can be applied as alternatives to more standard constrained formulations. First, we obtain Sobolev gradients using an equivalent definition of an H^1 inner product which takes into account rotation. Then, the Riemannian gradient (RG) steepest descent method is derived based on projected gradients and retraction of an intermediate solution back to the constraint manifold. Finally, we use the concept of the Riemannian vector transport to propose a Riemannian conjugate gradient (RCG) method for this problem. It is derived at the continuous level based on the "optimize-then-discretize" paradigm instead of the usual "discretize-then-optimize" approach, as this ensures robustness of the method when adaptive mesh refinement is performed in computations. We evaluate various design choices inherent in the formulation of the method and conclude with recommendations concerning selection of the best options. Numerical tests carried out in the finite-element setting based on Lagrangian piecewise quadratic space discretization demonstrate that the proposed RCG method outperforms the simple gradient descent RG method in terms of rate of convergence. While on simple problems a Newton-type method implemented in the `Ipopt` library exhibits a faster convergence than the RCG approach, the two methods perform similarly on more complex problems requiring the use of mesh adaptation. At the same time the RCG approach has far fewer tunable parameters. Finally, the RCG method is extensively tested by computing complicated vortex configurations in rotating Bose–Einstein condensates, a task made challenging by large values of the nonlinear interaction constant and the rotation rate as well as by strongly anisotropic trapping potentials.

Key words. Gross–Pitaevskii functional, Bose–Einstein condensate, Riemannian optimization, Sobolev gradients, conjugate gradients

AMS subject classifications. 68Q25, 68R10, 68U05

DOI. 10.1137/17M1121974

1. Introduction.

The rotating Bose–Einstein condensate (BEC) represents a highly controllable quantum system offering an ideal framework to study quantized vortices at a macroscopic level. A rich variety of vortex states, from a single vortex line to dense Abrikosov vortex lattices and giant vortices, were experimentally observed and extensively studied in the last two decades (see, e.g., [56]). A stan-

*Submitted to the journal's Computational Methods in Science and Engineering section March 22, 2017; accepted for publication (in revised form) September 29, 2017; published electronically November 30, 2017.

http://www.siam.org/journals/sisc/39-6/M112197.html

Funding: The first author's work was supported by ANR (France) grant ANR-12-MONU-0007-01 BECASIM. The second author's work was supported by an NSERC (Canada) Discovery Grant. Computational resources were provided by CRIANN (Centre Régional Informatique et d'Applications Numériques de Normandie, France) under the project 2015001.

†Laboratoire de Mathématiques Raphael Salem, Université de Rouen Normandie, Rouen 76801, France (ionut.danaila@univ-rouen.fr, http://lmrs.univ-rouen.fr/Persopage/Danaila/).

‡Department of Mathematics & Statistics, McMaster University, Hamilton, ON L85 4K1, Canada (bprotas@mcmaster.ca, http://www.math.mcmaster.ca/bprotas).

dard mathematical approach to describe equilibrium configurations with quantized vortices in rotating BEC is the minimization of the Gross–Pitaevskii (GP) energy functional with rotation [49, 41]. In addition to the global minima, which are the so-called "ground states" local minima are also of interest as they represent excited, or meta-stable, states which are more likely to be observed in experiments [20]. The minimizers have the form of complex-valued wavefunction fields dependent on the space variable, resulting in an infinite-dimensional minimization problem. The complexity of the minimization problem is further compounded by a constraint imposed on the L^2 norm of the minimizers which reflects the conservation of the number of atoms in the condensate. For the mathematical properties of the GP energy with rotation and the corresponding ground states, we refer the reader to [3, 6, 41, 15].

In this paper we address the problem of direct minimization of the GP energy functional with rotation when large nonlinear interaction constants and high rotation frequencies are considered. A number of approaches to direct minimization of the GP energy have been proposed based on various standard and emerging mathematical methods for optimization problems in finite dimension: the optimal damping algorithm [31, 38], a Newton-like method based on sequential quadratic programming (SQP) [21], the interior point method (Ipopt) [57], the inertial proximal algorithm (iPiano) [10], and a regularized Newton method with trust region [60]. Alternative approaches which do not involve direct minimization of energy rely instead on the solution of the corresponding Euler–Lagrange system, which has the form of a nonlinear eigenvalue problem. In the latter context, a wide variety of classical integration and iterative techniques have been employed such as the Newton [18], Runge–Kutta [22], and continuation methods [24]. We also mention the "deflated" Newton method, recently proposed in [25], which represents a systematic approach to determine several distinct solution branches.

Another class of approaches, pioneered in [16], relies on a normalized gradient flow for the GP functional and became popular due to their efficiency and ease of implementation (see also the review papers [13, 15, 9, 14]). These methods consist in first solving the gradient flow equation for the minimization of an unconstrained energy followed by a normalization of this "predictor" solution to bring it back to the constraint manifold. Solution of the gradient flow equation can be viewed as a pseudotime (or imaginary time) integration of the time-independent GP equation. Discretization of the gradient-flow equations using a (natural) steepest descent method would result in a very inefficient explicit Euler scheme for the (imaginary-)time integration. For this reason, in [16] the gradient-flow equation was solved using a semi-implicit backward Euler scheme, which proved even more efficient than the classical Crank–Nicolson scheme. The convergence of the original scheme suggested in [16] was recently improved in [11, 12] by using different discrete preconditioners. It is interesting to note that the gradient-flow equation for the GP functional has a structure similar to the complex-valued heat equation which makes it amenable to solution with different classical time-integration schemes such as Runge–Kutta–Fehlberg [34], backward Euler [16, 6, 17], second-order Strang time-splitting [16, 6], and combined Runge–Kutta–Crank–Nicolson [4, 5, 28].

As regards the development of numerical methods, there are two main paradigms, namely, "optimize-then-discretize" and "discretize-then-optimize" depending on whether the gradient expressions are derived at the continuous or discrete level. In the first case, the Sobolev gradient approach [44] represents the gradient-flow method formulated with respect to a judiciously selected inner product in a Hilbert space rather than the classical L^2 inner product. The required gradients are obtained via the Riesz representation theorem. When discretized, the Sobolev gradient approach can also be interpreted as suitable preconditioning applied to the L^2 gradient [32]. However, the key advantage of working with the optimize-then-discretize formulation is that the form of this preconditioning is dictated by the functional (Sobolev) setting of the problem and thus avoids the technically complicated search for a good (and mesh-independent) discrete preconditioner. Sobolev gradient methods were successfully applied to minimize the GP energy in the presence of rotation in [34, 30].

The purpose of this contribution is to develop and validate an efficient computational approach to minimization of the GP energy by combining the Sobolev gradient method with concepts of Riemannian optimization [1, 54]. This allows us to transform the original constrained optimization problem into an unconstrained problem on a Riemannian manifold with a very simple structure, which is amenable to solution using the conjugate gradient approach. We remark that while the "Riemannian structure" of an optimization problem may be exploited at various levels, in the present study we will focus solely on the basic concepts of "retraction" and "vector transport" that describe how information travels along a manifold, and we will not consider endowing the constraint manifold with a Riemannian metric. In other words, we will assume that the constraint manifold is equipped with the metric induced by the embedding space. We begin by formulating a Riemannian version of the Sobolev gradient approach in which the retraction operation ensures that the norm constraint is satisfied at all discrete times. Then, convergence is accelerated using a Riemannian version of the conjugate gradient method, which relies on the notion of the vector transport applied to the gradient and the descent direction. Such approaches are already well established in the context of problems formulated in finite dimensions [19] but have received only limited attention in the context of problems in infinite dimensions. Convergence of the Riemannian versions of the BFGS quasi-Newton approach and of the Fletcher–Reeves conjugate gradient method was established in [51], whereas some applications were considered in [7, 48, 8, 43]. To the best of our knowledge, these approaches represent a new direction as regards minimization of the GP energy. In our study, we carefully evaluate various design choices inherent in the formulation of the method and make recommendations concerning selection of the best options. Then, we demonstrate that in combination with a flexible finite-element discretization involving adaptive grid refinement [29, 57], the proposed approach outperforms a number of first-order techniques and performs on par with a Newton-type method implemented in the `Ipopt` library [59].

The structure of the paper is as follows. In the next section we state the mathematical model describing minimization of the GP energy; in section 3 we recall the Sobolev gradient method with its projected gradient variant, whereas the Riemannian gradient and conjugate gradient methods are introduced in section 4; numerical discretization based on adaptive finite elements and its software implementation are discussed in section 5; design choices to be explored in the formulation of the method are identified in section 6; in section 7 we use the method of the manufactured solutions to estimate the speeds of convergence of the different approaches; in section 8 we compute a number of challenging BEC configurations with vortices in various arrangements; and discussions and conclusions are deferred to section 9.

2. Mathematical model.

The energy of a rotating homogeneous BEC at zero temperature is given in terms of the GP energy functional [49, 41]. After application of standard scaling and dimension reduction [49, 15], its nondimensional form defined here on a two-dimensional domain $\mathscr{D} \subset \mathbb{R}^2$ becomes

$$E(u) = \int_{\mathscr{D}} \left[\frac{1}{2} |\boldsymbol{\nabla} u|^2 + C_{\text{trap}} \, |u|^2 + \frac{1}{2} C_g |u|^4 - i C_\Omega \, u^* \mathbf{A}^t \cdot \boldsymbol{\nabla} u \right] \, d\mathbf{x}, \tag{1}$$

where $u \ : \ \mathscr{D} \to \mathbb{C}$ is a complex-valued wavefunction and u^* is its complex conjugate, $\mathbf{A}^t = (y, -x)$, and $C_{\text{trap}} \ : \ \mathscr{D} \to \mathbb{R}$ is the trapping potential. C_g and C_Ω are real constants characterizing the strength of the nonlinear interactions and rotation frequency, respectively. The wavefunction u vanishes outside the trap and is therefore assumed to satisfy the homogeneous Dirichlet boundary conditions $u = 0$ on $\partial \mathscr{D}$. The conservation of the number of atoms in the condensate is expressed as

$$\|u\|_2 := \|u\|_{L^2(\mathscr{D}, \mathbb{C})} = \sqrt{\int_{\mathscr{D}} |u(\mathbf{x})|^2 \, d\mathbf{x}} = 1 \tag{2}$$

and serves as a constraint on u. For the energy functional (1) to be well-defined, the wavefunction u must belong to the Sobolev space $H_0^1(\mathscr{D}, \mathbb{C})$ of functions with square-integrable gradients [2] and vanishing traces on the boundary (precise definitions of the norms in this function space will be provided in section 3). The constraint (2) may now be interpreted as defining a manifold $\mathcal{M}$ in the solution space, i.e.,

$$\mathcal{M} := \left\{ u \in H_0^1(\mathscr{D}, \mathbb{C}) \ : \ \|u\|_2 = 1 \right\}. \tag{3}$$

We assume the trapping potential has the following general form allowing us to represent different trapping potentials used in experiments:

$$C_{\text{trap}}(x, y) = \frac{1}{2} \left(a_x x^2 + a_y y^2 + a_4 r^4 \right), \quad r^2 = (x^2 + y^2), \tag{4}$$

for some $a_x, a_y, a_4 \in \mathbb{R}$. Along with the energy (1), another important integral quantity describing the rotating BEC is the total angular momentum

$$L := L_z = i \int_{\mathscr{D}} u^* \mathbf{A}^t \cdot \boldsymbol{\nabla} u \, d\mathbf{x}, \tag{5}$$

which under the assumed homogeneous Dirichlet boundary conditions is real-valued.

Global minimizers of the energy functional (1) defined through

$$u_g = \arg \min_{u \in \mathcal{M}} E(u), \quad E(u_g) < \infty, \tag{6}$$

are called ground states. Local minimizers, with energy larger than that of the ground state, are referred to as excited or meta-stable states. The Euler–Lagrange system corresponding to the minimization problem (6) is derived using standard techniques and leads to the stationary GP equation

$$-\frac{1}{2} \nabla^2 u + C_{\text{trap}} u + C_g |u|^2 u - i C_\Omega \mathbf{A}^t \cdot \boldsymbol{\nabla} u = \mu \, u \qquad \text{in } \mathscr{D}, \tag{7a}$$

$$u = \ 0 \quad \text{on } \partial \mathscr{D}, \tag{7b}$$

$$\|u\|_2 = \ 1. \tag{7c}$$

The ground state and excited states are therefore eigenfunctions of the nonlinear eigenvalue problem (7).

3. Gradient flows and steepest descent Sobolev gradient methods. Numerical techniques for the solution of optimization problem (6) can be derived from a form of the gradient-flow equation which, for practical reasons, we state here in terms of the gradient of the energy functional (1) rather than the gradient of the corresponding Lagrangian functional

$$\begin{aligned} \frac{du}{dt} &= -\nabla_X E(u), \qquad t > 0, \\ u(0) &= u_0, \end{aligned} \tag{8}$$

where $u_0 \in \mathcal{M}$ is an initial guess and $\nabla_X E(u)$ represents the gradient of the GP energy functional (1) at u, computed with respect to the topology of the Hilbert space X (to be made specific below). The gradient flow needs to be additionally constrained to ensure that $u(t) \in \mathcal{M}$ for $t > 0$. This approach is similar to the so-called *normalized gradient-flow method* [16], which first evolves (8) and then projects the intermediate solution back onto the manifold. It can be viewed as a splitting method for solving the *continuous normalized gradient-flow equation* [16], which is the constrained version of problem (8) in which the gradient $\nabla_X E(u)$ is replaced with the gradient of the corresponding Lagrangian.

As shown below, many different computational approaches can be derived from (8) by making specific choices regarding (i) the Hilbert space X, (ii) discretization of the initial-value problem (8) with respect to pseudotime t, and (iii) how the constraint $u(t) \in \mathcal{M}$ is imposed.

As regards the expression of the gradient $\nabla_X E(u)$, it can be derived from the Gâteaux differential of energy (1) using the Riesz representation theorem [42], which depends on the choice of the inner-product space X. Since energy (1) is a twice continuously differentiable function from $H_0^1(\mathcal{D}, \mathbb{C})$ to $\mathbb{R}$, a natural choice for the inner product that will ensure the existence of a gradient is

$$\langle u, v \rangle_{H^1} = \int_{\mathcal{D}} (u, v) + (\boldsymbol{\nabla} u, \boldsymbol{\nabla} v)\, d\mathbf{x}, \tag{9}$$

where $(u, v) = uv^*$ is the complex ($\mathbb{C}$ or $\mathbb{C}^2$) inner product. The following new inner product equivalent (in the precise sense of norm equivalence) to (9) was suggested in [30]:

$$\langle u, v \rangle_{H_A} = \int_{\mathcal{D}} (u, v) + (\boldsymbol{\nabla}_A u, \boldsymbol{\nabla}_A v)\, d\mathbf{x}, \quad \boldsymbol{\nabla}_A = \boldsymbol{\nabla} + iC_\Omega \mathbf{A}, \tag{10}$$

and this will be adopted in our considerations below. Its definition was motivated by the following physically more revealing form of the energy functional equivalent to (1):

$$E(u) = \int_{\mathcal{D}} \left[\frac{1}{2} \left| \boldsymbol{\nabla} u + iC_\Omega\, \mathbf{A} u \right|^2 + C_{\text{trap}}^{\text{eff}}\, |u|^2 + \frac{1}{2} C_g |u|^4 \right] d\mathbf{x}, \tag{11}$$

where the effective nondimensional trapping potential $C_{\text{trap}}^{\text{eff}}$ is obtained from the original potential by subtracting a term representing the centrifugal force [55],

$$C_{\text{trap}}^{\text{eff}}(x, y) = C_{\text{trap}}(x, y) - \frac{1}{2} C_\Omega^2\, r^2. \tag{12}$$

We add that, since the solution space $H_0^1(\mathcal{D}, \mathbb{C})$ is a subspace of both $H^1(\mathcal{D}, \mathbb{C})$ and $H_A(\mathcal{D}, \mathbb{C})$, we will assume $H_0^1(\mathcal{D}, \mathbb{C})$ to be equipped with the inner product (9) or (10), and the notation $X = H^1$ or $X = H_A$ will make it clear which one is used.

For each $u \in X$, one can find an element of X, denoted $\nabla_X E(u)$, such that the directional Gâteaux derivative of the energy at u in the direction v, which is a continuous linear functional from X to $\mathbb{R}$, is expressed as

$$E'(u)v = \Re\left(\langle \nabla_X E(u), v\rangle_X\right) \quad \forall v \in X, \tag{13}$$

where $\Re(\cdot)$ denotes the real part of a complex number. We refer to such an element of X as the gradient of E at u. Computing the Gâteaux derivative of the energy functional (1), we obtain

$$E'(u)v = 2\Re\left(\int_{\mathcal{D}} \left[\frac{1}{2}(\boldsymbol{\nabla} u, \boldsymbol{\nabla} v) + \left(C_{\text{trap}} u + C_g|u|^2 u - iC_\Omega \mathbf{A}^t \cdot \boldsymbol{\nabla} u, v\right)\right] d\mathbf{x}\right), \tag{14}$$

which, together with (13), allows us to identify the gradient $\nabla_X E(u)$. In particular, the H_A gradient, hereafter denoted $G = \nabla_{H_A} E(u)$, will be computed by solving the elliptic boundary-value problem resulting from (14), (13), and (10), which we state here in the equivalent weak form

$$\begin{aligned}\forall v \in H_0^1(\mathcal{D}, \mathbb{R}), \int_{\mathcal{D}} &\left[\left(1 + C_\Omega^2(x^2+y^2)\right)(G, v) + (\boldsymbol{\nabla} G, \boldsymbol{\nabla} v) - 2iC_\Omega(\mathbf{A}^t \cdot \boldsymbol{\nabla} G, v)\right] d\mathbf{x} \\ &= 2\int_{\mathcal{D}} \left[\frac{1}{2}(\boldsymbol{\nabla} u, \boldsymbol{\nabla} v) + \left(C_{\text{trap}} u + C_g|u|^2 u - iC_\Omega \mathbf{A}^t \cdot \boldsymbol{\nabla} u, v\right)\right] d\mathbf{x}.\end{aligned} \tag{15}$$

3.1. Normalized gradient flow.

We note from (13) and (14) that, in order for $E'(u)$ to be bounded in the $L^2(\mathcal{D}, \mathbb{C})$ norm, stronger regularity assumptions must be imposed on the solution u, namely, $u \in H^2(\mathcal{D}, \mathbb{C})$. In this case, from (14) we obtain that

$$E'(u)v = 2\Re\left(\int_{\mathcal{D}} \left(-\frac{1}{2}\nabla^2 u + C_{\text{trap}} u + C_g|u|^2 u - iC_\Omega \mathbf{A}^t \cdot \boldsymbol{\nabla} u, v\right) d\mathbf{x}\right), \tag{16}$$

which allows us to formally derive an "L^2 gradient" corresponding to the L^2 inner product,

$$\langle u, v\rangle_{L^2} = \int_{\mathcal{D}} (u, v)\, d\mathbf{x}. \tag{17}$$

We add that this is the expression appearing on the left-hand side of the Euler–Lagrange equation (7a). The gradient flow equation (8) with this L^2 gradient was discretized in [16] using a semi-implicit backward Euler (BE) method,

$$\frac{\tilde{u} - u_n}{\delta t} = \frac{1}{2}\nabla^2 \tilde{u} - C_{\text{trap}}\tilde{u} - C_g|u_n|^2\tilde{u} + iC_\Omega \mathbf{A}^t \cdot \boldsymbol{\nabla}\tilde{u}, \tag{18}$$

where $u_n = u(t_n)$ denotes the approximation obtained at the nth discrete time level, $\tilde{u} = \tilde{u}(t_{n+1})$ is an intermediate (predictor) field, and δt is a fixed (pseudo)time step. The approximation u_{n+1} at the time level t_{n+1} has to satisfy the unit-norm constraint (2) and is therefore obtained by normalizing the predictor solution as

$$u_{n+1} = \frac{\tilde{u}(t_{n+1})}{\|\tilde{u}(t_{n+1})\|_2}. \tag{19}$$

This approach is referred to as the *normalized gradient-flow* method (see also [13, 15, 9, 14]). Different existing variants of this method use various numerical approaches to integrate the gradient-flow equation (8), e.g., Runge–Kutta methods [34, 4, 5, 28], different implicit schemes [16, 6, 17], and Strang-type time-splitting approaches [16, 6]. Even though some of these schemes possess the energy-decreasing property, they typically do not preserve the gradient-flow structure at the discrete level, in the sense that the expression on the right-hand side (RHS) of the gradient-flow equation (8) discretized with respect to the pseudotime t is no longer in the form of a gradient of $E(u)$ (this is because, as a result of the hybrid explicit/implicit treatment, different terms in this expression may depend on the state u approximated at different time levels, as happens in (18)). Therefore, such imaginary-time methods can be regarded as solving the nonlinear eigenvalue problem (7) rather than directly minimizing the GP energy. Another potential drawback of such approaches is that solutions of (7) are in general critical points of the energy which are not necessarily minima.

3.2. Steepest descent Sobolev gradient methods. Hereafter we focus on techniques which preserve the gradient-flow structure of (8) on the discrete level while explicitly accounting for the presence of the unit-norm constraint (2). As a starting point, we will thus consider an explicit discretization of (8) in the following generic form:

$$u_{n+1} = u_n - \tau_n\, G_n, \quad n = 0, 1, \dots, \tag{20}$$

where τ_n is a suitable descent step size, whereas $G_n = G(u_n) = \nabla_X E(u_n)$ is a Sobolev gradient defined for $X = H_A$ or H^1. Below we discuss two ways in which the information about the constraint $u \in \mathcal{M}$ can be incorporated into the gradient method.

3.2.1. Projected Sobolev gradient method. By considering the following identity derived from (20),

$$\|u_{n+1}\|_2^2 = \|u_n - \tau_n\, G_n\|_2^2 = \|u_n\|_2^2 - 2\tau_n \Re\langle u_n, G_n\rangle_{L^2} + \tau_n^2 \|G_n\|_2^2, \tag{21}$$

we note that using an unconstrained gradient G_n leads to an $\mathcal{O}(\tau_n)$ error in the satisfaction of the constraint (2) at each iteration. Normalization of the solution is then necessary to bring it back onto the manifold $\mathcal{M}$ (see Figure 1a).

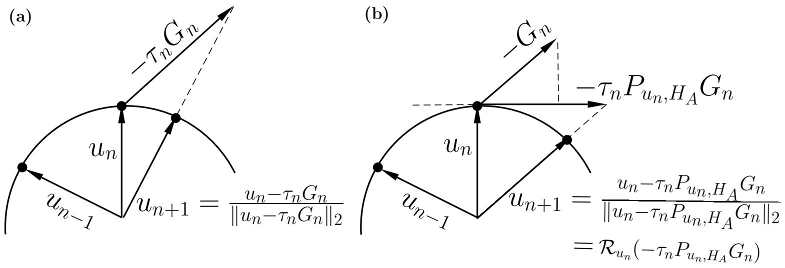
(a)
$-\tau_n G_n$
u_n
u_{n-1}
$u_{n+1} = \frac{u_n - \tau_n G_n}{\|u_n - \tau_n G_n\|_2}$
(b)
$-G_n$
$-\tau_n P_{u_n,H_A} G_n$
u_n
u_{n-1}
$u_{n+1} = \frac{u_n - \tau_n P_{u_n,H_A} G_n}{\|u_n - \tau_n P_{u_n,H_A} G_n\|_2}$
$= \mathcal{R}_{u_n}(-\tau_n P_{u_n,H_A} G_n)$

FIG. 1. *Schematic illustration of the principle of the steepest descent method on the spherical manifold $\mathcal{M}$ for* (a) *the simple ("unprojected") gradient method and* (b) *the projected gradient (PG) and the Riemannian gradient (RG) methods. In case* (b)*, since the projected gradient $P_{u_n,H_A} G_n$ belongs to the subspace $\mathcal{T}_{u_n}\mathcal{M}$ tangent to the manifold at u_n, normalization is equivalent to Riemannian retraction* (27).

This error can be reduced to second-order $\mathcal{O}(\tau_n^2)$ by requiring that $\langle u_n, G_n\rangle_{L^2} = 0$, which is achieved by projecting the gradient G_n onto the subspace

$$\mathcal{T}_{u_n}\mathcal{M} = \left\{v \in H_0^1(\mathcal{D}, \mathbb{C}) \; : \; \langle u_n, v\rangle_{L^2} = 0\right\} \tag{22}$$

tangent to the constraint manifold $\mathcal{M}$ at u_n. As was shown in [30, 40], the associated projection operator $P_{u_n,X}$ can be expressed in the general form

$$P_{u_n,X} G_n = G_n - \lambda\, v_X, \qquad \lambda = \frac{\Re\langle u_n, G_n\rangle_{L^2}}{\Re\langle u_n, v_X\rangle_{L^2}}, \tag{23}$$

where v_X is a solution of the variational problem

$$\langle v_X, v\rangle_X = \langle u_n, v\rangle_{L^2} \quad \forall v \in X. \tag{24}$$

We note that if $X = L^2$, $v_X = u$ in (24), and we recover the well-known explicit expression of the projected L^2 gradient (see, e.g., [7]).

Hereafter we will set $X = H_A$ and denote $G_n = \nabla_{H_A} E(u_n)$. Replacing G_n with $P_{u_n,H_A} G_n$ in (20), we obtain the *projected gradient (PG) method* suggested in [30]:

$$\text{(PG)} \qquad u_{n+1} = u_n - \tau_n \; P_{u_n,H_A} G_n, \quad n = 0, 1, \ldots. \tag{25}$$

While in [30] a fixed step size $\tau_n = \tau > 0$ was used, here we use an optimal step size found through the solution of a line-minimization problem,

$$\tau_n = \operatorname{argmin}_{\tau>0} E\left(u_n - \tau\, P_{u_n,H_A} G_n\right). \tag{26}$$

An explicit expression for the optimal descent step was derived in [57] based on a particular form of the GP energy. In this study, we prefer to solve problem (26) with a general line-minimization approach such as Brent's algorithm [50, 46] as it has the advantage of being easily adapted to the RG methods presented in the next section. To mitigate the $\mathcal{O}(\tau_n^2)$ drift away from the constraint manifold $\mathcal{M}$ allowed by the PG iterations, normalization analogous to (19) may be applied to the iterates u_n after a certain number of steps. The idea of the PG approach is illustrated schematically in Figure 1b.

4. Riemannian optimization.

In this section we discuss some basic concepts relevant to optimization on manifolds, known as Riemannian optimization [1, 54]. In contrast to the perspective developed in the previous section, here we pursue a different, "intrinsic" approach where optimization is performed directly on the manifold $\mathcal{M}$. The main advantage of such a formulation is that it allows one to treat (6) as an *unconstrained* optimization problem creating an opportunity to apply a suitable modification of the conjugate gradient method as an alternative to more traditional constrained approaches such as, e.g., sequential quadratic programming (SQP) for nonlinear optimization.

In addition to the definition of the projection on the tangent space $\mathcal{T}_u\mathcal{M}$ already introduced above (cf. (23)–(24)), we need to introduce two more concepts, namely, the "retraction" (also referred to as "exponential mapping") and the associated "vector transport." While in general these operators can have a rather complicated form, in the present problem, where the constraint manifold $\mathcal{M}$ is given by (3), they can be reduced to fairly simple expressions. We refer the reader to the monograph [1] for additional details concerning the differential-geometry foundations of this approach.

4.1. Riemannian gradient method. Given a tangent vector $\xi \in \mathcal{T}_u\mathcal{M}$, where $u \in \mathcal{M}$ is a state on the manifold, the *retraction* $\mathcal{R}_u : \mathcal{T}_u\mathcal{M} \to \mathcal{M}$ is defined as the operator

$$\mathcal{R}_u(\xi) = \frac{u+\xi}{\|u+\xi\|_2}, \tag{27}$$

where the norm used in the denominator is the same as the norm defining the constraint manifold in (3). We note that for the spherical manifold $\mathcal{M}$, the retraction operator is equivalent to normalization (19) already used in the previous sections.

The Riemannian gradient (RG) method is then obtained by applying relation (27) to the projected gradient $P_{u_n,H_A}G_n$ (cf. (23)–(24)) used in the PG approach, which yields

$$\text{(RG)} \qquad u_{n+1} = \mathcal{R}_{u_n}\left(-\tau_n P_{u_n,H_A}G_n\right), \qquad n = 0,1,\dots. \tag{28}$$

The step size τ_n is found optimally by solving a generalization of the line-minimization problem (26), which uses retraction (27) to constrain the samples to manifold $\mathcal{M}$, i.e.,

$$\tau_n = \operatorname{argmin}_{\tau>0} E\left(\mathcal{R}_{u_n}(-\tau P_{u_n,H_A}G_n)\right). \tag{29}$$

We refer to problem (29) as "arc-minimization." It is solved using a straightforward modification of Brent's algorithm [50, 46]. In addition to application of retraction (27) at every iteration in the latter case, the key difference between the PG and RG approaches lies in how the optimal step size τ_n is determined; cf. (26) vs. (29). The idea of the RG method is illustrated schematically in Figure 1b.

4.2. Riemannian conjugate gradient method. As a point of reference, we begin by recalling the nonlinear conjugate gradient method in the Euclidean case. Given a function $f : \mathbb{R}^N \to \mathbb{R}$, this approach finds its local minimum $\overline{\mathbf{u}}$ as $\overline{\mathbf{u}} = \lim_{n\to\infty} \mathbf{u}_n$, with the iterates $\mathbf{u}_n$ defined as [46]

$$\mathbf{u}_{n+1} = \mathbf{u}_n + \tau_n\, \mathbf{d}_n, \qquad n = 0,1,\dots, \tag{30}$$

where $\mathbf{u}_0$ is the initial guess and the descent direction $\mathbf{d}_n$ is constructed as

$$\begin{aligned} \mathbf{d}_0 &= -\mathbf{g}_0, \\ \mathbf{d}_n &= -\mathbf{g}_n + \beta_n\, \mathbf{d}_{n-1}, \qquad n = 1,2,\dots, \end{aligned} \tag{31}$$

in which $\mathbf{g}_n = \boldsymbol{\nabla} f(\mathbf{u}_n)$ and β_n is a "momentum" term chosen to enforce the conjugacy of the search directions $\mathbf{d}_k$, $k = 1,\dots,n$. When the objective function is convex-quadratic, i.e., $f(\mathbf{u}) = \mathbf{u}^T\mathbf{A}\mathbf{u}$ for some positive-definite matrix $\mathbf{A} \in \mathbb{R}^{N\times N}$, approach (30)–(31) reduces to the "linear" conjugate gradient method in which τ_n and β_n are given in terms of simple expressions involving $\mathbf{A}$ [46]. In the nonquadratic setting, which is the case of problem (6), the step size τ_n needs to be found via line minimization as described by (29), whereas the momentum term is typically computed using one of the following expressions:

$$\beta_n = \beta_n^{FR} := \frac{\langle \mathbf{g}_n, \mathbf{g}_n\rangle_{\mathcal{Y}}}{\langle \mathbf{g}_{n-1}, \mathbf{g}_{n-1}\rangle_{\mathcal{Y}}} \qquad \text{(Fletcher–Reeves)}, \tag{32a}$$

$$\beta_n = \beta_n^{PR} := \frac{\langle \mathbf{g}_n, (\mathbf{g}_n - \mathbf{g}_{n-1})\rangle_{\mathcal{Y}}}{\langle \mathbf{g}_{n-1}, \mathbf{g}_{n-1}\rangle_{\mathcal{Y}}} \qquad \text{(Polak–Ribière)}, \tag{32b}$$

where $\langle \cdot, \cdot \rangle_{\mathcal{Y}}$ is the inner product defined with respect to the metric $\mathcal{Y}$ (in the simplest case when $\mathbf{u} \in \mathbb{R}^N$, $\langle \mathbf{a}, \mathbf{b} \rangle_{\mathcal{Y}} = \mathbf{a}^T \mathbf{b}$ for $\mathbf{a}, \mathbf{b} \in \mathbb{R}^N$). The coefficient β_n may be periodically reset to zero, which is known to improve convergence for convex, nonquadratic problems [46]. It is well known that for optimization problems which are locally quadratic the conjugate gradient approach exhibits faster (though still linear) convergence than the convergence characterizing the simple gradient method, especially for poorly scaled problems [46]. Similar observations have also been reported for Riemannian optimization problems [1, 54].

We explain below how the conjugate gradient approach can be adapted to the Riemannian case involving the energy functional (1) defined for infinite-dimensional state variables $u \in H_0^1(\mathcal{D})$. There are two key issues which must be addressed:

1. The two terms on the RHS of formula (31) belong to two different linear spaces, which are the tangent spaces constructed at two consecutive iterations, i.e., $\mathbf{g}_n \in \mathcal{T}_{\mathbf{u}_n}\mathcal{M}$ and $\mathbf{d}_{n-1} \in \mathcal{T}_{\mathbf{u}_{n-1}}\mathcal{M}$; as a result, they cannot be simply added; the same problem also concerns the inner-product expressions in the numerator of the Polak–Ribière momentum term (32b).
2. While in the finite-dimensional setting all norms are equivalent, this is no longer the case in the infinite-dimensional setting, where the choice of the metric does play a significant role; in our approach, although the gradient descent equations are discretized in space for the purpose of the numerical solution, their specific form is derived in the infinite-dimensional setting (in other words, we follow the "optimize-then-discretize" paradigm [35]); in addition to the momentum term (32), the choice of the metric implied by the inner product also plays a role in the construction of the projection (23)–(24) and the vector transport, which will be defined below.

The key concept required in order to address the first issue is the *vector transport,*

$$\mathcal{T}\mathcal{M} \oplus \mathcal{T}\mathcal{M} \to \mathcal{T}\mathcal{M} \ : \ (\eta, \xi) \longmapsto \mathcal{T}_{\eta}(\xi) \in \mathcal{T}\mathcal{M}, \tag{33}$$

where $\mathcal{T}\mathcal{M} = \cup_{x \in \mathcal{M}} \mathcal{T}_x\mathcal{M}$ is the tangent bundle, describing how the vector field ξ is transported along the manifold $\mathcal{M}$ by the field η [1]. It therefore generalizes the concept of the parallel translation to the motion on the manifold and is also closely related to the "affine connection," which is one of the key differential-geometric quantities characterizing a manifold. The vector transport thus provides a map between the tangent spaces $\mathcal{T}_{u_{n-1}}\mathcal{M}$ and $\mathcal{T}_{u_n}\mathcal{M}$ obtained at two consecutive iterations, so that algebraic operations can be performed on vectors belonging to these subspaces.

In general, vector transport is not defined uniquely, and in the present case when the manifold is a sphere, the following two definitions lead to expressions particularly simple from the computational point of view. Let $u \in \mathcal{M}$ and $\eta_u, \xi_u \in \mathcal{T}_u\mathcal{M}$; the transport of ξ_u by η_u can be expressed by

- either vector transport via differentiated retraction,

$$\mathcal{T}_{\eta_u}(\xi_u) = \frac{d}{dt}\mathcal{R}_u(\eta_u + t\xi_u)\big|_{t=0} = \frac{1}{\|u + \eta_u\|}\left[\xi_u - \frac{\langle u + \eta_u, \xi_u \rangle}{\|u + \eta_u\|^2}(u + \eta_u)\right], \tag{34}$$

- or vector transport using Riemannian submanifold structure,

$$\mathcal{T}_{\eta_u}(\xi_u) = P_{\mathcal{R}_u(\eta_u)}\xi_u = \left[\xi_u - \frac{\langle u + \eta_u, \xi_u \rangle}{\|u + \eta_u\|^2}(u + \eta_u)\right], \tag{35}$$

where P_u is the orthogonal projector on $\mathcal{T}_u\mathcal{M}$. We note that these formulas differ only by a scalar factor $\|u + \eta_u\|^{-1}$, and expression (35) can be interpreted as a Riemannian

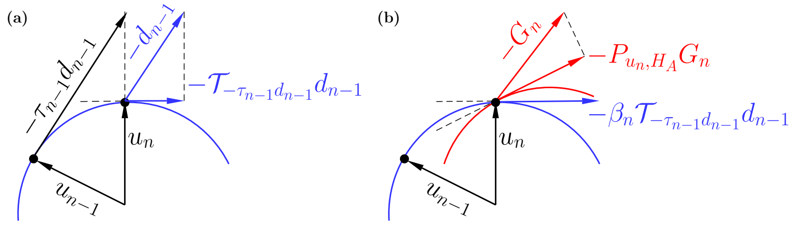
(a)
$-\tau_{n-1}d_{n-1}$
$-d_{n-1}$
$-\mathcal{T}_{-\tau_{n-1}d_{n-1}}d_{n-1}$
u_n
u_{n-1}
(b)
$-G_n$
$-P_{u_n,H_A}G_n$
$-\beta_n\mathcal{T}_{-\tau_{n-1}d_{n-1}}d_{n-1}$
u_n
u_{n-1}

FIG. 2. *Schematic illustration of the principle of Riemannian conjugate gradient (RCG) method on a spherical manifold.* (a) *Riemannian vector transport of the anterior conjugate direction d_{n-1}; the transport of the anterior gradient G_{n-1} is performed in a similar way.* (b) *Projection of the new Sobolev gradient G_n onto the tangent subspace $\mathcal{T}_{u_n}\mathcal{M}$ resulting in $P_{u_n,H_A}G_n$. The linear combination* (37) *of $P_{u_n,H_A}G_n$ and the transported anterior direction are computed in $\mathcal{T}_{u_n}\mathcal{M}$.*

parallel transport. We further remark that the vector transport $\mathcal{T}_{\eta_u}(\xi_u)$ is linear in the field ξ_u but not in η_u. The reader is referred to monograph [1] for details concerning the derivation of formulas (34)–(35). The numerical results presented in sections 7 and 8 are obtained using the vector transport expression (34) or (35) with the L^2 inner product and norm. This choice of metric is dictated by the norm defining the constraint manifold; cf. (2).

Finally, the conjugate gradient method (30)–(32) can be rewritten in the Riemannian infinite-dimensional setting as

$$\text{(RCG)} \qquad u_{n+1} = \mathcal{R}_{u_n}\left(-\tau_n\, d_n\right), \qquad n = 0, 1, \dots, \tag{36}$$

where

$$\begin{aligned} d_0 &= -P_{u_0,H_A}G_0, \\ d_n &= -P_{u_n,H_A}G_n + \beta_n\, \mathcal{T}_{-\tau_{n-1}d_{n-1}}(d_{n-1}), \qquad n = 1, 2, \dots, \end{aligned} \tag{37}$$

with the Polak–Ribière momentum term modified as follows (the corresponding term in the Fletcher–Reeves approach remains unchanged):

$$\beta_n = \beta_n^{PR} := \frac{\Big\langle P_{u_n,H_A}G_n, \left(P_{u_n,H_A}G_n - \mathcal{T}_{-\tau_{n-1}d_{n-1}}P_{u_{n-1},H_A}G_{n-1}\right)\Big\rangle_{H_A}}{\Big\langle P_{u_{n-1},H_A}G_{n-1}, P_{u_{n-1},H_A}G_{n-1}\Big\rangle_{H_A}}. \tag{38}$$

The optimal descent step τ_n in (36) is computed as in (29) by solving the corresponding arc-minimization problem

$$\tau_n = \operatorname{argmin}_{\tau>0} E\left(\mathcal{R}_{u_n}(-\tau d_n)\right) \tag{39}$$

using a generalization of Brent's method. We refer to approach (36)–(39) as the *Riemannian conjugate gradient (RCG) method*, and its idea is schematically illustrated in Figure 2.

5. Space discretization. We use a finite-element approximation constructed as follows. Let T_h be a family of triangulations of the domain $\mathcal{D}$ parametrized by the mesh size $h > 0$. We assume that T_h is a regular family in the sense of Ciarlet [26], with h belonging to a generalized sequence converging to zero. We denote by $P^l(T)$ the space of polynomial functions of degree not exceeding $l \geq 1$ defined on triangles

$T \in T_h$. We also introduce the finite-element approximation spaces

$$W_h^l = \left\{ w_h \in C^0(\bar{\mathscr{D}}_h);\ w_h|_T \in P^l(T)\ \forall T \in T_h \right\}, \tag{40}$$

$$V_h^l = \left\{ w_h \in W_h^l;\ w_h|_{\Gamma_h} = 0 \right\}, \quad \text{where } \Gamma_h = \partial \mathscr{D}_h. \tag{41}$$

The finite-dimensional space V_h^l is a subspace of $H_0^1(\mathscr{D})$ and therefore will be used to approximate the energy functional (1) and the different expressions representing gradients and descent directions in algorithms (PG), (RG), and (RCG). In the following we use P^4 ($l = 4$, piecewise quartic) finite elements to approximate the nonlinear terms in (15) and use the P^2 representation for the remaining terms for evaluation of the GP energy (1) and also to represent the approximate solution u_n. In addition, adaptive mesh refinement suggested in [29] and tested in [29, 57] is used to adapt the grid during iterations, leading to a significant reduction of the computational time. The approach is implemented in `FreeFEM++` [36, 37], where mesh adaptivity relies on metric control [23]. The main idea is to define a metric based on the Hessian and use a Delaunay procedure to build a new mesh such that all the edges are close to the unit length with respect to this new metric. We use the adaptive meshing strategy suggested in [29, 57]. The relative change of the energy of the solution (cf. (45)) is used as an indicator to trigger mesh adaptation in which the metrics are computed simultaneously using the real and imaginary parts of the solution. The implementation of the Riemannian retraction (27) and vector transport (34) or (35) is straightforward and was found to work very well with arc-minimization (39) and adaptive mesh refinement.

6. Design choices inherent in the RCG method.

As is evident from section 4.2, the RCG method offers a number of design choices, which can be exploited to optimize its performance for a specific problem. One of the goals of this study is to evaluate these options in the context of minimization of the GP energy, and we will focus on the following choices, which are the most relevant for the Riemannian aspect of the proposed approach:

(i) Form of the momentum term β_n: Fletcher–Reeves (32a) or Polak–Ribière (32b), with the corresponding variants of the RCG method referred to as (RCG)-(FR) and (RCG)-(PR), respectively.

(ii) Form of the vector transport $\mathscr{T}_{\eta_u}(\xi_u)$: Defined via differentiated retraction (34) or using the Riemannian submanifold structure (35), with the corresponding variants of the RCG method referred to as (RCG)-(VtDR) and (RCG)-(VtRS), respectively; in addition, we will also consider the classical conjugate gradient (CG) method without vector transport (i.e., with $\mathscr{T}_{\eta_u}(\xi_u) = \xi_u$).

Combining these different choices yields six distinct variants, i.e., (RCG)-(FR)-(VtDR), (RCG)-(FR)-(VtRS), (RCG)-(FR), (RCG)-(PR)-(VtDR), (RCG)-(PR)-(VtRS), and (RCG)-(PR). Evidently, there also exist other design choices, but we will not consider them here, because they are less relevant for the Riemannian aspect of the problem and/or have already been considered elsewhere. For example, the choice of the metric X defining the gradient in (13) and the projection in (23)–(24) was extensively analyzed in [30, 29, 57], and here we will exclusively use $X = H_A$, which was found to be the best choice for minimization of the GP energy in the presence of rotation. For the Riemannian operators of retraction and vector transport, we use the L^2 metric naturally induced by the spherical manifold defined in (3).

In principle, we could also consider the frequency of retractions as yet another design parameter; however, since this operation has a negligible cost, it is performed

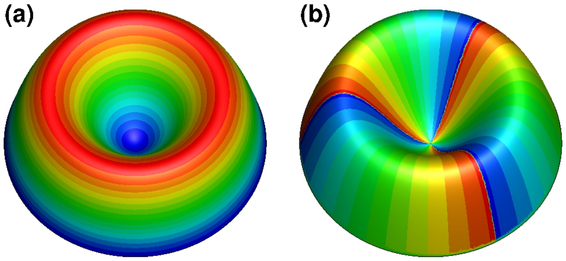

FIG. 3. *Manufactured solution* (42) *visualized with a* 3*D-rendering of the modulus* $|u_{ex}|$ *color-coded with* (a) *the modulus itself and* (b) *the phase of the solution for* $m = 3$.

at every iteration (which is also consistent with the need to reinterpolate the solution once a grid adaptation has taken place). On the other hand, we will consider the effect of periodically resetting the momentum term β_n to zero (cf. section 4.2). In the sections to follow we will analyze these different design choices in order to identify the most robust RCG method capable of handling mesh adaptivity, which in earlier studies [29, 57] was shown to be indispensable for computational efficiency.

7. Convergence speeds of different gradient methods. We start by assessing the convergence speed of the minimization algorithms (PG), (RG), (CG) and the different variants of the (RCG) approach using fixed meshes with different resolutions. We use the method of manufactured solutions [52], a general tool for verification of calculations which has the advantage of providing an exact solution to a modified problem related to the true one. The general idea is that, by introducing an extra source term, the original system of equations is modified to admit an exact solution given by a convenient analytic expression. Even though in most cases exact solutions constructed in this way are not physically realistic, this approach allows one to rigorously verify computations. Here we manufacture such an exact solution in the form

$$u_{ex}(x,y) = U(r)\,\exp(im\theta), \qquad U(r) = \frac{2\sqrt{21}}{\sqrt{\pi}}\,\frac{r^2\,(R-r)}{R^4}, \quad m \in \mathbb{N}, \tag{42}$$

where (r,θ) are the cylindrical coordinates of the point (x,y) and R is the radius of the circular domain $\mathcal{D}$. We note that this solution satisfies constraint (2) and qualitatively resembles a giant vortex in the condensate (see Figure 3 and section 8). It also satisfies an inhomogeneous form of the nonlinear problem (7), i.e.,

$$\begin{aligned} -\frac{1}{2}\nabla^2 u_{ex} + C_{\text{trap}} u_{ex} + C_g |u_{ex}|^2 u_{ex} - iC_\Omega \mathbf{A}^t \cdot \boldsymbol{\nabla} u_{ex} &= f \qquad \text{in } \mathcal{D},\\ u_{ex} &= 0 \quad \text{on } \partial\mathcal{D}, \end{aligned} \tag{43}$$

and is a critical point of the modified energy functional

$$E(u,f) = \int_{\mathcal{D}} \left[\frac{1}{2}|\boldsymbol{\nabla} u|^2 + C_{\text{trap}}\,|u|^2 + \frac{1}{2}C_g|u|^4 - iC_\Omega\, u^*\mathbf{A}^t \cdot \boldsymbol{\nabla} u - (f^*u + fu^*)\right] d\mathbf{x}. \tag{44}$$

For this energy functional, the L^2 gradient is expressed as discussed in section 3.1 but with a supplementary term $-2f$ added. Given the form (42) of the manufactured solution and assuming a harmonic trapping potential $C_{\text{trap}} = r^2/2$, from (43) we obtain $f(r,\theta) = F(r)\ \exp(im\theta)$, where $F(r)$ is a polynomial of degree 9. From this and relations (44) and (5) we can deduce exact expressions for the energy $E_{ex} := E(u_{ex})$ and the angular momentum $L_{ex} := L(u_{ex}) = m$.

The numerical tests are based on the manufactured solution (42) corresponding to the following parameter values: $C_{\text{trap}} = r^2/2$, $C_g = 500$, $R = 1$, $m = 3$, $C_\Omega = 10$, where, to make the problem more challenging, large values of the nonlinear interaction constant C_g and rotation frequency C_Ω are used (cf. Figure 3). The step size τ_n is determined at each iteration by line-minimization (26) for the (PG) method and arc-minimization (29) or (39) for the (RG), (CG), and (RCG) methods.

In order to assess the mesh-independent effect of the Sobolev gradient preconditioning (cf. section 5), we perform computations using two grids: Mesh 1 consisting of 24,454 triangles with $h_{min} = 0.0118$, and Mesh 2 consisting of 99,329 triangles with $h_{min} = 0.0059$, where h_{min} is the smallest grid size. In the present case no mesh refinement was performed during iterations. The initial guess is taken as $u_0 = 0$, whereas iterations are declared converged once the following condition based on the relative energy decrease is satisfied [29, 57]:

$$\varepsilon_E = |E_{n+1} - E_n|/E_n < \varepsilon_{st} = 10^{-12}, \qquad \text{where } E_n := E(u_n). \tag{45}$$

The performance of the approaches corresponding to the different design choices discussed in section 6 is summarized in Table 1, where all computations were performed on Mesh 2. In this and the following tables, the CPU time reflects the computations performed on a Linux workstation with two 3.10GHz Intel Xeon E5-2687w CPUs.

Table 1

Test case based on the manufactured solution (42). *Performance of the gradient methods corresponding to the different design choices (cf. section* 6*) measured in terms of the number of iterations (iter) and the computational time in seconds (CPU) required for convergence. Note that the total computational time depends both on the number of iterations and the number of evaluations of the energy functional in the line- or arc-minimization procedures.*

Method	iter	CPU
(RCG)-(PR)-(VtRS)	37	1270
(RCG)-(PR)-(VtDR)	38	1326
(CG)-(PR)	38	1529
(RCG)-(FR)-(VtRS)	54	1852
(RCG)-(FR)-(VtDR)	49	1668
(CG)-(FR)	31	1297
(RG)	180	5274
(PG)	219	3104

In the calculations reported in Table 1, for the (CG) and (RCG) methods we did not reset the momentum term β_n to zero. First, we note that all variants of the (RCG) and (CG) methods outperform the simple gradient methods (PG) and (RG). The reduced CPU time of the (PG) method is due to the fact that it implements a line-minimization strategy (26) based on analytical expressions derived from the particular form of the GP energy (see [57] for details). This is not the case for the arc-minimization used by all Riemannian gradient methods, where an adaptation of Brent's algorithm is employed. Second, we observe that the (RCG) algorithm with the Polak–Ribière (PR) momentum term is the least sensitive to the form of the vector transport. On the other hand, the Fletcher–Reeves version of the (RCG) method proves more sensitive to the form of the vector transport and, somewhat surprisingly,

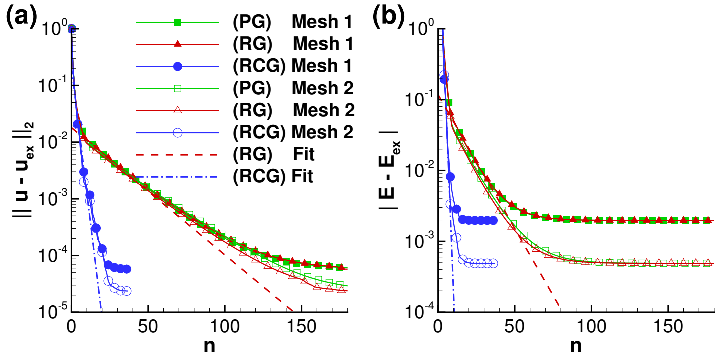

Fig. 4. *Test case based on the manufactured solution* (42). *Convergence of the (PG), (RG), and (RCG) methods.* (a) $\|(u_n - u_{ex})\|_2$ *and* (b) $|E_n - E_{ex}|$ *are shown as functions of iterations* n *for different discretizations. Dashed and dashed-dotted lines indicate the least-squares fits* (46).

the (CG)-(FR) approach (without vector transport) turns out to be the most efficient in terms of the number of iterations (although not in terms of the computational time). The method which converged in the shortest time was the (RCG)-(PR)-(VtRS) approach, and it will be used for further tests in the remainder of this section; for brevity, we will refer to it simply as (RCG).

To assess the speed of convergence of the (PG), (RG), and (RCG) approaches, the quantities $\|u_n - u_{ex}\|_2$ and $|E_n - E_{ex}|$ are shown as functions of iterations n for the two spatial discretizations in Figures 4a and 4b. In these figures we observe linear convergence followed by a slower convergence at final iterations. The change of the slope of error curves occurs at the level at which the minimization errors $(u_n - u_{ex})$ are comparable to the errors related to the spatial discretization. In other words, in the "optimize-then-discretize" setting adopted here, gradient expressions derived based on the continuous formulation (cf. (15)) may no longer accurately represent the sensitivity of the discretized objective function when the difference between u_n and u_{ex} is of the order of the space discretization errors. This is also confirmed by computing the errors $\|u_n - u_{ex}\|_{H^1}$ at which convergence stagnates. These errors drop by a factor of roughly 4 when the mesh is refined such that h_{min} is reduced by approximately one half (cf. Figure 4a), as expected from the well-known error estimates for the finite-element approximation.

In Figures 4a and 4b it is evident that the (RCG) method converges much more rapidly (39 iterations) than the (RG) approach (180 iterations). As expected, the convergence of the (PG) method (202 iterations) is similar to that of the (RG) method. To quantify the convergence rates, we use the following ansatz to represent the errors:

$$\|u_n - u_{ex}\|_2 \sim B_u\, A_u^n, \qquad |E_n - E_{ex}| \sim B_e\, A_e^n. \tag{46}$$

The values of the parameters A_u and A_e, which represent the factors by which the corresponding errors are reduced between two iterations, can be obtained from least-squares fits of the data in Figures 4a and 4b in the linear regime. These results are collected in Table 2 (the corresponding fits are also indicated in Figures 4a and 4b). First, these results demonstrate that the rate and speed of convergence are grid-independent as expected from the general theory of Sobolev gradient descent methods [44]. The data in Table 2 can also be interpreted in terms of the classical theory of the conjugate gradient method in the finite-dimensional Euclidean setting

[46], which for the minimization of quadratic functions predicts that $A_u \approx \sqrt{A_e}$. We see that the data from Table 2 satisfies this relationship with accuracy of a few percent. For the simple gradient and conjugate gradient methods we furthermore have the approximate relationships $A_u = (\kappa - 1)/(\kappa + 1)$ and $A_u = (\sqrt{\kappa} - 1)/(\sqrt{\kappa} + 1)$, respectively, where κ is the "effective" condition number characterizing the problem. It is defined in terms of the condition number of the discrete Hessian of the GP energy (1) at the minimum preconditioned by the metric of the Sobolev space $H_A(\mathscr{D})$ (cf. (10)), in which optimization is performed. Using the data from Table 2 (Mesh 2), we infer that $\kappa = (1 + A_u)/(1 - A_u) \approx 42.37$ for the gradient (RG) and $\kappa \approx 3.2$ for the conjugate gradient (RCG) method, indicating that the convergence acceleration produced in the present problem by the Riemannian conjugate gradient approach actually exceeds what can be expected from the standard theory.

TABLE 2
Parameters characterizing the least-squares fits (46) *of the data shown in Figure* 4.

	Mesh 1			Mesh 2		
	A_e	$\sqrt{A_e}$	A_u	A_e	$\sqrt{A_e}$	A_u
(RG)	0.9167	0.9574	0.9496	0.9268	0.9627	0.9538
(RCG)	0.2909	0.5394	0.5275	0.2924	0.5408	0.5238

Since the exact solution u_{ex} is usually unavailable in physically relevant problems, we now verify that convergence of iterations can be monitored based on quantities which do not involve u_{ex}. Indeed, the evolution of $\|u_{n+1} - u_n\|_2$ and $|E_{n+1} - E_n|$ with iterations n shown in Figures 5a and 5b exhibits the same trends as the data shown in Figures 4a and 4b, except for the slowdown observed in the latter case. This demonstrates that either of these two quantities can be used to monitor convergence and, in particular, check the stopping criterion (see also [16, 13, 15, 9, 14]).

There are two additional aspects of the convergence of the different methods we wish to comment on. In Figure 5c we show the evolution of the "drift" away from the constraint manifold $\mathcal{M}$ exhibited by the intermediate approximations $\hat{u}_n$ *before* retraction (27) is applied,

$$\delta_n = \left|1 - \|\hat{u}_n\|_2^2\right|, \qquad n = 0, 1, \dots. \tag{47}$$

This quantity measures how far the intermediate steps diverge from the constraint manifold. We see that, as compared to the (PG) and (RG) methods, in the (RCG) approach the intermediate approximations always remain closer to $\mathcal{M}$. Finally, the step size τ_n determined by the gradient approaches (PG), (RG), and (RCG) via line-/arc-minimization (cf. (26), (29), and (39)) is shown in Figure 5d. We see that the step sizes generated by the simple gradient methods, (PG) and (RG), tend to oscillate between two values—a behavior indicating that the iterations are trapped in narrow "valleys." This is a common behavior of the steepest descent method when applied to poorly conditioned problems and is not exhibited by the (RCG) iterations where, on average, the steps also tend to be longer. The data in Figures 5a,c,d offer interesting insights into the behavior of iterations in different approaches. It follows from relation (21) that $\|u_{n+1}\|_2^2 - \|u_n\|_2^2 = \tau_n^2 \|G_n\|_2^2$ is satisfied for all cases considered here, whereas in Figure 5d it is evident that the corresponding step sizes τ_n are bounded away from zero. We therefore deduce that the drift δ_n is directly linked to the magnitude of the gradient $\|G_n\|_2$, which is smallest for the approach with the fastest convergence, i.e., the (RCG) method (cf. Figure 5a). This demonstrates that the small drift away from the constraint manifold $\mathcal{M}$ observed in this approach (cf. Figure 5c) is a consequence

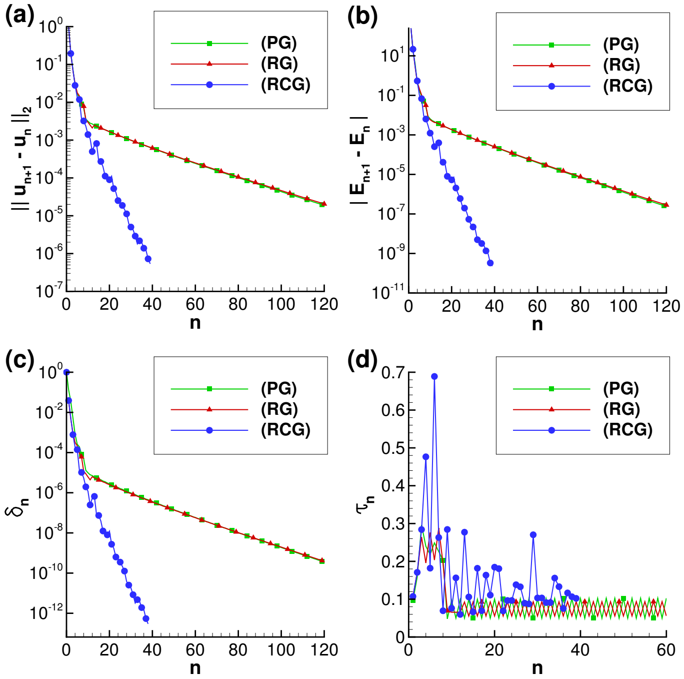

Fig. 5. *Test case based on the manufactured solution* (42). *Convergence of the different quantities with iterations n:* (a) $\|(u_{n+1} - u_n)\|_2$, (b) $|E_{n+1} - E_n|$, (c) δ_n *(cf.* (47)*) is the drift away from the constraint manifold and* (d) τ_n *is the optimal descent step.*

of its rapid convergence. Performance of the different methods applied to several realistic problems will be discussed in the next section.

8. Computation of rotating Bose–Einstein condensates. In this section we compare the performance of the minimization algorithms (PG), (RG), and (RCG) on a number of test cases involving configurations of rotating BEC with vortices. We consider increasingly complex arrangements: a single vortex, Abrikosov vortex lattices with more than 100 vortices, giant vortices, and, finally, condensates in anisotropic trapping potentials. To make these test cases more challenging, we consider large values of the nonlinear interaction constant C_g and large angular frequencies C_Ω. In some cases, we will provide comparisons between the gradient methods and other state-of-the-art techniques, one of which is the (BE) approach (18)–(19) implemented using the same P^2 finite-element setting. In addition, in order to offer a comparison with a higher-order method, we will also solve the minimization problem (6) using the library `Ipopt`, which is interfaced with `FreeFem++`. This approach, which we will refer to as (Ipopt), is based on a combination of an interior point minimization [58], barrier functions [45], and a filter line-search [59]. For problems with equality constraints only (such as the present problem), (Ipopt) reduces to a Newton-like method with an elaborate line-search used to optimally determine the step size. Here we use (Ipopt) to solve the Euler–Lagrange system (7), in the course of which it is provided with the expressions for the L^2 gradient and the Hessian of the GP energy, reformulated by separating the real and imaginary parts of the solution (see also [57]). Computations with (Ipopt) are based on several calls of the library where, for each call, the residual of the optimality condition (7), on which the termination criterion is based, is progressively decreased.

8.1. Test case 1: BEC with a single central vortex. We consider the case of a BEC trapped in a harmonic potential and rotating at low angular velocities: $C_{\text{trap}} = r^2/2$, $C_g = 500$, $C_\Omega = 0.4$. For this case, the Thomas–Fermi (TF) theory [55] offers a good approximation of the atomic density $\rho = |u|^2$ of the condensate $\rho \approx \rho_{\text{TF}} = \big((\mu - C^{\text{eff}}_{\text{trap}})/C_g\big)_+$, with the effective trapping potential $C^{\text{eff}}_{\text{trap}}$ given by (12). By imposing $\int_{\mathscr{D}} \rho_{\text{TF}} = 1$, we can derive analytical expressions for the corresponding approximation of the chemical potential $\mu \in \mathbb{R}$ [57]. From the same TF approximation, we can estimate the radius of the condensate as $R_{\text{TF}} = \sqrt{2\mu/(1 - C_\Omega)}$. Consequently, we set up the computational domain $\mathscr{D}$ as a disk of radius $R = 1.25R_{\text{TF}} = 6.56$. The initial guess u_0 is taken in the form of an off-center vortex placed at $(x_v = 0.25, y_v = 0)$; see Figure 6a. We use the ansatz $u_0 = \sqrt{\rho_{\text{TF}}}\, u_v$, where $u_v = r/\sqrt{r^2 + 2\xi^2}\, e^{i\theta}$, with (r, θ) representing the polar coordinates centered at (x_v, y_v) and $\xi = 1/\sqrt{2\mu}$ the nondimensional healing length, which is a good approximation of the vortex radius in rotating BEC [33]. A similar ansatz will also be used in subsequent sections to set up initial guesses with vortices for the calculation of more complicated BEC configurations. The stopping criterion (45) is used with the value $\varepsilon_{st} = 10^{-12}$. In these calculations the grid remains fixed (i.e., no grid adaptation is performed), with 9,578 vertices and 18,825 triangles. In the (BE) method the imaginary time step is chosen as $\delta t = 10$, which proved to be optimal for convergence after testing values of δt in the range from 0.01 to 100. In the (Ipopt) approach, two successive calls to the library were sufficient to converge the solution to the same level of accuracy as with other methods.

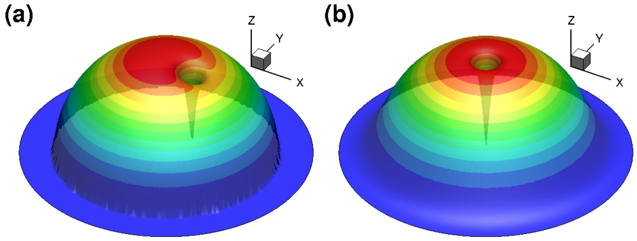

FIG. 6. *Computation of a rotating BEC with a single central vortex (cf. section* 8.1*).* 3*D-rendering of the atomic density* $\rho = |u|^2$ *for* (a) *the initial guess* u_0*, and* (b) *the converged ground state.*

For the considered physical parameters the ground state features a vortex centered at the origin. All considered methods, (PG), (RG), (RCG), (BE), and (Ipopt), converged to the same ground state shown in Figure 6b. In order to assess their respective rates of convergence, we compute a reference ("exact") solution u_{ex} using the same grid and starting the minimization algorithms from the initial guess u_0 with $(x_v = 0, y_v = 0)$. The corresponding energy and angular momentum are $E_{ex} = 8.36059$ and $L_{ex} = 1$.

The performance of the approaches corresponding to the different design choices discussed in section 6 is summarized in Table 3, whereas their convergence monitored in terms of the error norm $\|u_n - u_{ex}\|_2$ is illustrated in Figure 7. As compared with the results analyzed in section 7, the difference here is that we now allowed for periodic resets of the momentum term β_n to zero, and it was found by trial and error that the fastest convergence in terms of the CPU time was obtained when such resets were performed every 50 iterations. From Table 3 we conclude that the (RCG)-(PR) approach is more robust than the (RCG)-(FR) approach with respect to the choice of the vector transport and the reset frequency. We also note that ignoring the vector transport, which is the case in the (CG) methods, produces a significant increase of the computational time resulting from slow convergence of the arc-minimization procedure. As regards resetting the momentum term to zero, we note that it turns out to be particularly important for the (FR) approaches, concurring with the insights about them already known from the Euclidean setting [46]. In addition to being costly to determine, optimal momentum reset strategies also tend to be strongly problem-dependent. Thus, with this in mind, we conclude that the (RCG)-(PR)-(VtRS) approach again turns out to be the most efficient and is also characterized by the most consistent performance. Therefore, it will be used for further tests in the remainder of this section, and we will continue to refer to is as (RCG).

Table 3

Computation of a rotating BEC with a single central vortex (cf. section 8.1*). Performance of the gradient methods corresponding to the different design choices (cf. section* 6*) and of the (Ipopt) approach measured in terms of the number of iterations (iter) and the computational time in seconds (CPU) required for convergence.*

Method	Reset: none		Reset: 50 iter	
	iter	CPU	iter	CPU
(RCG)-(PR)-(VtRS)	196	913	190	903
(RCG)-(PR)-(VtDR)	186	954	180	868
(CG)-(PR)	198	1328	184	1271
(RCG)-(FR)-(VtRS)	242	1244	140	637
(RCG)-(FR)-(VtDR)	563	2652	359	1703
(CG)-(FR)	237	1583	150	1077
(RG)	2643	13872		
(PG)	3631	8472		
(BE)	2796	6838		
(Ipopt)	18	99		

The quantities $\|u_n - u_{ex}\|_2$, $|E_n - E_{ex}|$, and $|L_n - L_{ex}|$ shown in Figures 8a, 8b, and 8c as functions of n indicate that while the (RG) and (BE) methods converge with similar rates, the (RCG) approach converges much faster. The drift δ_n away from the constraint manifold $\mathcal{M}$ at intermediate steps $\hat{u}_n$ (cf. (47)) during the first 200 iterations is shown for different methods in Figure 8d. We note that for the (BE) method this quantity is always $\mathcal{O}(1)$, which is due to the fact that the RHS of (18) is based on an unprojected gradient further compounded by a large step size δt used. The normalization step is therefore crucial in this approach. On the other hand, δ_n is reduced faster in the (RCG) approach, where it also attains lower values than in the (PG) and (RG) methods.

Finally, let us note that, even though the (RCG) method outperforms all other first-order methods, the (Ipopt) approach actually converges much faster; cf. Table 3. Only 18 Hessian evaluations are needed during the two calls to `Ipopt,` and the total computational time is smaller by a factor of 10 as compared to the (RCG) methods. However, we stress that this test was performed with a fixed grid, and in fact this remarkable performance of the (Ipopt) method will be lost when computing more complicated cases that require mesh adaptivity (see the next section).

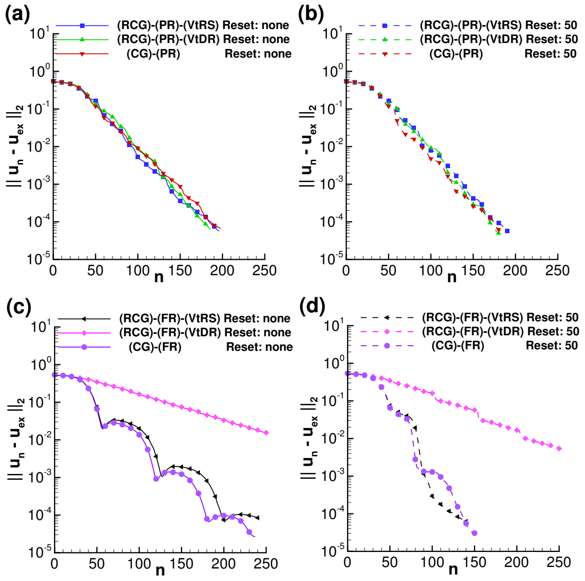

FIG. 7. *Computation of a rotating BEC with a single central vortex (cf. section* 8.1*). Comparison of different design choices for the (RCG) method in terms of convergence of the error norm* $\|u_n - u_{ex}\|_2$.

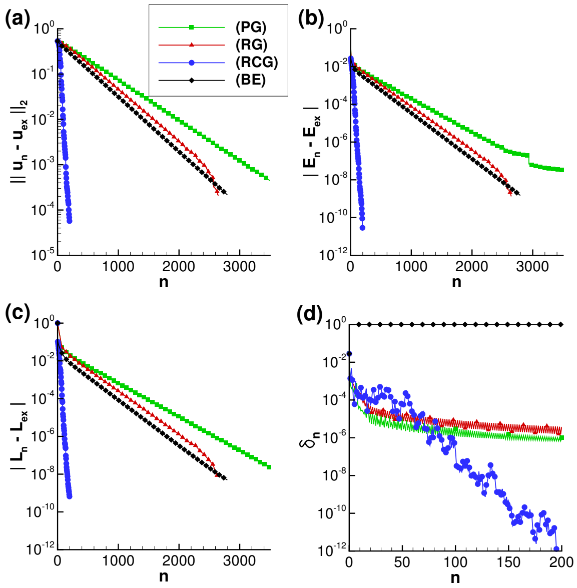

FIG. 8. *Computation of a rotating BEC with a single central vortex (cf. section* 8.1*). Convergence of the different quantities with iterations n:* (a) $\|(u_n - u_{ex})\|_2$, (b) $|E_n - E_{ex}|$, (c) $|L_n - L_{ex}|$, *and* (d) δ_n *(cf.* (47)*) is the drift away from the manifold.*

8.2. Test case 2: BEC with a dense Abrikosov vortex lattice. We now move on to consider more challenging test cases corresponding to a harmonic potential ($C_{\text{trap}} = r^2/2$), high rotation rate ($C_\Omega = 0.9$), and large values of the nonlinear interaction constant ($C_g = 1000$ to 15000). We note that for the harmonic trapping potential there is a physical limit occurring at the rotation frequency $C_\Omega = 1$ when the trapping is canceled by the centrifugal force (i.e., $C_{\text{trap}}^{\text{eff}} = 0$; see (12)). The next section will consider cases with a modified trapping potential, allowing for higher rotation frequencies.

We start with the case where $C_g = 1000$ and $C_\Omega = 0.9$ for which the ground state features over 50 vortices arranged in a regular triangular lattice called the Abrikosov lattice. The difficulty here is to obtain a very regular lattice, in particular for the vortices located near the border of the condensate where the atomic density is low. This explains why the results previously reported for this case exhibit configurations with somewhat different arrangements of the peripheral vortices, which nevertheless have very similar energy levels [61, 30, 11]. These differences can be attributed to the use of different initial guesses u_0. A nearly perfect arrangement of vortices on a triangular/hexagonal lattice is reported in the recent study [60] and will be considered here as a reference result used to validate our methods (the corresponding energy level is $E^{ref} = 6.3607$). Details of the computed stationary states depend on the initial guess u_0, and we used three distinct forms of u_0: (i) "ansatz d" proposed in [60] to model a central vortex using Gaussian functions, (ii) the TF approximation described in section 8.1 with one central vortex, and (iii) the TF approximation with six vortices. The corresponding stationary solutions obtained using the (RCG)-(PR)-(VtRS) method are shown in Figures 9a, 9b, and 9c. We can see that the central parts of the vortex lattices are in all cases essentially identical (modulo rotation), and some differences are detected among the peripheral vortices. The values of energy corresponding to these configurations differ by less than 0.01%.

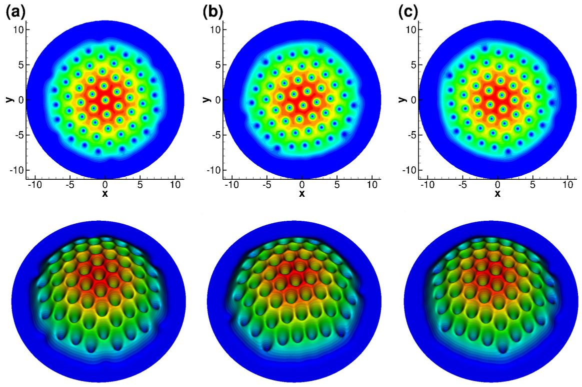

FIG. 9. *Computation of a rotating BEC with a dense Abrikosov vortex lattice (cf. section* 8.2*). Stationary states obtained using the (RCG)-(PR)-(VtRS) method with mesh adaptivity and different initial conditions* u_0*:* (a) *"ansatz d" suggested in* [60]*,* (b) *TF atomic density with one central vortex,* (c) *TF atomic density with a ring of six vortices. The figures in the first row show contours of the atomic density (normalized by its maximum value* ρ/ρ_{max}*) and in the second row they show the* 3*D-rendering of the same contours. The corresponding energies are* (a) $E = 6.3615$*,* (b) $E = 6.3621$*,* (c) $E = 6.3620$*, to be compared to the reference value* $E^{ref} = 6.3607$ *from* [60].

In the following we carry out computations starting from the initial guess (i), and mesh adaptation is now performed during iterations which are declared converged when the termination condition (45) with $\varepsilon_{st} = 10^{-9}$ is met. For this challenging test case the performance of a few selected design choices for the (RCG) and (CG) approaches is summarized in Table 4. Since the (FR) methods with and without

Table 4

Computation of a rotating BEC with a dense Abrikosov vortex using mesh adaptivity and initial condition u_0 described by the "ansatz d" suggested in [60] *(cf. section* 8.2*). Performance of the gradient methods corresponding to the indicated design choices (cf. section* 6*), and of the (Ipopt) approach measured in terms of the number of iterations (iter) and the computational time in seconds (CPU) required for convergence.*

Method	E	iter	CPU
(RCG)-(PR)-(VtRS)	6.3615	1339	23192
(RCG)-(PR)-(VtDR)	6.3615	2684	59431
(CG)-(PR)	6.3615	1557	25693
(Ipopt)	6.3621	354	22943

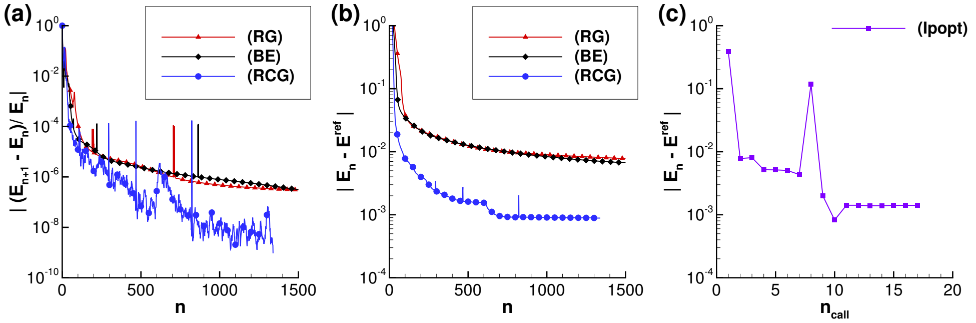

Fig. 10. *Computation of a rotating BEC with a dense Abrikosov vortex lattice (cf. section* 8.2*). Convergence of the different quantities with iterations n:* (a) $|(E_{n+1} - E_n)/E_n|$ *and* (b, c) $|E_n - E^{ref}|$*. All computations start from the initial guess suggested in* [60] *("ansatz d").*

momentum resets failed to converge to the same minimum as other approaches, we focus here on the (PR) techniques and note the fast convergence of the (RCG)-(PR)-(VtRS) approach, which will be used in further tests in this section; for brevity, we will refer to it simply as (RCG). It is interesting to note in Table 4 that the performance of the (Ipopt) method is significantly degraded as compared to the results from section 8.1 and is now comparable to that of the (RCG) method. The reason is that `Ipopt` is linked as an external library to `FreeFem++`, and therefore we cannot directly use mesh adaptivity in its internal algorithm. As a result, one has to use an external algorithm to couple the computation of the minimizer with the mesh adaptivity procedure employed in the other methods. The computations in the present case required 17 calls to `Ipopt`, with a total of 354 Hessian evaluations and a large number of internal iterations.

Convergence of the iterations carried out with the (RG), (RCG), and (BE) methods is compared in Figures 10a and 10b.

The gradient (RG) and the (BE) methods show a similar, but markedly slower, convergence (they were stopped after 5000 iterations) than the (RCG) method. We note that the peaks in the curves shown in Figures 10a and 10b result from reinterpolation of intermediate solutions after grid adaptation [29, 57]. In these figures we see that in all cases convergence slows down at later iterations, which is related to the slow rearrangement of vortices near the boundary of the condensate. Another possible reason is that since in [60] a different discretization was used (Fourier spectral approach with periodic boundary conditions), the value of E^{ref} taken from that reference might not exactly correspond to ours. Since convergence is monitored differently for the (Ipopt) method, in Figure 10c we show the decrease of the energy with the number of calls to the library (we add that the energy also tends to exhibit significant oscillations during iterations performed within each such call). While the energy level attained with the (Ipopt) method is a little higher than obtained with the (RCG) and (CG) methods, the corresponding minimizer is similar to that shown in Figure 9c.

8.3. Test case 3: BEC with a large number of vortices. In this section, we use the (RCG)-(PR)-(VtRS) method to compute fast rotating BEC ($C_\Omega = 0.9$) corresponding to large values of the nonlinear interaction constant with C_g varying from 1,000 to 15,000. For these difficult cases, a more physically relevant assessment of the convergence of iterations is provided by the alignment of vortices on parallel lines inside the vortex lattice. Since isocontours of atomic density do not always coincide with these lines, we developed a postprocessing approach to identify the centers of vortices by detecting local minima of the function $\rho_{\text{TF}} - \rho$. This postprocessing is similar to that used for experimental data [27] or 3D numerical simulations [28] and also allows us to build the Delaunay triangulation of the lattice and compute the radius of each vortex. The resulting stationary states are presented in this way in Figure 11. We notice an arrangement of vortices on a nearly perfect lattice for $C_g = 1,000$ and $5,000$, and a less regular arrangement for $C_g = 10{,}000$ and 15,000 with the presence of some defects in the lattice. This effect could be related to physical theories addressing the nonuniformity of the inter-vortex spacing in dense Abrikosov lattices [27, 53].

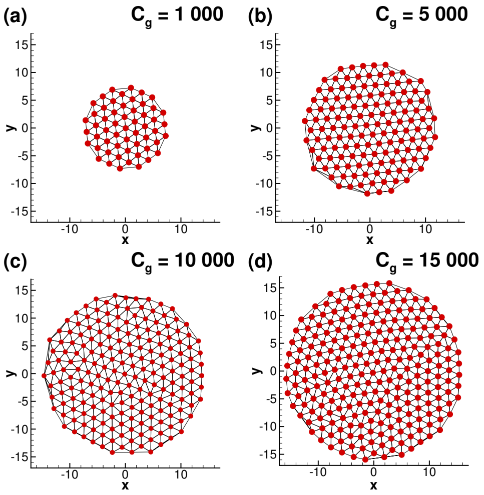

Fig. 11. *Computation of a fast rotating BEC ($C_\Omega = 0.9$) in a harmonic trapping potential (cf. section 8.2). The Abrikosov vortex lattice is represented using the Delaunay triangulation built from the detected vortex centers. Configurations obtained for large values of the nonlinear interaction constant: $C_g = 1000$ (55 vortices), $C_g = 5000$ (134 vortices), $C_g = 10000$ (193 vortices), and $C_g = 15000$ (237 vortices).*

8.4. Test case 4: BEC with giant vortex. To overcome the limit $C_\Omega = 1$ imposed by the harmonic trapping potential, a modified "harmonic-plus-Gaussian" potential was tested in experiments [20]. In [5, 28] this new experimental setup was modeled as

$$C_{\text{trap}}(x, y) = (1 - \alpha) r^2 + \frac{1}{4} k r^4, \tag{48}$$

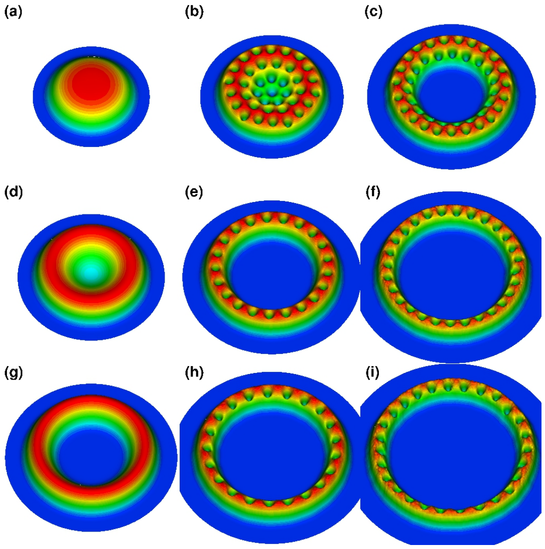

Fig. 12. *Computation of a rotating BEC with giant vortices (cf. section* 8.4*).* 3*D-rendering of the atomic density (normalized by its maximum value ρ/ρ_{max}) obtained in Regime* 1 (a, b, c)*, Regime* 2 (d, e, f)*, and Regime* 3 (g, h, i) *for different rotations $C_\Omega = 0$ (first column), $C_\Omega = 3$ (second column), and $C_\Omega = 4$ (third column).*

with the possibility to switch from a "quartic-plus-quadratic" potential ($\alpha < 1$), which corresponds to experiments, to a "quartic-minus-quadratic" potential ($\alpha > 1$), which is experimentally feasible but was never tested. Adapting the analysis from [5] to our 2D case, we obtain three possible regimes depending on the type of potential:

- Regime 1: "quartic-plus-quadratic" (or weak attractive) potential obtained for $\alpha < 1$ and $\mu > 0$ (see the TF approximation in section 8.1).
- Regime 2: weak "quartic-minus-quadratic" (or weak repulsive) potential obtained when $\alpha > 1$ and $\mu > 0$; this regime appears when $|1-\alpha| < \left(k\sqrt{3C_g/\pi}\right)/2$.
- Regime 3: strong "quartic-minus-quadratic" (or strong repulsive) potential obtained when $\alpha > 1$ and $\mu < 0$; this regime appears when $|1-\alpha| > \left(k\sqrt{3C_g/\pi}\right)/2$.

All computations are performed with the (RCG)-(PR)-(VrRS) method, and the obtained stationary states are presented in Figure 12. The parameters for these simulations are $C_g = 1000$, $k = 1$, and $\alpha = 1/2$ (Regime 1); $\alpha = 11/2$ (Regime 2); and $\alpha = 9$ (Regime 3). In the first column of Figure 12 we notice that the atomic density distribution in the condensate without rotation ($C_\Omega = 0$) changes from the classical parabolic profile in Regime 1 to a Mexican-hat type profile in Regime 2 and, finally, to a profile with a central hole in Regime 3. It is then expected that when rotation is applied, in Regimes 2 and 3 the condensate will develop a central hole (or giant vortex) at lower rotation frequencies than in Regime 1. This prediction is indeed supported by the results in Figure 12 (second and third columns). When rotation is increased, the condensate configuration evolves from a classical (Abrikosov) vortex lattice to a vortex lattice with a central depletion and, finally, to a giant vortex surrounded by a ring of individual vortices. The giant vortex is indeed obtained for lowest rotation frequencies in Regimes 2 and 3. The existence of a giant vortex was predicted theoretically (e.g., [33]) and was already observed in 2D [39] and 3D computations [5, 28].

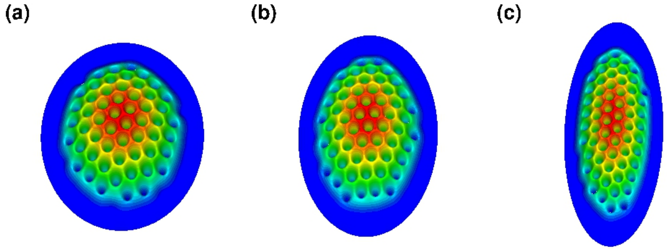

FIG. 13. *Computation of a rotating BEC in a strongly anisotropic trapping potential (cf. section* 8.5*). 3D-rendering of the atomic density (normalized by its maximum value ρ/ρ_{max}) obtained for* $\epsilon = 0.15$ (a)*,* $\epsilon = 0.35$ (b)*,* $\epsilon = 0.65$ (c)*.*

8.5. Test case 5: Strongly anisotropic BEC with vortices. To demonstrate the efficiency of the (RCG)-(PR)-(VrRS) method for the case without any symmetry, in this section we consider a rotating BEC trapped in a strongly anisotropic (asymmetrical) potential of the form suggested in [47],

$$C_{\text{trap}}(x,y) = \frac{1}{2}\left[(1+\eta^2)x^2 + (1-\eta)y^2\right], \quad \eta = 2(1-C_\Omega)\epsilon, \tag{49}$$

where $\epsilon < 1$ characterizes the anisotropy of the trap for very high rotation frequencies $C_\Omega \approx 1$. The theoretical analysis presented in [47] shows that when the condensate contains a large number of vortices, the deviation of the vortex lattice from a triangular arrangement is small. This finding is supported by our computational results shown in Figure 13 for three values of the anisotropy parameter ϵ. This example illustrates the flexibility of the finite-element discretization (cf. section 5) in handling highly deformed computational domains $\mathscr{D}$.

9. Conclusions. The difficulty of direct minimization of the GP energy functional with rotation comes from the unit-norm constraint (2). The novel idea proposed here is to transform this problem into an unconstrained Riemannian optimization problem defined on a spherical manifold and then develop a Riemannian conjugate gradient (RCG) method based on classical approaches. The key ingredients of this new method are the following: (i) the gradient direction is derived using the theory of Sobolev gradients and relies on a physically inspired definition of the inner product which accounts for rotation [30], thereby offering a good preconditioning for the problem; (ii) the gradient is projected on the subspace tangent to the spherical manifold before being used in simple gradient or conjugate gradient methods, which ensures the iterates stay close to manifold $\mathcal{M}$; (iii) the conjugate descent direction is computed using classical approaches (i.e., the Polak–Ribière or Fletcher–Reeves variant of the nonlinear conjugate gradient method), and the Riemannian vector transport is used to bring the gradient and descent directions determined at the previous iteration to the current tangent subspace $\mathcal{T}_{u_n}\mathcal{M}$; (iv) the optimal descent step is computed by solving an arc-minimization problem (in which samples are constrained to lie on the manifold), instead of the classical line-minimization; (v) finally, the updated solution is "retracted" back to the spherical manifold. In our study we carefully analyzed the effect of the key design choices, namely, the form of the momentum term and of the vector transport, on the performance of the Riemannian conjugate gradient approach. Based on tests involving several different problems, we conclude that the (RCG)-(PR)-(VrRS) approach, combining the Polak–Ribière form of the momentum term with the vector transport based on the Riemannian submanifold structure, exhibited the most robust and efficient performance. The (RCG)-(PR) methods in general also showed a systematic improvement over the (CG) approaches without vector transport.

As demonstrated by our tests performed in the finite-element setting, several features make the (RCG) method very appealing for practical computations: (i) since the "optimize-then-discretize" paradigm is used, the preconditioning is mesh-independent; (ii) the Riemannian retraction and transport operators are simple to implement; (iii) for the arc-minimization problem a classical approach such as Brent's method can be easily adapted; (iv) there are no tuning parameters or trust-region tests involved. In addition, general mesh refinement or mesh adaptivity strategies are compatible with the (RCG) method without any modifications. Our extensive numerical experiments showed a significant improvement of the convergence rate of the (RCG) method over the simple gradient and imaginary-time methods. For more involved problems requiring mesh adaptation the (RCG) approach exhibited performance comparable to the (Ipopt) method, which for equality-constrained problems implements a Newton-type technique. The reason is that there is no straightforward way to incorporate mesh adaptation in the (Ipopt) approach—something that can be done rather easily in the (RCG) method. We stress that the use of mesh adaptation is essential for efficient computational solution of problems of the type discussed in sections 8.2–8.5, and, to the best of our knowledge, implementation of mesh adaptation in Ipopt-type or, more generally, Newton-type methods, remains an open problem. We also emphasize that the (RCG) approach has far fewer parameters than the (Ipopt) method, which greatly simplifies its performance optimization. Finally, as a challenging test, the (RCG) method was used to compute vortex configurations in rotating BEC with high values of the nonlinear interaction constants and very high rotation rates, as well as in configurations with strongly anisotropic trapping potentials.

Lastly, we reiterate that the approach presented in this study does not exploit all opportunities inherent in the Riemannian formulation. In particular, it remains an open question whether the use of a well-adapted Riemannian metric defined on the constraint manifold could further improve the performance of the approach. In addition, one can also consider the Riemannian formulations of Newton's method and of different variants of the quasi-Newton method. Work is already ongoing for some of these problems, and results will be reported in the near future.

Acknowledgments. The authors are grateful to the anonymous referees for providing constructive feedback on this manuscript. The authors acknowledge the generous hospitality of the Fields Institute in Toronto during the Thematic Program on Multiscale Scientific Computing (January–April, 2016).

REFERENCES

[1] P.-A. Absil, R. Mahony, and R. Sepulchre, *Optimization Algorithms on Matrix Manifolds*, Princeton University Press, 2008.

[2] R. A. Adams and J. F. Fournier, *Sobolev Spaces*, Elsevier, 2005.

[3] A. Aftalion, *Vortices in Bose-Einstein Condensates*, Birkhäuser, 2006.

[4] A. Aftalion and I. Danaila, *Three-dimensional vortex configurations in a rotating Bose-Einstein condensate*, Phys. Rev. A, 68 (2003), 023603.

[5] A. Aftalion and I. Danaila, *Giant vortices in combined harmonic and quartic traps*, Phys. Rev. A, 69 (2004), 033608.

[6] A. Aftalion and Q. Du, *Vortices in a rotating Bose-Einstein condensate: Critical angular velocities and energy diagrams in the Thomas-Fermi regime*, Phys. Rev. A, 64 (2001), 063603.

[7] F. Alouges, *A new algorithm for computing liquid crystal stable configurations: The harmonic mapping case*, SIAM J. Numer. Anal., 34 (1997), pp. 1708–1726, https://doi.org/10.1137/S0036142994264249.

[8] F. Alouges and C. Audouze, *Gradient flows, and application to the Hartree-Fock functional*, Numer. Methods Partial Differential Equations, 25 (2009), pp. 380–400.

[9] X. Antoine, C. Besse, and W. Bao, *Computational methods for the dynamics of the nonlinear Schrödinger/Gross-Pitaevskii equations*, Comput. Phys. Comm., 184 (2013), pp. 2621–2633.

[10] X. Antoine, C. Besse, R. Duboscq, and V. Rispoli, *Acceleration of the imaginary time method for spectrally computing the stationary states of Gross-Pitaevskii equations*, Comput. Phys. Comm., 219 (2017), pp. 70–78.

[11] X. Antoine and R. Duboscq, *Robust and efficient preconditioned Krylov spectral solvers for computing the ground states of fast rotating and strongly interacting Bose-Einstein condensates*, J. Comput. Phys., 258 (2014), pp. 509–523.

[12] X. Antoine, A. Levitt, and Q. Tang, *Efficient spectral computation of the stationary states of rotating Bose-Einstein condensates by the preconditioned nonlinear conjugate gradient method*, J. Comput. Phys., 343 (2017), pp. 92–109.

[13] W. Bao, *Ground states and dynamics of rotating Bose-Einstein condensates*, in Modeling and Simulation in Science, Engineering and Technology, Birkhäuser, 2006, pp. 215–255.

[14] W. Bao, *Mathematical models and numerical methods for Bose-Einstein condensation*, in Proc. of the International Congress of Mathematicians (Seoul 2014), vol. IV, 2014, pp. 971–996.

[15] W. Bao and Y. Cai, *Mathematical theory and numerical methods for Bose-Einstein condensation*, Kinetic Related Models, 6 (2013), pp. 1–135.

[16] W. Bao and Q. Du, *Computing the ground state solution of Bose–Einstein condensates by a normalized gradient flow*, SIAM J. Sci. Comput., 25 (2004), pp. 1674–1697, https://doi.org/10.1137/S1064827503422956.

[17] W. Bao and J. Shen, *A generalized Laguerre-Hermite pseudospectral method for computing symmetric and central vortex states in Bose-Einstein condensates*, J. Comput. Phys., 227 (2008), pp. 9778–9793.

[18] W. Bao and W. Tang, *Ground-state solution of Bose–Einstein condensate by directly minimizing the energy functional*, J. Comput. Phys., 187 (2003), pp. 230–254.

[19] N. Boumal, B. Mishra, P.-A. Absil, and R. Sepulchre, *Manopt, a MATLAB toolbox for optimization on manifolds*, J. Mach. Learn. Res., 15 (2014), pp. 1455–1459.

[20] V. Bretin, S. Stock, Y. Seurin, and J. Dalibard, *Fast rotation of a Bose-Einstein condensate*, Phys. Rev. Lett., 92 (2004), 050403.

[21] M. Caliari and S. Rainer, *GSGPEs: A MATLAB code for computing the ground state of systems of Gross-Pitaevskii equations*, Comput. Phys. Comm., 184 (2013), pp. 812–823.

[22] R. Caplan, *NLSEmagic: Nonlinear Schrödinger equation multi-dimensional MATLAB-based GPU-accelerated integrators using compact high-order schemes*, Comput. Phys. Comm., 184 (2013), pp. 1250–1271.

[23] M. Castro-Diaz, F. Hecht, and B. Mohammadi, *Anisotropic grid adaptation for inviscid and viscous flows simulations*, Int. J. Comput. Fluid Dyn., 25 (2000), pp. 475–491.

[24] S.-L. Chang, C.-S. Chien, and B.-W. Jeng, *Computing wave functions of nonlinear Schrödinger equations: A time-independent approach*, J. Comput. Phys., 226 (2007), pp. 104–130.

[25] E. Charalampidis, P. Kevrekidis, and P. Farrell, *Computing stationary solutions of the two-dimensional Gross-Pitaevskii equation with deflated continuation*, Comm. Nonlinear Sci. Numer. Simul., 54 (2018), pp. 482–499.

[26] P. G. Ciarlet, *The Finite Element Method for Elliptic Problems*, Stud. Math. Appl., North-Holland, 1978. Reprinted as Classics Appl. Math. 40, SIAM, 2002, https://doi.org/10.1137/1.9780898719208.

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

[28] I. Danaila, *Three-dimensional vortex structure of a fast rotating Bose-Einstein condensate with harmonic-plus-quartic confinement*, Phys. Rev. A, 72 (2005), 013605.

[29] I. Danaila and F. Hecht, *A finite element method with mesh adaptivity for computing vortex states in fast-rotating Bose-Einstein condensates*, J. Comput. Phys., 229 (2010), pp. 6946–6960.

[30] I. Danaila and P. Kazemi, *A new Sobolev gradient method for direct minimization of the Gross–Pitaevskii energy with rotation*, SIAM J. Sci. Comput., 32 (2010), pp. 2447–2467, https://doi.org/10.1137/100782115.

[31] C. M. Dion and E. Cancès, *Ground state of the time-independent Gross-Pitaevskii equation*, Comput. Phys. Comm., 177 (2007), pp. 787–798.

[32] I. Faragò and J. Karàtson, *Numerical Solution of Nonlinear Elliptic Problems via Preconditioning Operators: Theory and Applications*, NOVA Science Publishers, 2002.

[33] A. L. Fetter, *Rotating vortex lattice in a Bose-Einstein condensate trapped in combined quadratic and quartic radial potentials*, Phys. Rev. A, 64 (2001), 063608.

[34] J. J. García-Ripoll and V. M. Pérez-García, *Optimizing Schrödinger functionals using Sobolev gradients: Application to quantum mechanics and nonlinear optics*, SIAM J. Sci. Comput., 23 (2001), pp. 1316–1334, https://doi.org/10.1137/S1064827500377721.

[35] M. D. Gunzburger, *Perspectives in Flow Control and Optimization*, Adv. Des. Control 5, SIAM, 2003, https://doi.org/10.1137/1.9780898718720.

[36] F. Hecht, *New developments in Freefem++*, J. Numer. Math., 20 (2012), pp. 251–266.

[37] F. Hecht, O. Pironneau, A. L. Hyaric, and K. Ohtsuke, *FreeFem++ Manual*, www.freefem.org, 2007.

[38] U. Hohenester, *OCTBEC: A MATLAB toolbox for optimal quantum control of Bose-Einstein condensates*, Comput. Phys. Comm., 185 (2014), pp. 194–216.

[39] K. Kasamatsu, M. Tsubota, and M. Ueda, *Giant hole and circular superflow in a fast rotating Bose-Einstein condensate*, Phys. Rev. A, 66 (2002), 053606.

[40] P. Kazemi and M. Eckart, *Minimizing the Gross-Pitaevskii energy functional with the Sobolev gradient—analytical and numerical results*, Int. J. Comput. Methods, 7 (2009), pp. 453–476.

[41] E. H. Lieb and R. Seiringer, *Derivation of the Gross-Pitaevskii equation for rotating Bose gases*, Comm. Math. Phys., 264 (2006), pp. 505–537.

[42] D. Luenberger, *Optimization by Vector Space Methods*, John Wiley and Sons, 1969.

[43] B. Merlet and T. N. Nguyen, *Convergence to equilibrium for discretizations of gradient-like flows on Riemannian manifolds*, Differential Integral Equations, 26 (2013), pp. 571–602.

[44] J. W. Neuberger, *Sobolev Gradients and Differential Equations*, Lecture Notes in Math. 1670, Springer, 1997. 2nd edition published 2010.

[45] A. W. J. Nocedal, A. Wächter, and R. A. Waltz, *Adaptive barrier update strategies for nonlinear interior methods*, SIAM J. Optim., 19 (2008), pp. 1674–1693, https://doi.org/10.1137/060649513.

[46] J. Nocedal and S. Wright, *Numerical Optimization*, Springer, 2002.

[47] M. Ö. Oktel, *Vortex lattice of a Bose-Einstein condensate in a rotating anisotropic trap*, Phys. Rev. A, 69 (2004), 023618.

[48] M. Pierre, *Newton and conjugate gradient for harmonic maps from the disc into the sphere*, ESAIM Control Optim. Calc. Var., 10 (2004), pp. 142–167.

[49] L. P. Pitaevskii and S. Stringari, *Bose-Einstein Condensation*, Clarendon Press, 2003.

[50] W. H. Press, B. P. Flanner, S. A. Teukolsky, and W. T. Vetterling, *Numerical Recipes: The Art of Scientific Computations*, Cambridge University Press, 1986.

[51] W. Ring and B. Wirth, *Optimization methods on Riemannian manifolds and their application to shape space*, SIAM J. Optim., 22 (2012), pp. 596–627, https://doi.org/10.1137/11082885X.

[52] P. J. Roache, *Verification and Validation in Computational Science and Engineering*, Hermosa Publishers, 1998.

[53] D. E. Sheehy and L. Radzihovsky, *Vortices in spatially inhomogeneous superfluids*, Phys. Rev. A, 70 (2004), 063620.

[54] S. T. Smith, *Optimization techniques on Riemannian manifolds*, Fields Inst. Comm., 3 (1994).

[55] S. Stringari, *Phase diagram of quantized vortices in a trapped Bose-Einstein condensed gas*, Phys. Rev. Lett., 82 (1999), pp. 4371–4375.

[56] M. Tsubota, *Quantized vortices in superfluid helium and Bose-Einstein condensates*, J. Phys. Conf. Ser., 31 (2006), pp. 88–94.

[57] G. Vergez, I. Danaila, S. Auliac, and F. Hecht, *A finite-element toolbox for the stationary Gross-Pitaevskii equation with rotation*, Comput. Phys. Comm., 209 (2016), pp. 144–162.

[58] A. Wächter, *An Interior Point Algorithm for Large-Scale Nonlinear Optimization with Applications in Process Engineering*, Ph.D. thesis, Carnegie Mellon University, 2002.

[59] A. Wächter and L. T. Biegler, *On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming*, Math. Programming, 106 (2006), pp. 25–57.

[60] X. Wu, Z. Wen, and W. Bao, *A regularized Newton method for computing ground states of Bose-Einstein condensates*, J. Sci. Comput., 73 (2017), pp. 303–329.

[61] R. Zeng and Y. Zhang, *Efficiently computing vortex lattices in rapid rotating Bose-Einstein condensates*, Comput. Phys. Comm., 180 (2009), pp. 854–860.