



CHALMERS
UNIVERSITY OF TECHNOLOGY

SuperConga: An open-source framework for mesoscopic superconductivity

Downloaded from: <https://research.chalmers.se>, 2024-04-28 17:50 UTC

Citation for the original published paper (version of record):

Holmvall, P., Wennerdal, N., Håkansson, M. et al (2023). SuperConga: An open-source framework for mesoscopic superconductivity. *Applied Physics Reviews*, 10(1).
<http://dx.doi.org/10.1063/5.0100324>

N.B. When citing this work, cite the original published paper.

SuperConga: An open-source framework for mesoscopic superconductivity

Cite as: Appl. Phys. Rev. **10**, 011317 (2023); doi: [10.1063/5.0100324](https://doi.org/10.1063/5.0100324)

Submitted: 23 May 2022 · Accepted: 13 February 2023 ·

Published Online: 22 March 2023



P. Holmvall,^{1,2}  N. Wall Wennerdal,¹  M. Håkansson,¹  P. Stadler,¹  O. Shevtsov,¹  T. Löfwander,¹ 
and M. Fogelström^{1,a)} 

AFFILIATIONS

¹Department of Microtechnology and Nanoscience—MC2, Chalmers University of Technology, SE-412 96 Göteborg, Sweden

²Department of Physics and Astronomy, Uppsala University, P. O. Box 516, S-751 20 Uppsala, Sweden

^{a)}Author to whom correspondence should be addressed: mikael.fogelstrom@chalmers.se

ABSTRACT

We present SuperConga, an open-source framework for simulating equilibrium properties of unconventional and ballistic singlet superconductors, confined to two-dimensional (2D) mesoscopic grains in a perpendicular external magnetic field, at arbitrary low temperatures. It aims at being both fast and easy to use, enabling research without access to a computer cluster, and visualization in real-time with OpenGL. The core is written in C++ and CUDA, exploiting the embarrassingly parallel nature of the quasiclassical theory of superconductivity by utilizing the parallel computational power of modern graphics processing units. The framework self-consistently computes both the superconducting order-parameter and the induced vector potential and finds the current density, free energy, induced flux density, local density of states (LDOS), and the magnetic moment. A user-friendly Python frontend is provided, enabling simulation parameters to be defined via intuitive configuration files, or via the command-line interface, without requiring a deep understanding of implementation details. For example, complicated geometries can be created with relative ease. The framework ships with simple tools for analyzing and visualizing the results, including an interactive plotter for spectroscopy. An overview of the theory is presented, as well as examples showcasing the framework's capabilities and ease of use. The framework is free to download from <https://gitlab.com/superconga/superconga>, which also links to the extensive user manual, containing even more examples, tutorials, and guides. To demonstrate and benchmark SuperConga, we study the magnetostatics, thermodynamics, and spectroscopy of various phenomena. In particular, we study flux quantization in solenoids, vortex physics, surface Andreev bound-states, and a “phase crystal.” We compare our numeric results with analytics and present experimental observables, e.g., the magnetic moment and LDOS, measurable with, for example, scanning probes, STM, and magnetometry.

© 2023 Author(s). All article content, except where otherwise noted, is licensed under a Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>). <https://doi.org/10.1063/5.0100324>

TABLE OF CONTENTS

I. INTRODUCTION	1	2. Solving London–Maxwell equation in a symmetric gauge	15
II. QUASICLASSICAL THEORY OF SUPERCONDUCTIVITY	2	B. A mesoscopic <i>s</i> -wave superconducting disk	18
A. Riccati formalism for inhomogeneous states	4	C. Spectroscopy on superconducting nanostructures	20
III. OVERVIEW OF THE IMPLEMENTATION	5	VI. SUMMARY	21
A. The main algorithm of SuperConga	5		
B. Framework overview	6		
1. Backend	7		
2. Frontend	8		
IV. DEMONSTRATION OF SUPERCONGA	9		
V. EXAMPLES AND RESULTS	12		
A. Superconducting annulus in a magnetic field	12		
1. The solenoid gauge	14		

I. INTRODUCTION

Over the last few decades, there has been a tremendous improvement in the fabrication and controllable downscaling of samples and devices. The physical size of a sample can now be comparable to characteristic quantum length scales of the composing materials. For a superconducting material, which is the topic of this article, the relevant length scales are the superconducting coherence length, the length

scale over which the superconducting order parameter typically varies, and the magnetic penetration depth, which determines the length scale of screening. In direct connection with the ability to make smaller samples and devices, new measurement and sensing tools have been developed. Current scanning probes can spatially resolve minute variations in magnetic fields using scanning SQUIDS^{1–5} or make detailed spatial maps of low-lying quasiparticle states and coherences using scanning tunneling spectroscopy (STS).^{6–15} By fabricating nano-scale cantilevers, the intrinsic state of a superconducting island can be monitored as changes in the cantilever-oscillation frequency.^{16–18} Other techniques of detecting and following the superconducting state in nanoscale devices include Hall magnetometry,^{19,20} scanning electron microscopy (SEM),²¹ and scanning Hall probe microscopy.^{22–24} Scanning single electron transistors (SET) are local probes that can map out charging, or parity, effects that become prominent for ultra-small superconductors where the energy-level spacing and the superconducting pairing are of the same order of magnitude.^{25–29} Experimental data gathered using the above-mentioned methods give rich and complex information about the strongly correlated material system under study. To fully understand these data, one needs to compare and contrast them to theoretical predictions and modeling.

Superconductivity in metals is explained by the Bardeen–Cooper–Schrieffer (BCS) theory.^{30–33} Its extension and generalization to unconventional superconductivity and superfluids^{34,35} makes the theory a corner stone of condensed matter physics. One method to model superconducting devices and to explore fundamental problems of superconductivity with the BCS theory as a starting point is the quasiclassical theory of superconductivity. It is the extension of Landau’s Fermi-liquid theory to include superconducting phenomena and was pioneered by Eilenberger,³⁶ Larkin and Ovchinnikov,³⁷ and Eliashberg.³⁸ The quasiclassical approximation relies on a separation between relevant energy, temperature, and length scales in the normal-metal and superconducting states. The normal-metal state is characterized by the Fermi energy, E_F , Fermi temperature, T_F , and inverse Fermi wave number, $1/k_F$, while the superconducting state is characterized by the superconducting order parameter, Δ , transition temperature, T_c , and coherence length, ξ_0 . Quasiclassical theory is then a controlled expansion in small parameters like Δ/E_F , T_c/T_F , or $1/(\xi_0 k_F)$, which are usually of order 10^{-2} – 10^{-3} in metals. As a Green’s function based theory,³⁹ it is very powerful and include in its most general form material effects such as impurity scattering, Fermi-liquid effects, electron-phonon interaction, and non-equilibrium situations for instance imposed by external fields or potentials.^{40,41} At the same time, it is versatile enough that it can be adapted to realistic device sizes and geometries.^{42–45} In its simplest form, quasiclassical theory is equivalent to the Andreev approximation of the Bogoliubov–deGennes equation for the ballistic case.^{32,46,47}

When the mean free path, ℓ , due to elastic impurity scattering is much smaller than the superconducting coherence length, ξ_0 , it is possible to derive diffusion equations from the more general Eilenberger–Larkin–Ovchinnikov equations. These diffusion equations, first derived by Usadel,^{48,49} are more easy to handle and is widely used to describe conventional diffusive *s*-wave samples and devices.^{50–52} Solution methods in the diffusive case include Nazarov’s circuit theory⁵³ and finite-element methods.⁵⁴ Here, we focus instead on the more general equations in the ballistic regime.

A complication when solving the equations of quasiclassical theory is that one often needs to resort to numerical methods. Developing

the necessary codes is technically demanding and time-consuming. Today, there is no open-source code general enough to describe ballistic devices or superconducting grains of mesoscopic size with the quasiclassical theory. This lack of an open-source code forces all researchers and students in the community to re-implement the theory for each individual problem.

To remedy this, in this paper, we present an open source framework for studies of two-dimensional superconducting grains. The application program interface (API) is written in C++ and CUDA and takes advantage of the speedup made possible by running the code on Graphics Processing Units (GPUs). At the same time, the Python-based frontend is sufficiently easy to use that any user interested only in the physics and results never has to dwell on the technical details of the implementation. The first version of the API that we present here can be used to study conventional and unconventional singlet superconducting grains of general geometry in two dimensions with an applied external magnetic field. The framework is sufficiently modular that in the future it can be extended to include more aspects of quasiclassical theory.

This paper is organized in the following: In Sec. II, we give an introduction to quasiclassical theory with the aim to provide a self-contained account of the theory, which the framework is based on. In this section, we also show some simple model calculations, serving as a background to the examples worked out in later sections. In Sec. III, we give an overview of the main algorithm of SuperConga and give sufficient background to understand the parameters that have to be set while running the simulations. In Sec. IV, we show by a simple example how to run the framework using the Python frontend. Section V contains a few more involved examples that benchmark and highlight the capabilities of SuperConga. We study the rather complicated physics and energetics of vortices in a superconducting grain. In particular, we find B_{c1} where it becomes energetically favorable to have one vortex in a finite size disk. By comparison to formulas from literature, we are also able to extract the vortex-core energy. At high fields, we demonstrate that many different vortex configurations can be stabilized with almost the same free energy. For lower fields, such configurations might be more separated in energy but can be studied as metastable states. In another example, we find the vortex configurations and the field-dependent local density of states (LDOS) in a grain studied experimentally by Timmermans *et al.*⁵⁵ Finally, we show that in a *d*-wave superconducting annulus, the free-energy parabolas are conventional³³ at high temperatures, while at low temperatures zero-energy Andreev bound states at the edges modifies the energetics through their paramagnetic response to the external field. Interestingly, spontaneous circular surface currents^{56–58} can coexist at high external magnetic fields with superconducting phase windings $n2\pi$ (n integer) around the annulus and integer flux $n\Phi_0$ threading the inner circular hole. Section VI contains a summary of the paper. Appendixes A–F contain additional information on the SuperConga external dependencies and parameter configuration file, details about self-consistency convergence acceleration, the technique to solve for the vector potential and the equations of motion, as well as tables summarizing our choice of units and order-parameter basis functions.

II. QUASICLASSICAL THEORY OF SUPERCONDUCTIVITY

The superconducting state is characterized by an order parameter,

$$\Delta(\mathbf{p}_F, \mathbf{R}) = \sum_{\Gamma} |\Delta_{\Gamma}(\mathbf{R})| e^{i\chi_{\Gamma}(\mathbf{R})} \eta_{\Gamma}(\mathbf{p}_F), \quad (1)$$

that is zero for temperatures above the superconducting transition ($T > T_c$) and non-zero below it. All superconductors break $U(1)$ -gauge symmetry, and hence, the minimal order parameter is described by an amplitude $|\Delta(\mathbf{R})|$ and a phase $\chi(\mathbf{R})$, both, in general, dependent on the spatial coordinates \mathbf{R} . The phase is directly coupled to the superfluid momentum, $\mathbf{p}_s(\mathbf{R})$, and the electromagnetic gauge field, $\mathbf{A}(\mathbf{R})$, via the gauge invariant expression,

$$\mathbf{p}_s(\mathbf{R}) = \frac{\hbar}{2} \nabla \chi(\mathbf{R}) - \frac{e}{c} \mathbf{A}(\mathbf{R}), \quad (2)$$

where $e = -|e|$ is the charge of the electron, \hbar is Planck's constant, and c is the speed of light. In addition to breaking the $U(1)$ symmetry, some superconducting compounds also break symmetries of the crystal lattice. This possibility is encoded in the complex-valued basis function $\eta_{\Gamma}(\mathbf{p}_F)$, which depends on the Fermi momentum, \mathbf{p}_F . The index Γ denotes the irreducible representation of the crystallographic point group that the basis function belongs to.⁵⁹ In the general case, we also need to account for the spin-degree of freedom, and the order-parameter field should, therefore, be written as a spin-matrix $\Delta_{\alpha\beta}(\mathbf{p}_F, \mathbf{R})$. However, we will focus on spin-singlet superconductivity, and the form in Eq. (1) is adequate for this.

The singlet order parameter satisfies the following gap equation:

$$\Delta(\mathbf{p}_F, \mathbf{R}) = N_F T \sum_n^{|e_n| < e_c} \langle V(\mathbf{p}_F, \mathbf{p}'_F) f(\mathbf{p}'_F, \mathbf{R}; \varepsilon_n) \rangle_{\mathbf{p}'_F}, \quad (3)$$

where $V(\mathbf{p}_F, \mathbf{p}'_F)$ is the effective superconducting pairing interaction. We use the approximation that the interaction can be separated into symmetry channels of the crystal point group,

$$V(\mathbf{p}_F, \mathbf{p}'_F) = \sum_{\Gamma} V_{\Gamma} \eta_{\Gamma}(\mathbf{p}_F) \eta_{\Gamma}^{\dagger}(\mathbf{p}'_F), \quad (4)$$

where V_{Γ} is the pairing strength in the symmetry channel Γ . In this case, the dependencies on position and momentum direction of the order parameter separate as in Eq. (1). The angle bracket $\langle \cdots \rangle_{\mathbf{p}_F}$ denotes a Fermi-surface average, which in 2D is a line integral around the Fermi surface according to^{60,61}

$$\langle \cdots \rangle_{\mathbf{p}_F} = \frac{1}{N_F} \oint_{\text{FS}} \frac{d\mathbf{p}_F}{(2\pi\hbar)^2 |v_F(\mathbf{p}_F)|} (\cdots), \quad (5)$$

where the total normal-state density of states per spin at the Fermi level is

$$N_F = \oint_{\text{FS}} \frac{d\mathbf{p}_F}{(2\pi\hbar)^2 |v_F(\mathbf{p}_F)|}. \quad (6)$$

We consider the extreme layered superconductor with conduction in stacked two-dimensional planes. In this case, $N_F = k_F / 2\pi\hbar v_F d$, where k_F , v_F are the in-plane Fermi-wave number and Fermi velocity, respectively, and d is the inter plane distance.⁶⁰ The vector potential is determined through Ampère's law,

$$\nabla \times \nabla \times \mathbf{A}(\mathbf{R}) = \frac{4\pi}{c} \mathbf{j}(\mathbf{R}), \quad (7)$$

where the charge current density is defined as

$$\mathbf{j}(\mathbf{R}) = 2eN_F T \sum_n \langle v_F(\mathbf{p}_F) g(\mathbf{p}_F, \mathbf{R}; \varepsilon_n) \rangle_{\mathbf{p}_F}, \quad (8)$$

where the factor 2 accounts for spin degeneracy. The Fermi velocity at \mathbf{p}_F on the Fermi surface is $v_F(\mathbf{p}_F) = \nabla_{\mathbf{p}} \varepsilon(\mathbf{p})|_{\mathbf{p}=\mathbf{p}_F}$, where $\varepsilon(\mathbf{p})$ is the dispersion.

The two functions $g(\mathbf{p}_F, \mathbf{R}; \varepsilon_n)$ and $f(\mathbf{p}_F, \mathbf{R}; \varepsilon_n)$ appearing in Eqs. (3) and (8) are components of the quasiclassical Green's function, or propagator,

$$\hat{g}(\mathbf{p}_F, \mathbf{R}; \varepsilon) = \begin{pmatrix} g(\mathbf{p}_F, \mathbf{R}; \varepsilon) & f(\mathbf{p}_F, \mathbf{R}; \varepsilon) \\ -\tilde{f}(\mathbf{p}_F, \mathbf{R}; \varepsilon) & \tilde{g}(\mathbf{p}_F, \mathbf{R}; \varepsilon) \end{pmatrix}. \quad (9)$$

It is a 2×2 matrix in electron-hole (Nambu) space, as indicated by the “ \sim ”-symbol. The propagator is determined from the quasiclassical counter part of the Gorkov–Dyson equation, the transport-like Eilenberger equation,

$$0 = i\mathbf{v}_F \cdot \nabla_{\mathbf{R}} \hat{g}(\mathbf{p}_F, \mathbf{R}; \varepsilon) + \left[\left(\varepsilon + \frac{e}{c} \mathbf{v}_F(\mathbf{p}_F) \cdot \mathbf{A}(\mathbf{R}) \right) \hat{\tau}_3 - \hat{\Delta}(\mathbf{p}_F, \mathbf{R}), \hat{g}(\mathbf{p}_F, \mathbf{R}; \varepsilon) \right], \quad (10)$$

where $\hat{\tau}_3$ is the third Pauli matrix in Nambu space. In addition to Eq. (10), the propagator obeys the normalization condition,

$$\hat{g}^2(\mathbf{p}_F, \mathbf{R}; \varepsilon) = -\pi^2 \hat{\mathbb{I}}, \quad (11)$$

where $\hat{\mathbb{I}}$ is the identity matrix.

In our case of weak-coupling superconductivity in the clean limit, the self-energy $\hat{\Delta}(\mathbf{p}_F, \mathbf{R})$ entering in Eq. (10) simply consists of the order parameter,

$$\hat{\Delta}(\mathbf{p}_F, \mathbf{R}) = \begin{pmatrix} 0 & \Delta(\mathbf{p}_F, \mathbf{R}) \\ -\tilde{\Delta}(\mathbf{p}_F, \mathbf{R}) & 0 \end{pmatrix}. \quad (12)$$

There is some redundancy in the parametrization of the propagator, and the following symmetry^{35,62}

$$\tilde{x}(\mathbf{p}_F, \mathbf{R}; \varepsilon) = x(-\mathbf{p}_F, \mathbf{R}; -\varepsilon^*), \quad (13)$$

between tilded (\tilde{x}) and un-tilded (x) quantities holds. The propagator may be evaluated at imaginary frequencies $\varepsilon \rightarrow i\varepsilon_n = i\pi T(2n+1)$ giving the Matsubara propagator or at real frequencies giving either the Retarded ($\varepsilon \rightarrow \varepsilon + i\delta$), or the Advanced propagator ($\varepsilon \rightarrow \varepsilon - i\delta$). Here, δ is a small and positive energy broadening. To compute the order parameter, the current, and the back-coupling through the gauge field, it is enough to work with the Matsubara propagator. For the local density of states, defined as

$$N(\mathbf{p}_F, \mathbf{R}; \varepsilon) = -\frac{N_F}{\pi} \text{Im } g^R(\mathbf{p}_F, \mathbf{R}; \varepsilon), \quad (14)$$

we need to also compute the retarded Green's function with the order parameter and gauge field as input.

The superconducting state has a characteristic energy scale given by the transition temperature T_c , namely, $2\pi k_B T_c$, and a natural length scale, the coherence length $\xi_0 = \hbar v_F / 2\pi k_B T_c$. We will use the natural units $\hbar = k_B = 1$. The normal state density of states at the Fermi level N_F is in units $[\text{Energy} \times \text{unit cell} \times \text{spin}]^{-1}$. In addition to the coherence length, the length scale for screening of electromagnetic fields enters the theory through the penetration depth λ_0 , defined as

$$\frac{1}{\lambda_0^2} = \frac{4\pi e^2}{c^2} v_F^2 N_F. \quad (15)$$

We use the Ginzburg–Landau parameter κ , defined as $\kappa = \lambda_0/\xi_0$, to give the value of the penetration depth in relation to the coherence length. To define the magnetic units we start from the superconducting magnetic flux quantum $\Phi_0 = hc/2|e|$. The magnetic field,

$$\mathbf{B} = \nabla \times \mathbf{A}, \quad (16)$$

is then given in units of $B_0 = \Phi_0/\pi\xi_0^2$ and the vector potential in units of $A_0 = \Phi_0/\pi\xi_0$. Finally, current densities will be given in units of $j_0 = 2\pi k_B T_c |e| v_F N_F$. A summary of these units and natural scales are given in Table II in Appendix F.

The quasiclassical theory is a conserving theory in the sense that the equations for \hat{g} , $\hat{\Delta}$, and \mathbf{A} above may be derived as stationarity conditions from the Luttinger–Ward free-energy functional.^{35,63} For the theory to be conserving, the stationarity equations are to be solved to self-consistency. At convergence, we solve Eq. (10) with $\Delta(\mathbf{p}_F, \mathbf{R})$ and $\mathbf{A}(\mathbf{R})$, recalculate Eqs. (3) and (7), and get back the same $\Delta(\mathbf{p}_F, \mathbf{R})$ and $\mathbf{A}(\mathbf{R})$ within some given tolerance. Current conservation is then fulfilled to the same tolerance. At self-consistency, we have found an extremum of the quasiclassical version of the Luttinger–Ward functional.^{35,56,64} This functional always give a measure of the free energy with respect to the normal state,³⁵ $\Omega_S[T] - \Omega_N[T]$, and goes to zero as $\Delta \rightarrow 0$. To simplify notation, we denote this free energy difference $\Omega[T]$, and it has the form,

$$\Omega[T] = \int d\mathbf{R} \left\{ \frac{|\mathbf{B}_{\text{ind}}(\mathbf{R})|^2}{8\pi} + N_F \sum_{\Gamma} |\Delta_{\Gamma}(\mathbf{R})|^2 \ln \frac{T}{T_c^{\Gamma}} + \pi N_F k_B T \sum_n \frac{|\Delta(\mathbf{R})|^2}{|\varepsilon_n|} - \mathcal{I}(\mathbf{R}) \right\}, \quad (17)$$

where $\mathbf{B}_{\text{ind}}(\mathbf{R})$ is the induced magnetic field, $|\Delta(\mathbf{R})|^2 = \langle |\Delta(\mathbf{p}_F, \mathbf{R})|^2 \rangle_{\mathbf{p}_F}$, and

$$\mathcal{I}(\mathbf{R}) = \int_0^1 d\lambda N_F k_B T \sum_n \langle \tilde{\Delta}(\mathbf{p}_F, \mathbf{R}) f_{\lambda}(\mathbf{p}_F, \mathbf{R}; \varepsilon_n) + \Delta(\mathbf{p}_F, \mathbf{R}) \tilde{f}_{\lambda}(\mathbf{p}_F, \mathbf{R}; \varepsilon_n) \rangle_{\mathbf{p}_F}. \quad (18)$$

The auxiliary propagator \hat{g}_{λ} is obtained by solving the Eilenberger equation with scaled self-energy field $\lambda \hat{\Delta}$ so the integral over the dummy variable λ in Eq. (18) may be evaluated. In the framework SuperConga, we also apply the original Eilenberger-form of the free-energy functional,³⁶ in which case the term with the λ -integration is simplified to

$$\mathcal{I}(\mathbf{R}) = \pi N_F k_B T \sum_n \left\langle \frac{\tilde{\Delta}(\mathbf{p}_F, \mathbf{R}) f(\mathbf{p}_F, \mathbf{R}; \varepsilon_n) + \Delta(\mathbf{p}_F, \mathbf{R}) \tilde{f}(\mathbf{p}_F, \mathbf{R}; \varepsilon_n)}{\pi + i g(\mathbf{p}_F, \mathbf{R}; \varepsilon_n)} \right\rangle_{\mathbf{p}_F}. \quad (19)$$

The advantage of the Eilenberger-form is that it can be evaluated without the λ -integration at little computational cost. However, it must be used with some caution as it has been shown to not be generally valid and final results for the free energy needs to be confirmed by Eq. (17).⁶⁰

The above form of the free energy is consistent with the following regularized gap equation:

$$\Delta_{\Gamma}(\mathbf{R}) \ln \frac{T}{T_c^{\Gamma}} = T \sum_n \left\langle \eta_{\Gamma}^*(\mathbf{p}_F) \left[f(\mathbf{p}_F, \mathbf{R}; \varepsilon_n) - \frac{\pi \Delta(\mathbf{p}_F, \mathbf{R})}{|\varepsilon_n|} \right] \right\rangle_{\mathbf{p}_F}, \quad (20)$$

for superconductivity with the symmetry given by the representation Γ . The dependencies on the pairing interaction strength V_{Γ} and the Matsubara sum cutoff ε_c have been replaced by the measurable transition temperature T_c^{Γ} . After this regularization, the Matsubara sum is not cutoff dependent. The connection between Eqs. (3) and (20) can be seen as replacing the coupling strength by

$$\frac{1}{N_F V_{\Gamma}} \rightarrow \ln \frac{T}{T_c^{\Gamma}} + \pi T \sum_n \frac{1}{|\varepsilon_n|}. \quad (21)$$

By moving the second term on the right-hand side of Eq. (20) over to the left-hand side, and using the fact that the basis functions are orthonormal, $\langle \eta_i^*(\mathbf{p}_F) \eta_j(\mathbf{p}_F) \rangle_{\mathbf{p}_F} = \delta_{ij}$, we can divide down the resulting prefactor of $\Delta_{\Gamma}(\mathbf{R})$, obtaining the gap equation in a fix point form,

$$\Delta_{\Gamma}(\mathbf{R}) = \frac{T \sum_n \langle \eta_{\Gamma}^*(\mathbf{p}_F) f(\mathbf{p}_F, \mathbf{R}; \varepsilon_n) \rangle_{\mathbf{p}_F}}{\ln \frac{T}{T_c^{\Gamma}} + \pi T \sum_n \frac{1}{|\varepsilon_n|}}. \quad (22)$$

The currently available symmetries and the corresponding basis functions $\eta_{\Gamma}(\mathbf{p}_F)$ are listed in Table III in Appendix F.

A. Riccati formalism for inhomogeneous states

For inhomogeneous superconducting states we need to solve the first order differential equation for the propagator $\hat{g}(\mathbf{p}_F, \mathbf{R}; \varepsilon)$ in Eq. (10), subject to the normalization condition in Eq. (11). The gradient term $i\mathbf{v}_F \cdot \nabla_{\mathbf{R}} \hat{g}(\mathbf{p}_F, \mathbf{R}; \varepsilon)$ couples the momentum direction and spatial coordinates. Quasiparticle trajectories are then defined by this directional derivative and we may introduce a trajectory coordinate s as

$$\mathbf{R}(s) = \mathbf{R}_{\min} + s \hat{\mathbf{v}}_F, \quad (23)$$

where $\hat{\mathbf{v}}_F = \mathbf{v}_F/v_F$ is a unit vector. Starting with an initial value, for instance at a boundary point \mathbf{R}_{\min} , the propagator should be found by integrating along the trajectory until another boundary point \mathbf{R}_{\max} is met. At that point, a boundary condition must be supplemented to the theory. In general, boundaries present strong perturbations and their treatment lay outside the strict validity of the quasiclassical approximation. Nevertheless, one can model scattering off surfaces or at semitransparent interfaces via effective boundary conditions derived and carried over from a more microscopic theory.^{35,47,62,65–70} In certain cases, boundary and interface effects may arise, which lay outside the standard quasiclassical approximation.^{71–73} In SuperConga a simple specular boundary condition is used at all boundaries, see illustration in Fig. 1.

In practice, the most convenient and stable formulation is based on a parametrization^{62,74–76} of the Green's function in terms of coherence functions $\gamma(\mathbf{p}_F, \mathbf{R}; \varepsilon)$ and $\tilde{\gamma}(\mathbf{p}_F, \mathbf{R}; \varepsilon)$. They are related by the symmetry in Eq. (13) and quantify electron–hole coherence along the trajectory in the superconducting state. The quasiclassical Green's function can be written in terms of them as

$$\hat{g} = \frac{-i\pi}{1 + \gamma\tilde{\gamma}} \begin{pmatrix} 1 - \gamma\tilde{\gamma} & 2\gamma \\ 2\tilde{\gamma} & -1 + \gamma\tilde{\gamma} \end{pmatrix}, \quad (24)$$

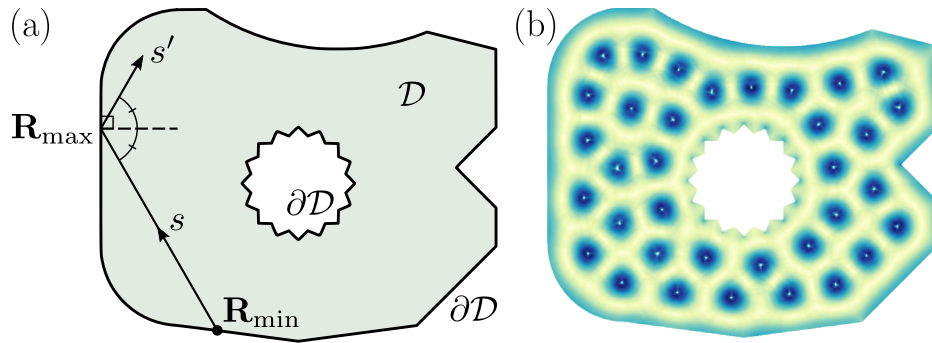


FIG. 1. (a) The system domain is denoted \mathcal{D} and has a boundary $\partial\mathcal{D}$. A quasiparticle trajectory parameterized by the coordinate s starts at one boundary point \mathbf{R}_{\min} and ends at another boundary point \mathbf{R}_{\max} . The specular boundary condition connects incoming Riccati amplitudes on trajectory s to starting Riccati amplitudes on another trajectory s' by simply requiring continuity of the amplitudes, i.e., $\gamma(s') = \gamma(s)$ at the boundary point. (b) Self-consistently computed vortex lattice in the same geometry as in panel (a), see Secs. III–V for more details.

where we for brevity dropped the explicit dependencies on \mathbf{p}_F , \mathbf{R} , and ε . The normalization of the Green's function in Eq. (11) is now automatically satisfied. The coherence functions satisfy a set of Riccati equations,

$$i\mathbf{v}_F \cdot \nabla \gamma + 2 \left(\varepsilon + \frac{e}{c} \mathbf{v}_F \cdot \mathbf{A} \right) \gamma + \tilde{\Delta} \gamma^2 + \Delta = 0, \quad (25)$$

$$i\mathbf{v}_F \cdot \nabla \tilde{\gamma} - 2 \left(\varepsilon + \frac{e}{c} \mathbf{v}_F \cdot \mathbf{A} \right) \tilde{\gamma} + \Delta \tilde{\gamma}^2 + \tilde{\Delta} = 0. \quad (26)$$

The equation for $\gamma(\mathbf{p}_F, \mathbf{R}; \varepsilon)$ is stable to integrate in the direction parallel to \mathbf{v}_F , while the equation for $\tilde{\gamma}(\mathbf{p}_F, \mathbf{R}; \varepsilon)$ is stable to be integrated in the opposite direction. After introducing the trajectory coordinate, we may let $\mathbf{v}_F \cdot \nabla \gamma \rightarrow v_F \partial_s \gamma$.

III. OVERVIEW OF THE IMPLEMENTATION

In this section, we give a brief overview of the main algorithm of SuperConga and describe a few key features of the framework. The aim of this section is to explain why certain parameters related to the implementation have to be set and how the parameter choices may influence the simulations.

A. The main algorithm of SuperConga

We solve the Riccati equations, Eqs. (25) and (26), numerically (see Appendix C) along trajectories through the grain. An example of a trajectory is shown in Fig. 1. Given an initial boundary value, $\gamma(\mathbf{R}_{\min})$, we solve Eq. (25) until we reach \mathbf{R}_{\max} . Similarly, given an initial boundary value, $\tilde{\gamma}(\mathbf{R}_{\max})$, we solve Eq. (26) until we reach \mathbf{R}_{\min} .

Clearly, the boundaries are special points in that we need a starting value for $\gamma(\mathbf{R}_{\min})$ and we obtain after stepping along the trajectory an end value $\gamma(s_N)$ at another boundary, where N is the number of steps taken along the trajectory s . All boundary values are related to each other through the boundary condition. The complication is then that the end values become, after the boundary condition has been applied, a starting value for a different trajectory. This can be solved by having an initial guess for all boundary starting values and self-consistently computing the boundary values by repeatedly stepping along all trajectories and applying the boundary condition. Since the order parameter $\Delta(\mathbf{R})$ is also solved self-consistently by iteration, the update of the boundary values can be done in parallel. At the end of

the computation, when the gap equation is satisfied, the boundary condition is also self-consistently satisfied. A caveat in SuperConga is that while the order parameter and vector potential are saved to file, the coherence functions are not, due to being too disk-space intensive. When restarting the calculation using a nominally self-consistent order parameter, the self-consistency of the boundary conditions must be reached again. This is solved via a “burn-in” period, where the order parameter and vector potential are kept constant for a few iterations when restarting a calculation, until the coherence functions and boundary conditions are also self-consistent.

In SuperConga, the order parameter and the vector potential have been discretized in two-dimensional (2D) real space: $\Delta(\mathbf{R}) = \Delta(x, y) \rightarrow \Delta(i, j)$ and $\mathbf{A}(\mathbf{R}) = \mathbf{A}(x, y) \rightarrow \mathbf{A}(i, j)$, where i and j are integers. The grid is a simple square grid with spacing h , i.e., $x = ih$ and $y = jh$, but we still take into account the exact location and shape of boundaries, such that there is no aliasing or similar artifacts. Figure 2 visualizes the implementation and discretization. The Fermi surface is parameterized by an azimuthal angle $\varphi \in [0, 2\pi)$ and discretized according to $\varphi_k = k\Delta\varphi$ (k integer). The Fermi surface averages are computed through numeric integration, via one of the following methods: Boole's, Simpson's 3/8, Simpson's 1/3, or the trapezoidal rule, depending on if the number of momenta is divisible by 4, 3, 2, otherwise, respectively. For a particular point φ_k on the Fermi surface, we obtain a Fermi velocity $\mathbf{v}_F(\varphi_k)$, which determines a direction for a set of trajectories in real space. Each trajectory coordinate has been discretized as outlined above. Considering all parallel trajectories for a certain point on the Fermi surface, the discrete grid in real space then forms a simple square grid with the same lattice spacing h , but rotated relative to the reference grid where the order parameter and vector potential are defined. Bilinear interpolation is then performed on the order parameter and the vector potential from the reference frame to the rotated frame. After solving the Riccati equations, the Riccati amplitudes are known in the rotated frame. The φ_k terms in the Fermi-surface averages for the order parameter, and the charge-current density are computed in the rotated frame by constructing the relevant matrix element of the Green's function, summing over energies, and then rotating back to the reference frame. In this way, the order parameter $\Delta(i, j)$, observables such as the current $\mathbf{j}(i, j)$, and also the vector potential $\mathbf{A}(i, j)$ can be updated through Eqs. (3)–(7).

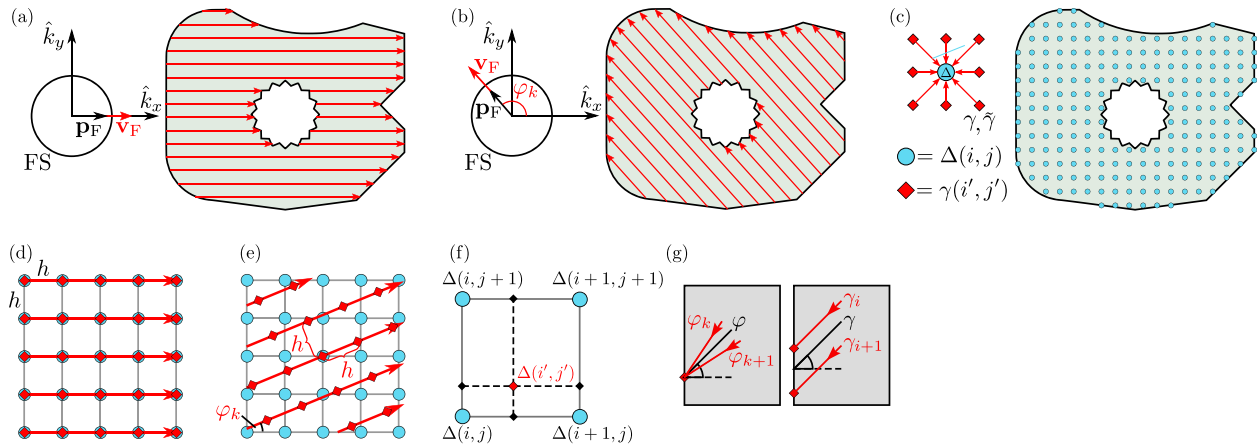


FIG. 2. Visualization of the implementation and main algorithm. (a) and (b) The Fermi surface is parametrized by the discrete angle φ_k and the Fermi velocity $v_F(\varphi_k)$, which together define quasiparticle trajectories in real space. As in Fig. 1, each trajectory has a well-defined start point R_{\min} and end point R_{\max} , which might either lie on an internal or external boundary. (c) The order parameter and vector potential are defined on a discretization of the geometry with grid spacing h and are computed as the Fermi-surface average of all trajectories passing through the point. (d)–(f) We let h be the discrete step-size along each trajectory, as well as the separation between trajectories, defining a square lattice. For nearly all angles, the discrete geometry is incommensurate with the trajectories, and interpolation becomes necessary. (g) Similarly, the discrete Fermi surface makes the reflection and the matching of incoming and outgoing trajectories incommensurate, such that for a particular trajectory starting at a boundary, no incoming trajectory exists for the corresponding angle or boundary coordinate. The solution is to interpolate from neighboring incoming trajectories, in both angle and boundary coordinate.

The boundary condition is problematic as the azimuthal angle of the specularly reflected momentum at a given surface might not exist in the set $\{\varphi_k\}$. Furthermore, due to the rotation of the grid, the trajectories hit the boundary at different points for each Fermi velocity $v_F(\varphi_k)$. In order to determine the initial conditions for the coherence functions, we once again perform bilinear interpolation. Here, the interpolation is between spatial coordinates on the boundary $R \in \partial\mathcal{D}$ and azimuthal angles.

An overview of the algorithm of SuperConga is shown in Algorithm 1. The user defines, e.g., the system domain, physical parameters, the discretization h in real space, the discretization $\Delta\varphi$ in momentum space, and an energy cutoff for the frequency summations [see, e.g., Eq. (3)]. The frequency summations are not over the usual Matsubara frequencies, but instead a more efficient summation due to Ozaki⁷⁷ has been implemented. In addition, a convergence criterion for the self-consistency must be set. The global error is defined as

$$\varepsilon_G = \frac{\|\Delta^i - \Delta^{i-1}\|_p}{\|\Delta^{i-1}\|_p}, \quad (27)$$

where i here denotes the iteration number in the self-consistency loop, and $p \in \{1, 2, \infty\}$ is set by the user. The loop continues until the error ε_G is smaller than the convergence criterion, or the given maximum number of iterations have been performed. The gap equation can be solved by direct fixed-point iteration, but in SuperConga a few methods for convergence acceleration have been implemented. In simplified terms, the accelerators may save a number of computed order parameters, $\{\Delta^i, \Delta^{i-1}, \Delta^{i-2}, \dots\}$, and make a more educated guess for Δ^{i+1} . When observables, the vector potential and free energy are also computed, the convergence criterion is the same as for the order parameter. All the details of the stepping along trajectories, interpolation, as well as the treatment of the boundaries are automatically taken care of by the API. The user must be aware of, however, that the discretization in real and momentum space can be a source of error, and it is necessary to check convergence by running the same problem with finer grids. Also, depending on the problem being solved, the path to self-consistency may not be direct. For the simplest examples in this paper, this is not a problem. However, for more advanced problems, it is advisable to experiment with different starting guesses for the order parameter and also carefully choose a suitable convergence acceleration strategy.

B. Framework overview

The main algorithm and functionality of SuperConga is implemented as a fairly modular framework, in the form of a *backend* written in C++ and CUDA.^{78,79} A user can in principle choose which modules to use and in what manner, to set up highly customized simulations, or even extend the existing functionality by exploiting the modularity. However, this requires writing C++ code that imports or

ALGORITHM 1. A sketch of the SuperConga main loop.

```

Define domain  $\mathcal{D}$  and input parameters
Initialize  $\Delta$  and  $A_{\text{ind}}$ 
Initialize  $\gamma_{\partial\mathcal{D}}$  and  $\tilde{\gamma}_{\partial\mathcal{D}}$  ▷ The boundary
while not converged do
  for all  $p_F, \varepsilon_n$  do
    Rotate  $\Delta$  and  $A$  so that  $\hat{y} \parallel v_F$ 
    Compute  $\gamma$  and  $\tilde{\gamma}$  along the  $y$ -axis ▷ See App. C
    Update  $\gamma_{\partial\mathcal{D}}$  and  $\tilde{\gamma}_{\partial\mathcal{D}}$  ▷ Write incoming  $p_F$ 
    Update  $\Delta, j$ , and  $\Omega$  ▷ Eqs. (22), (8), (17)
  Compute  $A_{\text{ind}}$  ▷ See App. D
  Check convergence ▷ Eq. (27)
   $\Delta, A_{\text{ind}} \leftarrow$  Accelerator ▷ See App. E
  Update  $\gamma_{\partial\mathcal{D}}$  and  $\tilde{\gamma}_{\partial\mathcal{D}}$  ▷ Boundary condition

```

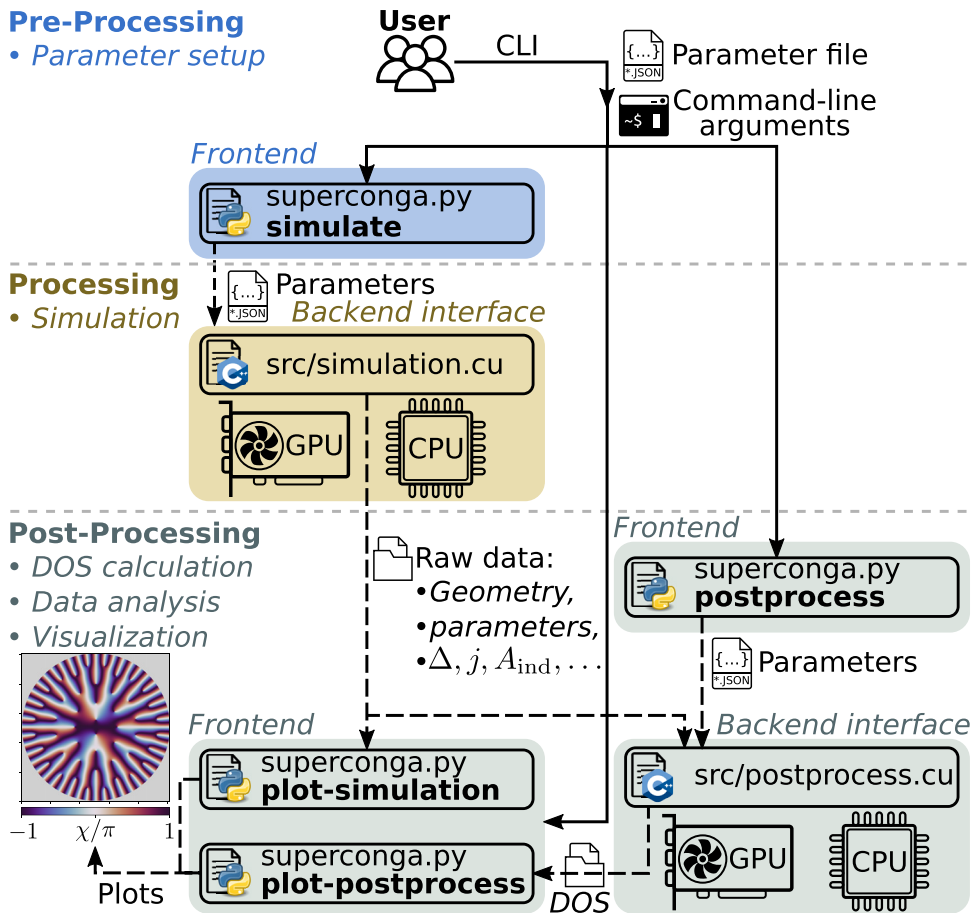


FIG. 3. A usage flow chart for SuperConga. A user can use command-line interface (CLI) to pass command-line arguments and a parameter-file to the Python3 frontend `superconga.py`, which sanity checks the user input. If all input is valid, the frontend calls the binary built from the C++/CUDA backend interface `simulate.cu`. The latter includes the appropriate functionality from the backend framework (implemented as header files) and runs self-consistent simulations according to Algorithm 1. The user can, for example, choose to enable live visualization and printing of simulation status to the terminal, and how frequently data should be saved to file. The user can perform data analysis directly on the raw data with their favorite external tool, or use the provided Python frontend, e.g., via `python superconga.py plot-simulation` and `python superconga.py plot-convergence`. Additionally, the data can be used in a postprocessing step to compute the DOS and LDOS via `python superconga.py post-process` (which calls the binary built from the backend interface `postprocess.cu`). Finally, the DOS and LDOS can be illustrated with the interactive spectroscopy tool `python superconga.py plot-postprocess`. Note that the user can also write their own backend interfaces to harness the full functionality of the SuperConga framework.

adds to the desired features. While this is certainly possible and a valid approach, a main goal of SuperConga is to also enable users to perform advanced simulations without having to write any code, by instead only specifying simulation parameters, but without losing crucial functionality. To make this possible, SuperConga comes with a user friendly *frontend*, and a set of *backend interfaces*. The frontend manages and validates the user input, and then sends it to the appropriate backend interface. The backend interface takes care of importing and calling the necessary backend modules with the settings provided by the user. See Fig. 3 for a schematic overview. SuperConga also comes with a set of ready-to-run examples (see Sec. IV for a demonstration), a detailed user manual full of tutorials and guides,⁸⁰ a set of helpful tools, a Singularity container which simplifies setup and improves portability, and an extensive testing framework to test the validity of the code. The latter ranges from smaller unit tests of basic functionality to larger end-to-end physics test of, e.g., current conservation and well-known analytic results. In the following, we give further insight into the backend and frontend, and list the most important modules.

1. Backend

The backend modules are implemented as a set of classes and functions via header files. The modularity is typically built up by an inheritance structure, from virtual base classes to derived classes. The

virtual base classes describe the general signature or pattern for a module but does not necessarily contain any functionality. This is instead taken care of by the derived classes, which inherits the signature from the corresponding virtual class, and implements the actual functionality (or offloads it to external libraries, see App. A for a description of dependencies). For example, `BoundaryCondition.h` contains a virtual base class defining the general properties of a boundary condition. The derived class in `BoundaryConditionSpecular.h` inherits from `BoundaryCondition.h` and implements a specular boundary condition. Similarly, there are virtual base classes for Fermi surfaces, geometric shapes, and even Riccati solvers, with specific implementations of, e.g., a circular Fermi surface, disk- and polygon-geometries, and a Riccati solver for finite grains. The idea is that the modularity should allow for straightforward extension of anything from boundary conditions to new observables. Below is a list of the most important modules:

Boundary conditions: Implementation of the boundary conditions to handle interface scattering of quasiparticles, the latter which follow the Eilenberger–Riccati trajectories. Currently only implements specular boundary conditions for superconductor-vacuum interfaces.

Fermi surface: A class that takes care of the momentum-averaging over the Fermi surface. Can either treat circular Fermi-surfaces, or

parametrize a more general Fermi-surface shape based on a tight-binding hopping model.

Geometry: Classes and functionality for creating mesoscopic grains from basic shape primitives (i.e., disks and polygons).

Observables: Implementation of various observables and quantities, specifically the superconducting gap, charge current density, magnetic moment, LDOS, free energy, vector potential, and the magnetic induction. Also, takes care of computing the residual for the self-consistency iteration.

Accelerator: Implementation of self-consistency accelerators. The user can currently choose from basic Picard iterations, Polyak's⁸¹ "small heavy sphere," a variant of the Barzilai–Borwein (BB) method,⁸² and CongAcc which is an adaptive method developed for SuperConga. See [Appendix E](#) for details and comparisons.

Order parameter: Class for setting up the order parameter, possibly with multiple components like $d + is$ and chiral d -wave. Note that the implementation is currently limited to spin-singlet with a single band and spin degenerate Fermi-surfaces.

Riccati solver: Implements a solver for the Riccati equations, currently based on the mid-point method for confined (finite) geometries. To give an idea, this class could more generally be extended with, e.g., a higher order Runge–Kutta solver, or other types of systems like bulk or semi-infinite superconductors.

Self-consistency solver: A collection of classes that take care of adding the contribution of the Riccati trajectories for each degree of freedom (energy, Fermi-surface angle, coordinate) to self-consistently solve the order parameter from the superconducting gap equation, together with the vector potential from the corresponding Maxwell equation.

See the API documentation for more modules and information.⁸⁰

SuperConga currently comes with two backend interfaces, that expose the functionality of the above modules to the user (via the frontend). The first backend interface is `simulation.cu`, which acts as the "main" program, essentially implementing Algorithm 1 to self-consistently solve the gap equation and Maxwell's equation. More specifically, `simulation.cu` starts by generating a superconducting grain and Fermi surface based on the user-provided geometry and model. The superconducting order parameter and its components are initialized in this grain, according to the specified basis functions, transition temperatures, and start guess. The program then continues to set up the self-consistency accelerator, Riccati solver, boundary conditions, observables, and vector potential. An OpenGL instance is initialized if live visualization is enabled. Finally, the self-consistency loop is run until reaching either the convergence criterion or maximum number of iterations, upon which data are saved to disk. The second backend interface is `postprocess.cu`, which computes the LDOS based on results from a previous simulation (e.g., computed with `simulation.cu`).

Note that both backend interfaces take as command-line input the location of a configuration file, which should contain all parameters necessary to completely specify the simulation. The simulation will not run if the configuration file is not found, or if the contents are invalid. The purpose of the frontend is to make it easier to set up all simulation parameters correctly, with validation and proper help messages to guide the user if anything goes wrong.

2. Frontend

The SuperConga frontend consists of a main run-file, `superconga.py`, which draws on an assortment of functionality implemented in a modular Python library, the latter located in the folder `frontend/`. The goal of the frontend is to facilitate setting up and analyzing simulations. For example, the frontend consists of Python modules for parsing command-line arguments, validating them and providing help for the user, reading and writing configuration files used by the backend, calling the correct backend interfaces, reading and visualizing simulation data, converting data between different formats, and modifying data by, e.g., adding noise. The functionality is divided into a set of subcommands, used via `python superconga.py <subcommand>`. A list of available frontend subcommands is obtained by calling `python superconga.py -help`, and each subcommand also has its own help message. A short summary of each subcommand will now be given, and Sec. IV demonstrates how to use them.

The first subcommand is `setup`, which is used to setup and generate the SuperConga build system via CMake.

The second subcommand is `compile`, which is used after the setup to compile the SuperConga framework. These two steps have to be performed before any simulations can be run.

The third subcommand is `simulate`, which calls the binary generated from `simulation.cu` to run self-consistent simulations. It takes as input the location of a configuration file, defining all simulation parameters. Optionally, each parameter can be set or overridden via the command-line interface (CLI). The parameters will then be validated for errors, attempting to provide the user with helpful messages if failing. If valid, the binary is called with the user-defined settings.

The fourth subcommand is `plot-simulation`, which takes the location of data from a self-consistent simulation, and plots computed quantities (see Fig. 4 for example). The user can control what to plot, in what order, and which settings to use, for example, color-maps and fonts. If the user has specified that data should be saved when running `simulate`, then `plot-simulation` is automatically called, using the default settings, and a PDF is saved together with the data.

The fifth subcommand is `postprocess`, which calls the binary generated from `postprocess.cu` to calculate the LDOS from self-consistently converged data. The runner takes as command-line input the location of the data, and parameters specifying which resolution to compute the LDOS with. The latter parameters can either be set directly via the CLI, or specified in an external configuration file.

The sixth subcommand is `plot-postprocess`, which is a spectroscopy tool. It takes as input the path to a directory containing LDOS data computed with `postprocess`. If found, the data are read and visualized in an interactive plot (see Fig. 5 for example).

The seventh subcommand is `plot-convergence`, which plots the residuals of computed quantities vs iteration number. This tool can be run on convergence data from either `simulate` or `postprocess` subcommands.

The eighth subcommand is `convert`, which is used to convert data back and forth between the file formats `h5` and `csv`.

Having given an overview of the SuperConga framework and its frontend, we will now demonstrate how to use it to set up simulations and do data analysis.

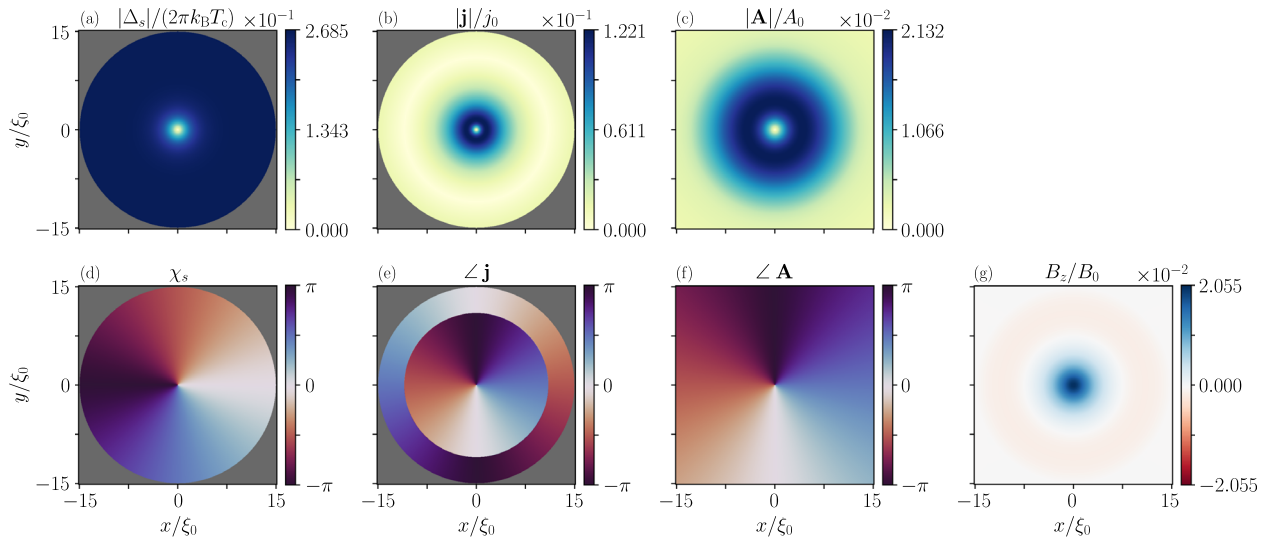


FIG. 4. Plot generated with the SuperConga “plot-simulation” command, visualizing the example with a single Abrikosov vortex in a conventional superconducting disk, by following Listing 2 then 3. Here, $T = 0.5T_c$, $B_{\text{ext}} = 1.5\Phi_0/\mathcal{A}$, and $\kappa = 5$ with full back-coupling of the magnetic gauge field. (a) The magnitude of the order parameter. (b) The magnitude of the charge-current density. (c) The magnitude of the induced vector potential. (d) The phase of the order parameter. (e) The polar angle (i.e., direction) of the charge-current density. The paramagnetic and diamagnetic regions are clearly distinguished. (f) The polar angle of the induced vector potential. (f) The induced magnetic-flux density.

IV. DEMONSTRATION OF SUPERCONGA

The SuperConga framework ships with a few ready-to-run examples that illustrate how to set up and run simulations. These are also explained in the online documentation, together with detailed tutorials and research guides.⁸⁰ One of the examples will now be shown, namely, the simulation of a single Abrikosov vortex in a conventional

superconducting disk. The demonstration can be split into the following generic steps:

Pre-processing: Setting up and defining the simulation,

Processing: Running the self-consistency simulations, optionally with live visualization,

Post-processing: Data analysis, visualization and spectroscopy.

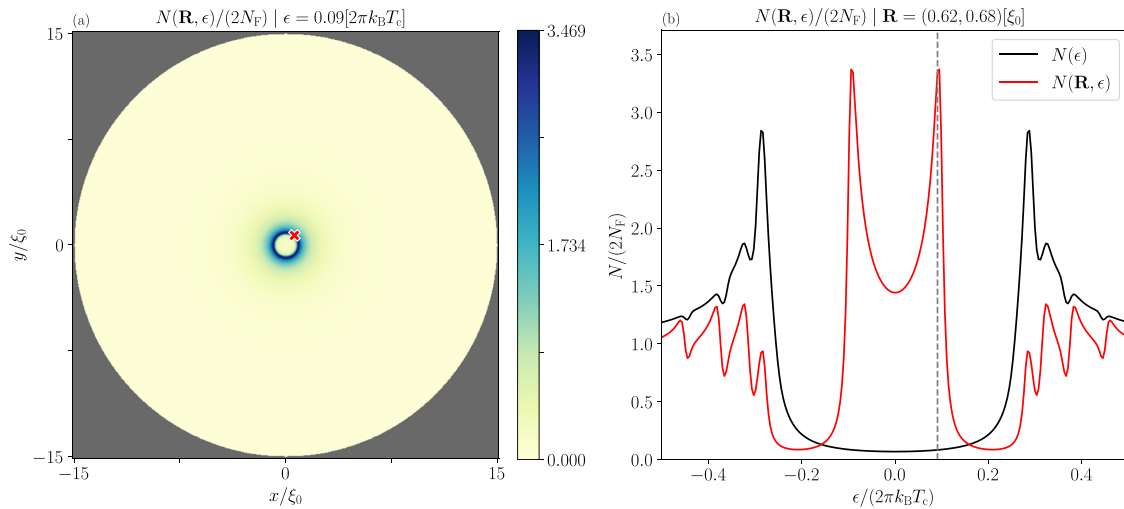


FIG. 5. Screenshot of the SuperConga interactive spectroscopy command, “plot-postprocess.” The plots show the local density of states (LDOS), $N(\mathbf{R}, \epsilon)$, for a single vortex at the center of a conventional superconducting disk, after following Listings 2, 4, and then 6 in order. Here, $T = 0.5T_c$, $B_{\text{ext}} = 1.5\Phi_0/\mathcal{A}$, and $\kappa = 5$ with full back-coupling of the magnetic gauge field. An energy broadening of $\delta = 0.008 \cdot 2\pi k_B T_c$ was used. (a) Heatmap of the LDOS vs spatial coordinates, at the finite energy shown in the title, as well as by the vertical dashed line in panel (b). (b) LDOS (solid red curve) at the coordinates clicked by the user in the heatmap (indicated by the red cross) in panel (a), and area-averaged DOS (solid black curve). The user can change the energy shown in panel (a) by clicking in panel (b). Similarly, the user can change the spatial point being visualized in panel (b) by clicking in panel (a).

The simulation configuration for the vortex example can be found in `examples/swave_disc_vortex/` in the SuperConga main folder.⁸³ Appendix B lists the full contents of this file, with a detailed explanation of the format. In short, each configuration file contains all necessary parameters needed to completely specify a simulation. The parameters are grouped into different sections: Listing 1 shows the group of “physics” parameters for the Abrikosov example, specifying the temperature $T = 0.5T_c$, the external flux $\Phi_{\text{ext}} = 1.5\Phi_0$, the penetration depth $\lambda = 5\xi_0$, and an s -wave order parameter with a phase winding of -2π as a start guess.

LISTING 1: Part of `simulation_config.json` in `examples/swave_disc_vortex/`.

```

“physics”: {
  “temperature”: 0.5,
  “external_flux_quanta”: 1.5,
  “penetration_depth”: 5.0,
  “crystal_axes_rotation”: 0.0,
  “gauge”: “symmetric”,
  “charge_sign”: -1,
  “order_parameter”: {
    “s”: {
      “critical_temperature”: 1.0,
      “initial_phase_shift”: 0.0,
      “initial_noise_stddev”: 0.0,
      “vortices”: [
        {
          “center_x”: 0.0,
          “center_y”: 0.0,
          “winding_number”: -1.0
        }
      ]
    }
  }
}

```

Other parameter groups deal with, for example, the geometry, numerical accuracy, convergence criteria, convergence acceleration, and so on. The parameters can also be set or changed directly via command line, effectively overriding the settings in the configuration file, as shown below. Assuming that the framework has been properly installed and compiled (see the online documentation and installation guide⁸³) the configuration file can be used to start the simulation by running the following from the root directory of SuperConga,

LISTING 2: Running the example with a single Abrikosov vortex in a conventional superconducting disk.

```
python superconga.py simulate -C examples/
swave_disc_vortex/simulation_config.json
```

which should run a few tens of iterations until the convergence criterion in `simulation_config.json` is fulfilled, printing simulation

status for each iteration to terminal. Here, the flag `-C` is used to specify the relative path to the configuration file. Adding the arguments `“-T 0.1 -B 10.0”` will change the temperature to $0.1T_c$, and the external flux to $10\Phi_0$. To start with an order parameter from a file, corresponding to, e.g., a previous simulation or an arbitrary computed guess, the `-L` argument can be used. Using the flag `--help` gives further information about which parameters are available and their usage.

We note that live visualization can be turned on or off with the flags `--visualize` and `--no-visualize`, respectively. The main purpose of the live-visualization is to get an overview of the simulation progress. Thus, it is geared toward speed rather than producing production-ready plots. For the latter, we instead recommend to visualize the fully converged results, via the data files generated by the program. The user can either use their favorite plotting tool, or use SuperConga as described in the following. Having followed the example in Listing 2, the Abrikosov vortex results can be visualized by running the following from the root directory.

LISTING 3: Plotting all spatially dependent quantities in the Abrikosov vortex example.

```
python superconga.py plot-simulation -L data/
examples/swave_disc_vortex
```

This plots all the computed quantities, as shown in Fig. 4. Various properties of the plot can be controlled via command-line arguments (as described by the help message), like fonts, colormaps, which quantities are plotted, and if saving the plot directly to file rather than drawing in a window. The latter makes it easy to automatically generate plots from a large number of simulations.

The local density of states (LDOS) can be calculated from converged data as a post-processing step, by calling,

LISTING 4: Computing the LDOS for the Abrikosov vortex example.

```
python superconga.py postprocess -C examples/
swave_disc_vortex/postprocess_config.json
```

where the contents of `postprocess_config.json` is shown in Listing 5.

LISTING 5: Configuration file used for computing (postprocessing) the LDOS: `postprocess_config.json` in `examples/swave_disc_vortex/`.

```

{
  “spectroscopy”: {
    “energy_max”: 0.5,
    “energy_min”: 0.0,
    “energy_broadening”: 0.008,
    “num_energies”: 128
  },
  “numerics”: {
    “convergence_criterion”: 1e-4,

```

```

“norm”: “l2”,
“num_energies_per_block”: 32,
“num_fermi_momenta”: 720,
“num_iterations_burnin”: -1,
“num_iterations_max”: 1000,
“num_iterations_min”: 0
},
“misc”: {
  “data_format”: “h5”,
  “load_path”: “data/examples/swave_disc_vortex”,
  “save_path”: “”,
  “verbose”: true
}
}

```

This file specifies, e.g., which energies and resolution to use in the LDOS calculation, and where the order-parameter data are located. The empty save path indicates that the LDOS data will be saved to the same directory as the order parameter data.

The LDOS can be particularly tricky to visualize due to the high dimensionality. To aid with this, the framework provides a spectroscopy tool in the form of an interactive LDOS visualizer. Assuming that the vortex LDOS has been calculated with the above post-processing example in Listing 4, vortex spectroscopy can be done by running the following from the root directory:

LISTING 6: Performing vortex spectroscopy.

```

python superconga.py plot-postprocess -L data/
  examples/swave_disc_vortex

```

Figure 5 shows a screenshot of the interactive LDOS plotter for this example, illustrating that the user can click in the window to choose which energy to plot the LDOS at (as a 2D-heatmap vs coordinates), and which point in space to plot the LDOS vs energy at (as a 1D curve).

To extract the temperature dependence, or the dependence on any other parameter for that matter, the most straightforward approach is to write a script which calls the program multiple times, but with unique values of the parameter. A few such examples are included in SuperConga, and the following one illustrates how to simulate an Abrikosov vortex for different values of Ginzburg–Landau coefficient κ , throughout the whole type-II range $\kappa \in [1, \infty)$:

LISTING 7: Simulating an Abrikosov vortex, for various values of the Ginzburg–Landau coefficient.

```

./examples/parameter_sweeps/
  swave_disc_vortex_kappa_sweep.sh

```

The results, seen in Fig. 6, can subsequently be visualized via

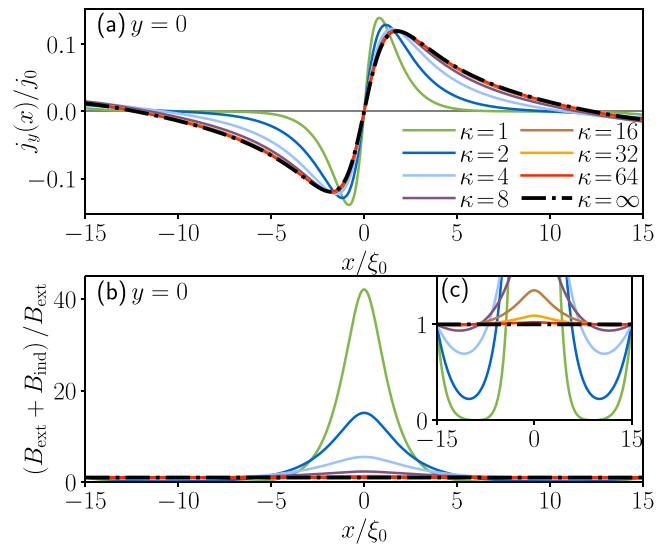


FIG. 6. Spatial dependence of the (a) charge-current density and the (b) total magnetic-flux density, as distance from a vortex core. The system is an s-wave disk with radius $R = 15\xi_0$ at temperature $T = 0.5T_c$, and external flux $\Phi_{\text{ext}} = 1.5\Phi_0$ (the same as in Figs. 4 and 5). Different lines correspond to different values of κ , as indicated by the legend. Panel (c) is a zoom of (b), illustrating more clearly the diamagnetic and paramagnetic regions. Due to the small radius, the system is only partially screened. Full screening ($B_{\text{ext}} + B_{\text{ind}} = 0$) is only achieved in a narrow region for $\kappa = 1$. Increasing the ratio R/λ will increase the region of full screening.

LISTING 8: Plotting the vortex dependence on penetration depth.

```

python examples/parameter_sweeps/
  plot_vortex_kappa_sweep.py

```

So far, the demonstration has focused on a superconducting disk. SuperConga allows the user to specify more general composite geometries by successively adding and removing disks and polygons. Both regular and free-form polygons are implemented, enabling quite intricate systems that, e.g., has holes, are multiply connected, or even disconnected but coupled via induction. For example, Fig. 1 was created in this way by setting the “geometry” parameter section of the configuration file. Listing 9 shows a SQUID-like geometry, created by adding a central region shaped like an octagon, two arms in the form of a rectangle, and then removing a disk from the central region:

LISTING 9: Example setup of a composite geometry.

```

“geometry”: [
  {
    “polygon”: {
      “add”: true,
      “vertices_x”: [0.0, 0.0, 100.0, 100.0],
      “vertices_y”: [5.0, -5.0, -5.0, 5.0]
    }
  },

```



```

{
  "regular_polygon": {
    "add": true,
    "center_x": 50.0,
    "center_y": 0.0,
    "num_edges": 8,
    "rotation": 0,
    "side_length": 15.0
  },
  {
    "disc": {
      "add": false,
      "center_x": 50.0,
      "center_y": 0.0,
      "radius": 8.0
    }
  }
}

```

In Listing 9, the entry “geometry” consists of a list which specifies in which order (from top to bottom) the shapes should be added or removed, the latter indicated by “add”: true or “add”: false, respectively. All coordinates and lengths are given in units of ξ_0 , and the user can choose whichever origin they prefer, as it does not influence the physics. The disk is uniquely defined via its center coordinates and radius, a regular polygon by center coordinates, number of edges, rotation (in units of 2π), and side-length. For free-range polygons, a list of x - and y -coordinates for the vertices are specified (in counterclockwise order). Currently, the free-range polygons are limited to convex hulls. Note that an arbitrary number of each shape in principle can be listed, which was previously used to, e.g., study mesoscopic roughness by generating boundaries with random removal of polygons.⁸⁴ Note, however, that for extremely large number of added/removed shapes, the performance might eventually be reduced due to having to deal with a large set of complicated trajectories and boundary conditions.

It is straightforward to extend the above example of a single Abrikosov vortex, to vortex lattices, by increasing the external field. As an example, Fig. 7 plots the zero-energy LDOS in nanoscale grains of various geometries exposed to high external fields. Vortex cores show up as dot-like dark spots, illustrating vortex lattices with various mesoscopic properties. In contrast to bulk lattices, the number of vortices and their separation in a finite system does not only depend on the external field strength, but also on, e.g., the boundary conditions, the system size relative to the penetration depth, and the shape of the system. While bulk lattices are usually triangular, a mesoscopic vortex lattice can mimic the shape and symmetry of the superconducting grain. In addition, rather than entering one by one as the external field is increased, several vortices might suddenly enter at a time, sometimes as concentric shells (hence referred to as the vortex-shell effect). These effects have been observed, e.g., in grains shaped like disks,^{3,19,21} squares,^{21,55,88–91} triangles,^{21,92–95} and pentagons.⁹⁶ Such scenarios are great examples of when a sufficiently sophisticated simulation tool like SuperConga is crucial, since one needs to capture the combined effects

of a confined geometry and a finite penetration depth with back-coupling of the induction. In particular, one needs to use a powerful convergence accelerator that can properly traverse the phase space. For example, we note that with the proper choice of start guess and accelerator settings, it is in principle possible to study giant vortices,^{97–102} as well as vortex-antivortex pairs.^{90,91,103–106} Careful treatment and use of the adaptive convergence accelerators show, however, that these structures are generally unstable and decay to more traditional vortex structures with a lower free energy.

V. EXAMPLES AND RESULTS

In this section, we present several examples of studies that are quite straightforward to perform with the framework SuperConga. In a few special cases, it is possible to find analytic results, which enable us to check the overall correctness of the framework. First, we study a 2D superconducting annulus. In the case of an s -wave superconductor, we may compare with analytics, while in the d -wave superconducting case, we demonstrate the corrections due to suppression of the order parameter at the edges and the formation of spontaneous currents at low temperature.^{57,58,84,107–111} Second, we study a superconducting disk also subjected to an external magnetic field. Here, we return to the well-studied problem of vortex lattice formation, which has a long history going back to the original work of Abrikosov,¹¹² and in particular, to that of Pearl considering superconducting disks.¹¹³ For the cases of zero or one vortex at the center of the disk, we compare with analytic results. Then, we continue with higher fields, where we have to resort to a numerical treatment. Finally, we simulate irregular superconducting islands in an external magnetic field. The study is inspired by experimental work done by Timmermans *et al.*⁵⁵ on spectroscopy on superconducting nanostructures assembled by small squares of $\text{Mo}_{79}\text{Ge}_{21}$.

A. Superconducting annulus in a magnetic field

We begin by studying homogeneous superflow in a superconductor. Superflow implies that there is a finite superfluid momentum caused by phase gradients and/or a vector potential, as defined in Eq. (2). The fact that the phase gradient enters in the superfluid momentum becomes clear when doing a gauge transformation. In quasiclassical theory this is done as follows. We start with the order-parameter $\Delta(\mathbf{p}_F, \mathbf{R}) = \Delta(\mathbf{R})\eta(\mathbf{p}_F) \exp[i\chi(\mathbf{R})]$, which, in general, is complex valued. The order-parameter matrix can then be decomposed as

$$\hat{\Delta}(\mathbf{p}_F, \mathbf{R}) = \hat{U}(\chi) \hat{\Delta}_0(\mathbf{p}_F, \mathbf{R}) \hat{U}^\dagger(\chi), \quad (28)$$

with the transformation matrix

$$\hat{U}(\chi) = \begin{pmatrix} e^{i\chi(\mathbf{R})/2} & 0 \\ 0 & e^{-i\chi(\mathbf{R})/2} \end{pmatrix} = e^{i\chi(\mathbf{R})\hat{\tau}_3/2}, \quad (29)$$

and $\Delta_0(\mathbf{R})$ a purely real amplitude (i.e., $\chi_0(\mathbf{R}) \equiv 0$ while the basis function $\eta(\mathbf{p}_F)$ may still be complex valued). Applying the same transformation to the Eilenberger Eq. (10), suppressing the arguments of \hat{g} and $\hat{\Delta}$, we see that if \hat{g} solves

$$i\mathbf{v}_F \cdot \nabla_R \hat{g} + \left[\left(i\hat{e}_n + \frac{e}{c} \mathbf{v}_F \cdot \mathbf{A} \right) \hat{\tau}_3 - \hat{\Delta}, \hat{g} \right] = 0, \quad (30)$$

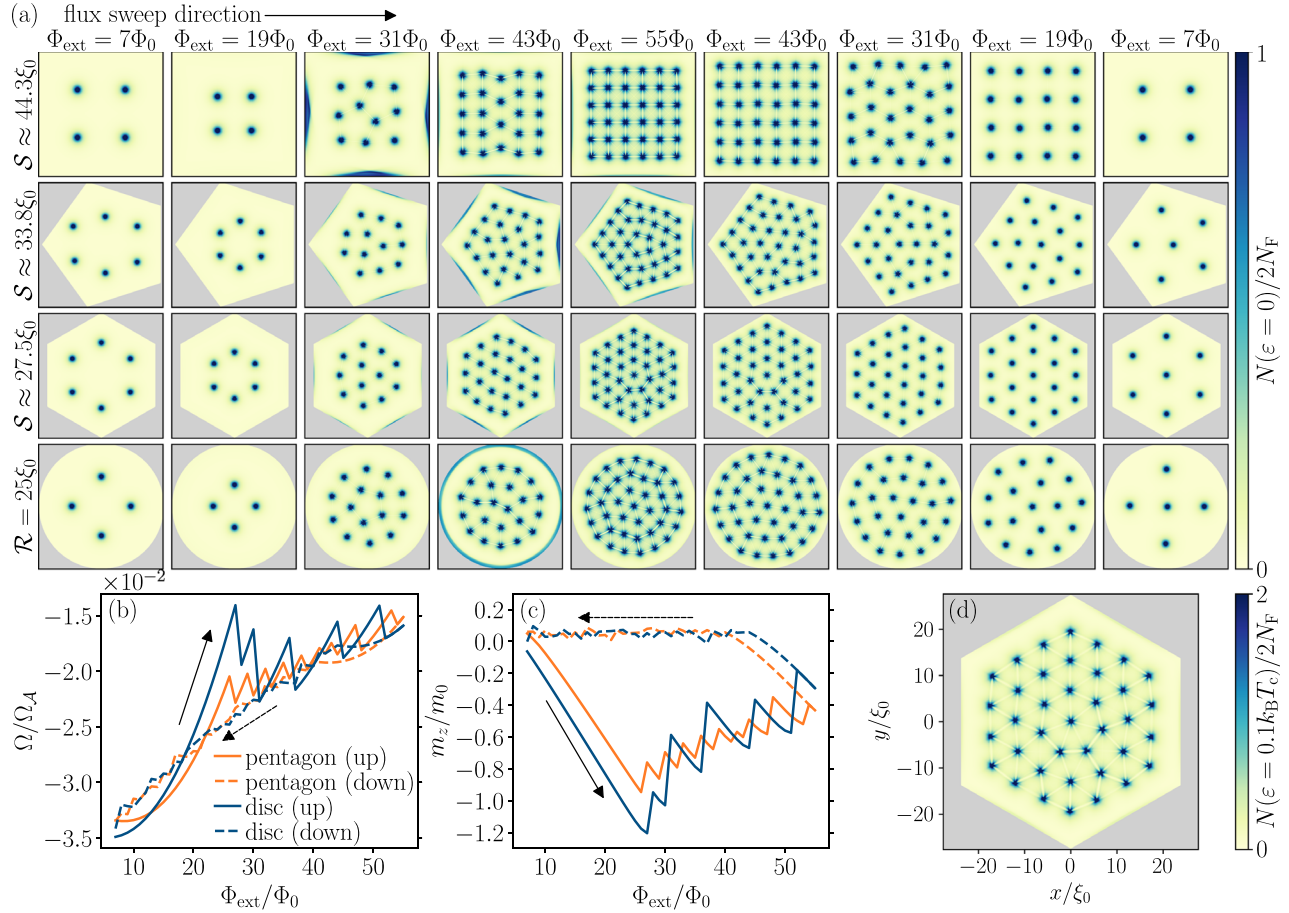


FIG. 7. (a) Mesoscopic vortex lattices in conventional s-wave superconducting grains of different shapes (rows), exposed to different external fluxes (columns) in a directed sweep. We set $\kappa = 10$ and $T = 0.1T_c$, and the grain side-lengths (S) was chosen such that the area is the same as that of the disk, $A = \pi R^2$. Colors indicate the LDOS at zero energy, with localized Caroli-de Gennes-Matricon states^{15,85,86} in each vortex core. To give the vortex cores a visible extent, a smearing of $\delta = 0.008 \cdot 2\pi k_B T_c$ was used, and an arbitrary cutoff of $2N_F$ was introduced. The plots illustrate that vortex lattices in mesoscopic superconductors can be highly modified by finite-size effects, such as the grain shape. In particular, as the external field increases, the vortices are packed into a tighter configuration, with the introduction of vortex shells, and depending on the flux-sweep strategy, vortex-lattice “dislocations” might enter the system. In this case, the flux is varied in steps of Φ_0 from low fluxes (left-most column) to high fluxes (middle column), and then from high fluxes to low fluxes again (right-most column). For each new value of the flux, the converged results from the previous simulation are used as a start guess. Different sweep directions lead to different local minima in the free energy. We note that far from the free-energy minimum, when there are few vortices compared to the external flux, there is significant pair-breaking and zero-energy states due to large screening currents at the edges of the system, see, e.g., square in the third column. Panels (b) and (c) show how the free energy and magnetic moment evolve using this directed-sweep strategy, respectively, for the pentagon (orange) and disk (blue). Full lines are for sweeping toward higher fields and dashed for sweeps in the opposite direction. This asymmetry in the free energy vs applied field, between the two sweep directions, can be interpreted as a Bean-Livingston barrier that makes vortex entry energetically harder than vortex exit.⁸⁷ (d) LDOS at a finite energy $\varepsilon = 0.9k_B T_c$, at external flux $\Phi_{\text{ext}} = 55\Phi_0$.

with $\hat{\Delta} = \hat{U} \hat{\Delta}_0 \hat{U}^\dagger$, then $\hat{g}_0 = \hat{U}^\dagger \hat{g} U$ is a solution to

$$i\mathbf{v}_F \cdot \nabla_R \hat{g}_0 + \left[\left(i\mathbf{e}_n + \frac{e}{c} \mathbf{v}_F \cdot \mathbf{A} - \frac{1}{2} \mathbf{v}_F \cdot \nabla \chi \right) \hat{\tau}_3 - \hat{\Delta}_0 \hat{g}_0 \right] = 0. \quad (31)$$

The superfluid momentum \mathbf{p}_s in Eq. (2) is now naturally formed, and we get the Eilenberger equation in the form

$$i\mathbf{v}_F \cdot \nabla_R \hat{g}_0 + \left[(i\mathbf{e}_n - \mathbf{v}_F \cdot \mathbf{p}_s) \hat{\tau}_3 - \hat{\Delta}_0 \hat{g}_0 \right] = 0, \quad (32)$$

with the purely real order-parameter matrix.

Let us look at the linear response to a small and spatially homogeneous superfluid momentum, $p_s \ll \Delta/v_F$. To do this, we write the propagator as a perturbation expansion $\hat{g} = \hat{g}_0 + \delta\hat{g} + \mathcal{O}[p_s^2]$. Using the normalization condition on the propagator $\hat{g}^2 = -\pi^2 \hat{1}$, we get

$$\hat{g}_0^2 = -\pi^2 \hat{1} \quad \text{and} \quad \{\hat{g}_0, \delta\hat{g}\} = 0.$$

Since $\hat{g}_0 \propto i\mathbf{e}_n \hat{\tau}_3 - \hat{\Delta}_0(\mathbf{p}_F)$, we get via the Eilenberger equation the solution

$$\delta\hat{g}(\mathbf{p}_F; \varepsilon_n) = \pi \frac{\mathbf{v}_F \cdot \mathbf{p}_s}{\Lambda_n^3(\mathbf{p}_F)} \begin{pmatrix} |\Delta(\mathbf{p}_F)|^2 & -i\mathbf{e}_n \Delta(\mathbf{p}_F) \\ i\mathbf{e}_n \Delta^*(\mathbf{p}_F) & -|\Delta(\mathbf{p}_F)|^2 \end{pmatrix}, \quad (33)$$

defining $\Lambda_n(\mathbf{p}_F) = \sqrt{|\Delta(\mathbf{p}_F)|^2 + \varepsilon_n^2}$. From the form of $\delta\hat{g}$, we can directly see that the linear correction to the anomalous Green's function component, $f(\mathbf{p}_F; \varepsilon_n)$, is odd in frequency and will not give a correction in leading order in p_s to the order parameter $\Delta(\mathbf{p}_F)$. The diagonal part of $\delta\hat{g}$ allows us to calculate the current to linear order in p_s as

$$\delta j = 2eN_F \langle \mathbf{v}_F(\mathbf{p}_F) Y_{3/2}(\mathbf{p}_F) \mathbf{v}_F(\mathbf{p}_F) \cdot \mathbf{p}_s \rangle_{\mathbf{p}_F}, \quad (34)$$

with the angle-dependent Yoshida function,

$$Y_{3/2}(\mathbf{p}_F) = \pi T \sum_n \frac{|\Delta(\mathbf{p}_F)|^2}{\Lambda_n^3(\mathbf{p}_F)}.$$

Equation (34) is usually written in the well-known form,

$$\mathbf{j}_s = e \langle \bar{\rho}_s(\mathbf{p}_F) \cdot \mathbf{p}_s \rangle_{\mathbf{p}_F}, \quad (35)$$

defining

$$\bar{\rho}_s(\mathbf{p}_F) = 2N_F Y_{3/2}(\mathbf{p}_F) \mathbf{v}_F(\mathbf{p}_F) \mathbf{v}_F^\top(\mathbf{p}_F), \quad (36)$$

the angle-dependent superfluid-density, or superfluid stiffness tensor.¹¹⁴

As a final step, we write the change in the free energy in linear response as

$$\Omega = \Omega_{\text{bulk}} + \delta\Omega_{\text{kin}} + \delta\Omega_{\text{magn}}, \quad (37)$$

where

$$\Omega_{\text{bulk}} = \mathcal{A} N_F \left\langle |\Delta(\mathbf{p}_F)|^2 \ln \frac{T}{T_c} + \pi T \sum_n \frac{|\Delta(\mathbf{p}_F)|^4}{|\varepsilon_n| (\Lambda_n(\mathbf{p}_F) + \varepsilon_n)^2} \right\rangle_{\mathbf{p}_F}, \quad (38)$$

is the bulk superconducting free energy (see, e.g., Ref. 35), and \mathcal{A} is the area of the superconducting sample considered. The change in free energy in Eq. (37) due to superflow consists of a kinetic contribution,⁵⁸

$$\delta\Omega_{\text{kin}} = \frac{1}{2} \int d\mathbf{R} \langle \mathbf{p}_s^\top(\mathbf{R}) \bar{\rho}_s(\mathbf{p}_F) \mathbf{p}_s(\mathbf{R}) \rangle_{\mathbf{p}_F}, \quad (39)$$

and a magnetic contribution

$$\delta\Omega_{\text{magn}} = \frac{1}{8\pi} \int d\mathbf{R} |\nabla \times \mathbf{A}(\mathbf{R}) - \mathbf{B}_{\text{ext}}|^2, \quad (40)$$

where \mathbf{B}_{ext} is the applied external magnetic field.

1. The solenoid gauge

With the above introduction to superflow in a superconductor, we are ready to study a superconducting annulus with outer (inner) radius $\mathcal{R}_>$ ($\mathcal{R}_<$), as illustrated in Fig. 8(c). The area of the annulus is $\mathcal{A} = \pi(\mathcal{R}_>^2 - \mathcal{R}_<^2)$. Let us first consider the following text-book vector potential¹¹⁵ used to illustrate the Aharonov–Bohm effect,¹¹⁶

$$\mathbf{A}_{\text{ext}}(\mathbf{R}) = \frac{\Phi_{\text{ext}}}{2\pi R} \hat{\boldsymbol{\phi}}, \quad R \in [\mathcal{R}_<, \mathcal{R}_>], \quad (41)$$

where $\hat{\boldsymbol{\phi}}$ is the unit vector along the azimuthal direction and Φ_{ext} is an externally applied magnetic flux threading only the hole. We call this

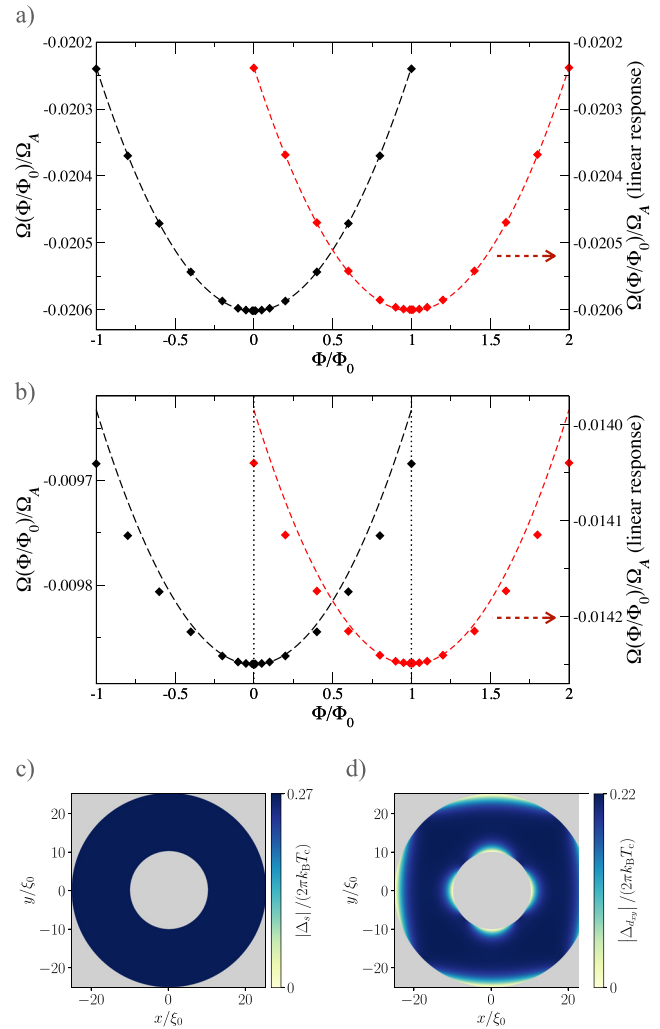


FIG. 8. The magnetic flux dependence of the free energy of an s-wave (a) and a d_{xy} -wave (b) superconducting annulus using the solenoid gauge Eq. (41). The outer (inner) radius of the annulus is $\mathcal{R}_> = 25\xi_0$ ($\mathcal{R}_< = 10\xi_0$), and the temperature is set to $T = 0.5T_c$. The order-parameter amplitude for the two cases are shown in (c), s-wave and (d), d -wave. In (a) and (b), the diamonds are results for the free energy extracted from SuperConga with the scale on the left side y-axis. The dashed lines are the free energy given by Eqs. (38) and (43) with scale on the right-side y-axis. The black (left) and red (right) parabolas correspond to enforced superconducting phase windings of $n = 0$ and $n = -1$.

particular gauge the solenoid gauge. It can be seen as a mathematical idealization to guarantee zero external magnetic flux through the superconducting annulus, while the vector potential remains finite everywhere. One possible physical realization of this gauge would be to let the superconducting annulus encircle a strong magnet.¹¹⁷

Using the gauge transformation Eq. (28), we can impose a phase winding in the order-parameter field going around the annulus. One of the constraints on an order parameter is that it is single valued. This condition leads to that χ in Eq. (28) must be chosen so that the total winding going around the void is a multiple n of 2π , and we can write

$\chi = n\varphi$, where n can take any integer value. If we now look at Eq. (2) we get, reinstating \hbar and c ,

$$\mathbf{p}_s = \frac{\hbar}{2} \nabla \chi - \frac{e \Phi_{\text{ext}}}{c 2\pi R} \hat{\boldsymbol{\phi}} = \frac{\hbar}{2R} \left(n + \frac{\Phi_{\text{ext}}}{\Phi_0} \right) \hat{\boldsymbol{\phi}}. \quad (42)$$

We see that \mathbf{p}_s can be made to vanish for certain values of the external flux, $\Phi_{\text{ext}} = -n\Phi_0$, i.e., the superfluid momenta vanishes throughout the superconducting annulus for every external field that matches integer multiples of the flux quantum Φ_0 . At those matching fields, $\delta\Omega_{\text{kin}} = 0$. As a consequence of the gauge properties, we see that if we have a self-consistent solution for the pair $\hat{g}_n, \hat{\Delta}_n$ at a given magnetic flux Φ_{ext} threading the hole, we can, via an appropriate gauge transformation \hat{U} , construct a solution at any other external flux that differ from Φ_{ext} by an integer number of flux quanta. This setup nicely demonstrates the fundamental concept of flux quantization.^{118–122} Note that the negative sign of the phase winding n reflects our choice of unit, $e = -|e|$, where the particle probability current flows with the gradient of the phase, while the associated charge current is in the opposite direction, see Eq. (35).

a. An s-wave superconducting annulus in the solenoid gauge. The simplest s-wave superconductor has an isotropic order parameter with $\Delta(\mathbf{p}_F) = \Delta$, where Δ is a complex scalar quantity and the basis function $\eta(\mathbf{p}_F) = 1$. For such an order parameter, scattering off sample surfaces leaves the superconducting state unaffected. As a consequence, linear-response theory will be sufficient to describe the annulus up to quite large superfluid momenta. We can, thus, calculate the kinetic contribution to the free energy Eq. (39), using the homogeneous solution of Eq. (20), to get

$$\frac{\delta\Omega_{\text{kin}}}{(2\pi T_c)^2 N_F} = \frac{\pi}{4} \varepsilon_0^2 Y_{3/2}(T) \ln \left(\frac{\mathcal{R}_>}{\mathcal{R}_<} \right) \left(n + \frac{\Phi_{\text{ext}}}{\Phi_0} \right)^2, \quad (43)$$

after integrating over the superconducting area and setting

$$Y_{3/2}(T) = \langle \hat{\mathbf{v}}_F(\mathbf{p}_F) Y_{3/2}(\mathbf{p}_F; T) \hat{\mathbf{v}}_F(\mathbf{p}_F) \cdot \hat{\mathbf{p}}_s \rangle_{\mathbf{p}_F}, \quad (44)$$

where $\hat{\mathbf{v}}_F(\mathbf{p}_F)$ and $\hat{\mathbf{p}}_s$ are unit vectors. For this particular gauge, we let the penetration depth $\lambda \rightarrow \infty$ and neglect screening effects. This leads to that the free-energy contribution $\delta\Omega_{\text{magn}}$ is zero. The free energy comes in a family of parabolas as function of the external flux. Each parabola is centered around $-n\Phi_0$ where n is the integer flux quantization, or phase winding, enforced on the order parameter. This is clearly shown in Fig. 8(a), where we also show results obtained by direct calculation using SuperConga, for phase windings $n=0$ and $n=-1$. The bottom of each parabola corresponds to the free energy $\Omega_{\text{bulk}}(T)$ at zero external flux for the considered temperature.

We finally mention that at temperatures close to the transition temperature, T_c , $\delta\Omega_{\text{kin}}$ can made to exceed the energy gain Ω_{bulk} when the external flux is close to $(n + \frac{1}{2})\Phi_0$. This is a manifestation of the Little–Parks effect,^{33,123–125} where one can induce a field-modulation of the critical temperature of a superconducting film enclosing a hole.

b. A d-wave superconducting annulus in the solenoid gauge. Next, we change the superconductor to be a d-wave superconductor. The order-parameter field for the d-wave depends on position on the

Fermi surface, \mathbf{p}_F , via the basis function, either the $d_{x^2-y^2}$ or d_{xy} as listed in Table III. This momentum dependence leads to that scattering off a surface may be pair breaking and one must solve for the spatial dependence of the order parameter using Eq. (20).

As for the s-wave superconductor, the free energy in an external magnetic field come in a family of parabolas $\sim (n + \Phi_{\text{ext}}/\Phi_0)^2$. However, as shown in Fig. 8(b), neither $\Omega_{\text{bulk}}(T)$ nor $\delta\Omega_{\text{kin}}(T, n, \Phi_{\text{ext}})$, are captured correctly by a calculation with a homogeneous order parameter magnitude and superflow treated in linear response. The order-parameter magnitude of a d_{xy} -wave superconducting annulus is shown in Fig. 8(d). As seen there are regions near the surfaces where the magnitude is severely reduced, which leads to correction to $\Omega_{\text{bulk}}(T)$.

The pair-breaking at interfaces that can be found in unconventional superconductors is due to low-energy Andreev states that are localized in a region of a few coherence lengths width at the surface.¹²⁶ These states modify low-temperature properties of the superconductor. A disk-shaped unconventional superconductor was studied by Suzuki and Asano finding a paramagnetic instability at low temperatures as a response to a small magnetic field.¹²⁷ We find similar paramagnetic behavior for the annular superconductor at low temperatures and relate the effects to spontaneous generation of phase gradients,^{57,107} in a state known as a phase crystal.^{58,61,128}

In Fig. 9, the free-energy is shown for a set of temperatures for the two cases where the initial guess for the order-parameter field is homogeneous, and where we lock a phase winding of 2π around the annulus. Below $T \sim 0.2T_c$, the existence of zero-energy Andreev states start to drive the superconductor to generate a finite \mathbf{p}_s by spontaneous phase gradients.^{57,107} This finite superfluid momentum gives rise to a Doppler shift, $\mathbf{v}_F \cdot \mathbf{p}_s$, that lifts the zero-energy states away from the Fermi level and, thus, lowers the free energy of the system. The free-energy minimum below $T^* \approx 0.18T_c$ is shifted to occur at finite magnetic flux. At the lowest temperature, we show here, $T = 0.1T_c$, we can find several metastable states at small fluxes. The orange rhombus in Fig. 9 has a purely real order parameter and the zero-energy states are not shifted away from the Fermi level. This state is only stable at $\Phi_{\text{ext}} \equiv 0$, and any small seed of a phase gradient in the order parameter, or a minute magnetic perturbation, will generate a configuration with spontaneous currents. The two low-energy states we find are combinations of two possible states as showed in Fig. 9.

2. Solving London-Maxwell equation in a symmetric gauge

In contrast to the case with the solenoid gauge above, a more common situation experimentally is when the external magnetic field $\mathbf{B}_{\text{ext}} = B_{\text{ext}} \hat{\mathbf{z}}$ is applied in a homogeneous fashion and penetrates also the superconductor. We incorporate this case through a symmetrical gauge $\mathbf{A}_{\text{ext}}(\mathbf{R}) = \frac{1}{2} B_{\text{ext}} (-y \hat{\mathbf{x}} + x \hat{\mathbf{y}})$ subject to the condition $\nabla \cdot \mathbf{A} = 0$.

Let us study the London–Maxwell equation for the vector potential \mathbf{A} in the symmetrical gauge,^{32,33,113,129,130}

$$-\nabla^2 \mathbf{A} = \frac{4\pi e}{c} \langle \bar{\rho}_s(\mathbf{p}_F) \mathbf{p}_s \rangle_{\mathbf{p}_F} = -\frac{1}{\lambda^2(T)} \left(\frac{\Phi_0}{2\pi R} n \hat{\boldsymbol{\phi}} + \mathbf{A} \right), \quad (45)$$

for the same annulus geometry as in Sec. V A 1. We allow for a phase winding $2\pi n$ of the order parameter going around the annulus and

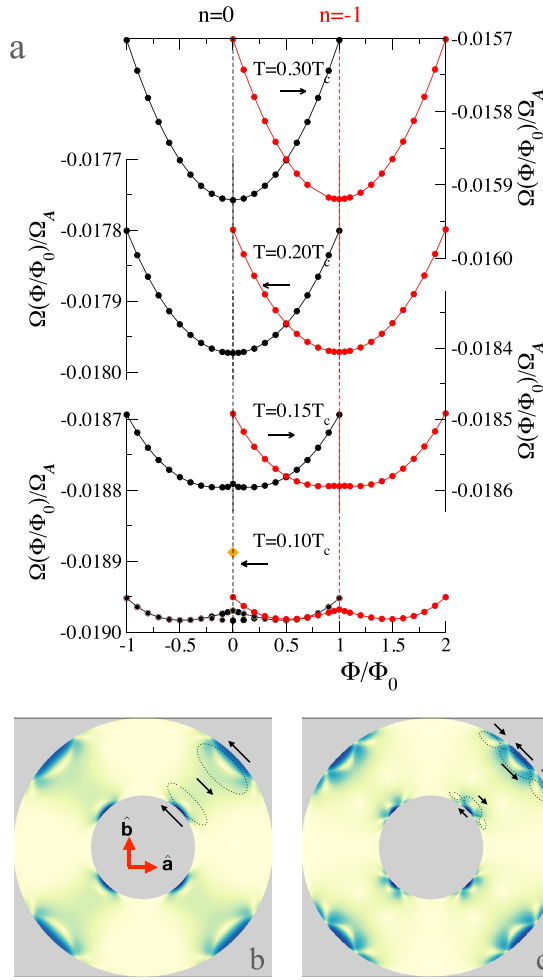


FIG. 9. (a) The free energy of a $d_{x^2-y^2}$ superconducting annulus as function of external flux using the solenoid gauge. The outer (inner) radius of the annulus is $\mathcal{R}_> = 25\xi_0$ ($\mathcal{R}_< = 10\xi_0$). The penetration depth is assumed to be much larger than the coherence length so that screening can be neglected. We look at two cases with and without a phase winding 2π around the annulus. In panel (a), from top to bottom, we lower the temperature from $T = 0.3T_c$ down to $T = 0.1T_c$. In panels (b) and (c), we show the configuration of spontaneous currents that signifies the two low-energy states at zero magnetic field. It is the current configuration showed in (c) that has the lowest free energy. The magnitudes of the current density are at maximum $\leq 0.1j_0$. The orange diamond in panel (a) is a special metastable state with no spontaneous currents.

include screening of the externally applied field. The temperature-dependent penetration depth $\lambda(T)$ entering Eq. (45) is defined as

$$\frac{1}{\lambda^2(T)} = \frac{1}{\lambda_0^2} Y_{3/2}(T). \quad (46)$$

As long as the inner radius is much larger than the coherence length and the external magnetic field is moderate in strength, the superfluid momentum will be small, and we can disregard any reduction of the order-parameter amplitude for an s -wave annulus. In this case, linear-response theory remains valid and gives a platform to verify more

involved calculations produced by SuperConga. Below, we introduce the dimensionless applied magnetic flux,

$$\alpha = \frac{\Phi_{\text{ext}}}{\Phi_0}. \quad (47)$$

a. Analytic solution of the London–Maxwell equation. Equation (45) is a diffusion equation and can be solved analytically. We give the solution here for completeness. In a cylindrically symmetric case, the solution to Eq. (45) is given by the modified Bessel functions of first and second kind, $I_\mu(x)$ and $K_\mu(x)$, respectively. Writing $\mathbf{A} = A(R)\hat{\phi}$, the general solution, in the external magnetic field $\mathbf{B}_{\text{ext}} = \alpha\Phi_0\hat{z}/\mathcal{A}$, may be written as

$$A(R) = \Phi_0\lambda \left[a I_1\left(\frac{R}{\lambda}\right) + b K_1\left(\frac{R}{\lambda}\right) \right] - \frac{n\Phi_0}{2\pi R}, \quad R \in [\mathcal{R}_<, \mathcal{R}_>], \quad (48)$$

$$A(R) = \beta \frac{\Phi_0 R}{\mathcal{A}_h 2}, \quad R \in [0, \mathcal{R}_<], \quad (49)$$

where $\mathcal{A} = \pi(\mathcal{R}_>^2 - \mathcal{R}_<^2)$ and $\mathcal{A}_h = \pi\mathcal{R}_<^2$ are the areas of the annulus and of the inner hole. The term $\sim 1/R$ is the particular solution needed when $n \neq 0$. The unknown constants (a, b, β) are determined so that \mathbf{A} is bounded for $R < \mathcal{R}_<$, and continuous at $R = \mathcal{R}_<$ and that the magnetic field $\mathbf{B} = \nabla \times \mathbf{A}$ is continuous at $R = \mathcal{R}_>$. We state them for completeness

$$a = \frac{1}{M} \left(\frac{K_2^<}{\mathcal{A}} \alpha + \frac{K_0^>}{\mathcal{A}_h} n \right), \quad (50)$$

$$b = \frac{1}{M} \left(\frac{I_2^<}{\mathcal{A}} \alpha + \frac{I_0^>}{\mathcal{A}_h} n \right), \quad (51)$$

$$\beta = \frac{1}{M} \left(\frac{2\pi\lambda^2}{\mathcal{A}} \alpha - [I_0^> K_0^< - I_0^< K_0^>] n \right), \quad (52)$$

with $M = I_0^> K_2^< - I_2^< K_0^>$ and $I_\mu^{>,<} = I_\mu\left(\frac{\mathcal{R}_{>,<}}{\lambda}\right)$ and the same for $K_\mu^{>,<}$. The corresponding magnetic field, $\mathbf{B} = B(R)\hat{z}$, in and around the annulus is

$$B(R) = \Phi_0 \left[a I_0\left(\frac{R}{\lambda}\right) - b K_0\left(\frac{R}{\lambda}\right) \right], \quad R \in [\mathcal{R}_<, \mathcal{R}_>], \quad (53)$$

$$B(R) = \beta \frac{\Phi_0}{\mathcal{A}_h}, \quad R \in [0, \mathcal{R}_<].$$

The constant β gives the flux strength in the inner hole in a similar way as α does for the annulus.

b. An s -wave superconducting annulus. For the s -wave annulus, Eqs. (49) and (53) give a good account for the response to an external field. The free energy of the superconducting annulus can, thus, be evaluated using Eqs. (39) and (40), and the results compare well with results obtained by SuperConga. By inspection, one sees that the functional form of the free-energy correction due to an external field (α) and a possible phase winding (n) will be¹³⁰

$$\delta\Omega_{\text{kin}} + \delta\Omega_{\text{magn}} = \delta\Omega_0(\omega_\lambda n^2 + (\omega_B n - \alpha)^2). \quad (54)$$

We determine the factors $\delta\Omega_0$, ω_λ , and ω_B numerically. They depend both on geometry and on temperature via the temperature dependence of the penetration depth.

The results for the free energy are shown in Fig. 10 for $T = 0.2T_c$ and for a set of different penetration depths. The main purpose here is to verify SuperConga by comparison to the results obtained by linear response. The agreement is excellent for small phase windings and large penetration depths. However, the agreement worsens with decreasing λ and increasing n . This is a numerical issue that is remedied by a finer spatial resolution of the simulation. This highlights that one needs verify results from SuperConga by varying the parameters that are related to numerical accuracy. These are typically spatial resolution of the geometry, angular resolution of the Fermi-surface averages, and the cutoff in the frequency sums.

c. A d-wave superconducting annulus. For the *d*-wave superconductor, we solve Eqs. (7), (8), and (10) self-consistently. We set the zero-temperature penetration depth to $50\xi_0$ as most *d*-wave superconductors are extreme type-II superconductors. The results are shown in Fig. 11 for zero and one phase winding locked into the annulus. Well above $T \approx 0.2T_c$ we find that the free energy has the parabolic functional dependence on applied field as for the *s*-wave superconductor. At lower temperatures, the low-energy surface Andreev states dominate, and their paramagnetic response modifies the free energy to host two minima at finite external flux, signaling that it is energetically favorable to form spontaneous currents as discussed above using the solenoid gauge.

d. Magnetic moment. The presence of a circulating current in an annulus can be detected as a magnetic moment,

$$\mathbf{m} = \frac{1}{2} \int_V d\mathbf{R} \mathbf{R} \times \mathbf{j}_{3D}(\mathbf{R}). \quad (55)$$

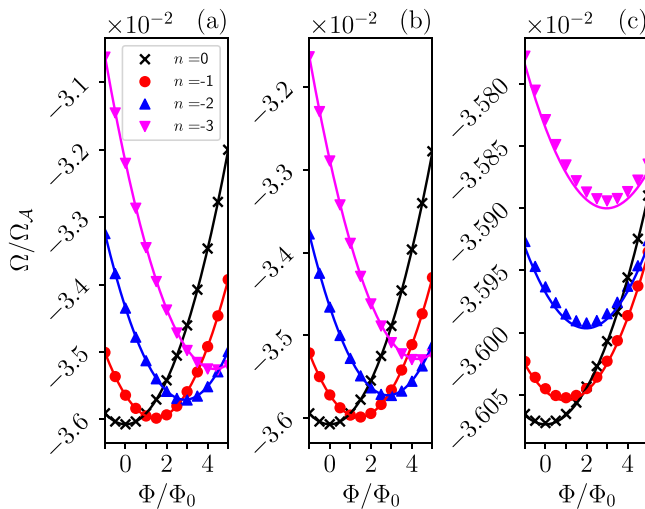


FIG. 10. The free energy for an *s*-wave superconductor in an annulus, with an outer (inner) radius $R_> = 25\xi_0$ ($R_< = 10\xi_0$), in an external magnetic field in symmetric gauge, at $T = 0.2T_c$. The magnetic flux-density is given by $B_z = \Phi_{\text{ext}}/\mathcal{A}$, where $\mathcal{A} = \pi(R_>^2 - R_<^2)$. Four different values of the order-parameter phase-winding, n , are shown in each plot. The points are numerical results from SuperConga, and the solid lines are analytical results. The only parameter that differs between the plots is the Ginzburg–Landau parameter $\kappa = \lambda/\xi_0$, which is (a) $\kappa = 100$, (b) $\kappa = 20$, and (c) $\kappa = 2$.

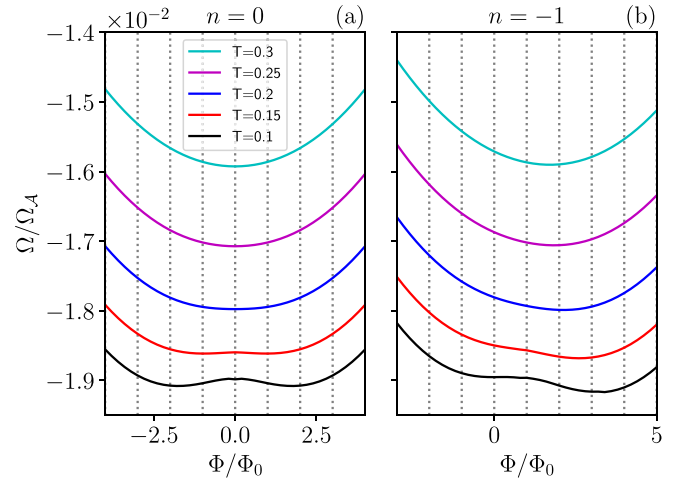


FIG. 11. The free energy as a function of the external magnetic field for a *d*-wave superconducting annulus, with an outer (inner) radius $R_> = 25\xi_0$ ($R_< = 10\xi_0$), obtained using SuperConga. The penetration depth is much larger than the coherence length, $\lambda = 50\xi_0$. We look at two cases; without ($n=0$) and with ($n=-1$) a phase winding of 2π around the annulus. From top to bottom, we lower the temperature from $T = 0.3T_c$ down to $T = 0.1T_c$ as in Fig. 9. The dotted vertical lines are a guide to the eye.

Since the supercurrent, $\mathbf{j}(\mathbf{R})$, we compute with SuperConga is confined to flow in a two-dimensional plane, the current density in a stack of layers is given as

$$\mathbf{j}_{3D}(\mathbf{R}) = d \sum_{\text{layers}} \delta(z - z_l) \mathbf{j}(\mathbf{R}),$$

i.e., the magnetic moment is given as

$$\mathbf{m} = \frac{N_l d}{2} \int_{\mathcal{A}} d\mathbf{R} \mathbf{R} \times \mathbf{j}(\mathbf{R}), \quad (56)$$

where $N_l d$ is the thickness of the grain along the z -axis assuming N_l layers separated by the c -axis distance d . The unit for the magnetic moment is $m_0 = j_0 \xi_0 \mathcal{V}$, where

$$j_0 \xi_0 = \frac{c^2 B_0}{4\pi \kappa^2} \text{ (cgs)} = \frac{1}{\mu_0} \frac{B_0}{\kappa^2} \text{ (SI)}, \quad (57)$$

with $B_0 = \Phi_0/(\pi \xi_0^2)$ and $\mathcal{V} = N_l d \mathcal{A}$. The parameters that determine m_0 are, thus, the geometrical size (and shape) of the object and the magnetic penetration depth specific of the superconducting material. M_0 can also be expressed in the perhaps more familiar form $m_0 = 2\mu_B n \mathcal{V}$, where $\mu_B = \hbar e/2m_e$ is the Bohr magneton and $n = v_F p_F N_F$ the number density of charge carriers. Inserting numerical values for Φ_0 and μ_0 and setting the length scale to nm :s for λ_0 and \mathcal{V} gives

$$\frac{m_0}{(\mathcal{V}/\lambda_0^2)} \approx 0.5 \times 10^{-18} \text{ Am}^2 = 0.5 \times 10^{-21} \text{ emu}. \quad (58)$$

Inserting the analytical form of the vector potential from Eq. (49) into Eq. (34), we can evaluate the magnetic field (α) and phase-winding (n) dependence of the magnetic moment of a stack of 2D *s*-wave annuli to be

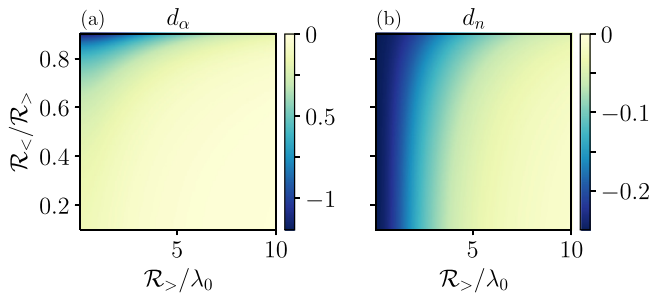


FIG. 12. (a) The dependence of d_α and (b) d_n on the relative sizes of the inner (outer) radius $R_<$ ($R_>$) and the penetration depth λ_0 . Here, d_α and d_n are factors entering the expression for the magnetic moment of an s-wave superconducting annulus, Eq. (59). Hence, together with this equation, the figures show how a larger magnetic moment, and therefore, experimental signature, can be obtained by creating annulus samples with different inner and outer radii.

$$\mathbf{m}(\alpha, n) = m_0(d_\alpha \alpha + d_n n) \hat{\mathbf{z}}, \quad (59)$$

where

$$d_\alpha = \frac{\pi \lambda_0^2}{\mathcal{A}} \frac{\mathcal{A} + \mathcal{A}_h}{M \mathcal{A}} (I_2^< K_2^> - I_2^> K_2^<), \quad (60)$$

$$d_n = \frac{\pi \lambda_0^2}{\mathcal{A}} \left(\frac{2\pi \lambda^2}{M \mathcal{A}_h} - 1 \right). \quad (61)$$

How d_α and d_n depend on the annulus radii and the penetration depth is shown in Fig. 12. In Fig. 13, we show how the magnetic moment depends on the penetration depth and the forced phase winding as function of applied field. The computed values of the magnetic moment using SuperConga fall right on the analytical ones of Eq. (59).

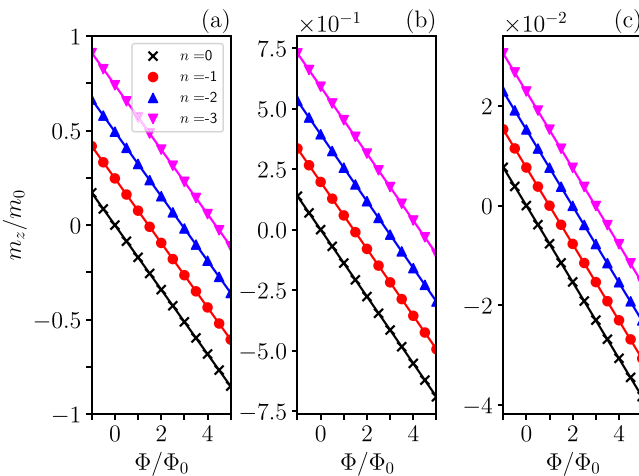


FIG. 13. The magnetic moment, Eq. (56), for an s-wave superconducting annulus, with an outer (inner) radius $R_> = 25\xi_0$ ($R_< = 10\xi_0$), in an external magnetic field in symmetric gauge, at $T = 0.2T_c$. The external flux density is given by $\mathbf{B}_{\text{ext}} = \Phi_{\text{ext}} \hat{\mathbf{z}} / \mathcal{A}$, where $\mathcal{A} = \pi(R_>^2 - R_<^2)$. Four different values of the order-parameter phase-winding, n , are shown in each plot. The points are numerical results from SuperConga, and the solid lines are analytical results, Eq. (59). The only parameter that differs between the plots is the Ginzburg–Landau parameter $\kappa = \lambda_0/\xi_0$, which is (a) $\kappa = 100$, (b) $\kappa = 20$, and (c) $\kappa = 2$.

While the magnetic moment of the s-wave annulus holds little surprise, the magnetic moment of the d-wave annulus shows trace of a low-temperature transition where screening currents are modified by the paramagnetic nature of the zero-energy Andreev states.^{57,107,131–133} The free energy dependence on temperature and magnetic field may be hard to directly measure whereas the magnetic moment of a d-wave annulus could be a possible route to detecting this low-temperature transition. In Fig. 14, we show the magnetic moment of a d-wave superconducting annulus as function of external flux at several temperatures. The dimensions are the same as for the s-wave case, i.e., $R_> = 25\xi_0$ and $R_< = 10\xi_0$, and the penetration depth is $\lambda_0 = 50\xi_0$. For low temperatures and small external fields, the signature of the Andreev states is that the magnetic moment of the annulus is reversed compared to that at high fields.

B. A mesoscopic s-wave superconducting disk

As another example, we consider an s-wave superconductor in the form of a 2D-disk, with radius R , in an external magnetic field. We will allow for arbitrary strength of the applied magnetic field so that multiple vortices can enter the disk. The general solution can and will break the continuous rotational symmetry of the disk around its central axis. Investigating this problem requires that we solve the Eilenberger equation, Eq. (10), with a self-consistently determined order-parameter field, Eq. (3), and vector potential Eq. (7). The magnetic field where the first vortex enters is the first critical field of a superconductor and is referred to as B_{c1} . At this field, it becomes energetically favorable for the superconducting material to host one vortex

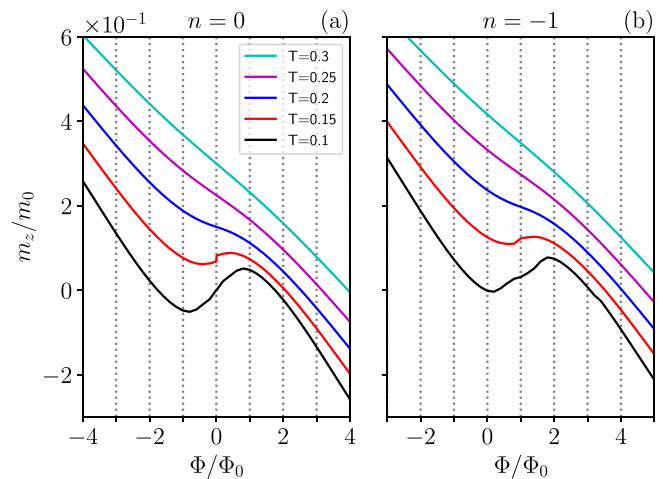


FIG. 14. The figure shows how the flux dependence of the magnetic moment $m_z(\Phi)$ in a d-wave superconducting annulus changes with temperature. The configuration is the same as in Fig. 11. The two panels differ by (a) having no order-parameter phase-winding ($n=0$) and (b) having one ($n=-1$). In both panels, the magnetic moment goes from being linear in applied flux at higher temperatures $T \geq 0.25T_c$ to highly non-linear and with a complete reversal in the sign of the slope close to $\Phi/\Phi_0 = n$ at low temperatures $T \leq 0.15T_c$. This temperature dependence of the magnetic moment is a direct consequence of surface-scattering induced Andreev states in a d-wave superconductor. The non-linear flux dependence at low temperatures is a signature that a “phase crystal” is established. The dotted vertical lines in both panels are a guide for the eye. Each line is shifted 0.075 units upward from the line below.

located in its interior well away from its physical edges. This compared to generating large screening currents flowing at the sample edges. The first critical field will depend on several parameters. Parameters intrinsic to the superconducting material are the penetration depth, $\lambda(T)$, and the coherence length, $\xi(T)$. Both depend on temperature via the temperature dependence of the superconducting energy gap $\Delta(T)$. Extrinsic parameters are, e.g., the physical size and shape of the superconducting sample, which in our example here will simply be the radius R of the disk. Also, temperature and magnetic-field strength are externally controlled. We will assume that our superconductor is a layered 2D-material with negligible inter-layer coupling. As the field is increased above B_{c1} , a cascade of mesoscopic vortex configurations are possible, which are separated by relatively small energy barriers. At high fields, a vortex lattice may be established.¹¹² Reaching the second critical field, B_{c2} , bulk superconductivity ultimately vanishes. With our framework SuperConga, we can treat the three relevant length scales of this problem, i.e., ξ_0 , λ_0 , and R , on the same footing at general temperatures and at general magnetic fields up to B_{c2} (and in principle B_{c3}). Throughout this section, we focus on a disk with a radius fixed to $25\xi_0$ at a temperature set to $0.2T_c$ if not otherwise stated. The analytic results that we give serve as an important tool to verify the results extracted from SuperConga.

Let us first focus on B_{c1} which was nicely treated by Fetter in Ref. 129 for a thin-film superconducting disk. In the case that the penetration depth is the largest scale in the system, the back-coupling of the current can be omitted. The corresponding superfluid momentum is

$$\mathbf{p}_s(R) = \frac{\hbar}{2} \left(\frac{n}{R} + \alpha \frac{R}{\mathcal{R}^2} \right) \hat{\phi}.$$

Calculating the free-energy density correction due to the superfluid momentum above with ($n = -1$) and without ($n = 0$), a single vortex gives

$$n = 0 : (\Omega(\alpha) - \Omega_{\text{bulk}})/\mathcal{A} = \delta\omega_{\text{kin}}\alpha^2, \quad (62)$$

$$n = -1 : (\Omega(\alpha) - \Omega_{\text{bulk}})/\mathcal{A} = \delta\omega_{\text{kin}}(\varepsilon_{\text{core}} - 4\alpha + \alpha^2). \quad (63)$$

where Ω_{bulk} is the bulk free-energy contribution Eq. (38) and

$$\delta\omega_{\text{kin}} = \frac{1}{16} \left(\frac{\xi_0}{R} \right)^2 Y_{3/2}(T), \quad (64)$$

gives the free-energy contribution from the induced supercurrent due to the combined effect of the external field and to the phase-winding of 2π around the vortex core. The term $\varepsilon_{\text{core}}$, in principle, consists of two terms. The first is proportional to $\ln(R/\mathcal{R}_c)$, with the vortex-core radius \mathcal{R}_c , due to the $1/R$ dependence of \mathbf{p}_s . The second, ε_0 , is due to the gradient-energy terms that come from the suppression of the order-parameter amplitude needed to accommodate the singularity in the vortex-core center. This second term is not captured in the simple London–Maxwell theory as it requires a theory that includes also the coherence-length scale ξ_0 in the calculations. In Fig. 15, we show calculations made with SuperConga for $\lambda_0 = \infty$. As seen, numerical results fall straight on the theoretical prediction for $(\Omega(\alpha) - \Omega_{\text{bulk}})/\mathcal{A}$ given by Eqs. (62) and (63). The main physical observable to extract is B_{c1} , defined as the field where having a vortex becomes energetically favorable compared to the vortex-free case. For this disk at this temperature, B_{c1} occurs at $\alpha \approx 3.8$.

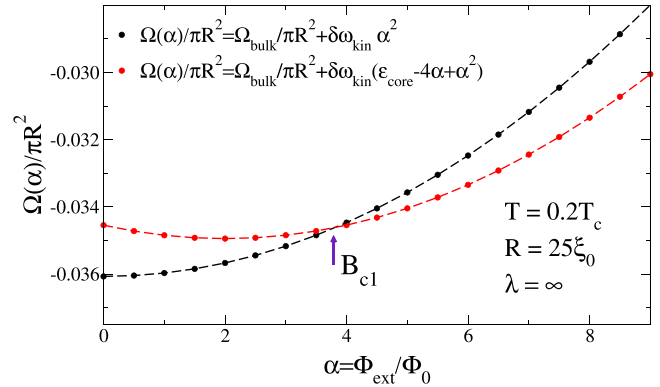


FIG. 15. The figure shows how the first critical field (B_{c1}) can be extracted from the intersection of the solutions for zero ($n=0$) and one ($n=1$) Abrikosov vortex. Here, the free energy per unit area is plotted as a function of externally applied field for an s-wave superconducting disk of radius $25\xi_0$, computed analytically (points) and numerically with SuperConga (dashed lines). Here, we show the low field free energy in the case $\lambda = \infty$ and at a temperature set to $0.2T_c$. The first critical field (B_{c1}) can be extracted from the intersection of the solutions for zero ($n=0$, black curves) and one ($n=1$, red curves) Abrikosov vortex.

When the penetration depth is comparable to or smaller than the size of the disk, we need to analyze Eqs. (49)–(53) in the limit that $\mathcal{R}_c \rightarrow 0$. The result is

$$A_\phi(R) = \lambda B_{\text{eff}}(\alpha, n) I_1 \left(\frac{R}{\lambda} \right) + n \frac{\Phi_0}{2\pi} \left[\frac{1}{R} - \frac{1}{\lambda} K_1 \left(\frac{R}{\lambda} \right) \right], \quad (65)$$

$$B_z(R) = B_{\text{eff}}(\alpha, n) I_0 \left(\frac{R}{\lambda} \right) + n \frac{\Phi_0}{2\pi\lambda^2} K_0 \left(\frac{R}{\lambda} \right), \quad (66)$$

with

$$B_{\text{eff}}(\alpha, n) = \frac{\Phi_0}{\pi R^2} \frac{1}{I_0 \left(\frac{R}{\lambda} \right)} \left[\alpha - \frac{n}{2} \left(\frac{R}{\lambda} \right)^2 K_0 \left(\frac{R}{\lambda} \right) \right]. \quad (67)$$

The functional form of Eqs. (65) and (66) allows an explicit evaluation of the free-energy contributions $\delta\Omega_{\text{kin}}$ and $\delta\Omega_{\text{magn}}$ in the case of no vorticity ($n=0$) and $\delta\omega_{\text{kin}}$ is modified for finite λ in the following:

$$\delta\omega_{\text{kin}}(\lambda) = \left(k \left(\frac{R}{\lambda} \right) + b \left(\frac{R}{\lambda} \right) \right) \left(\frac{\xi_0}{R} \right)^2 Y_{3/2}(T), \quad (68)$$

with

$$k(x) = \frac{1}{x^2} \frac{I_1^2(x) + \frac{2}{x} I_0(x) I_1(x) - I_0^2(x)}{2I_0^2(x)}, \quad (69)$$

$$b(x) = \frac{2I_0^2(x) - \frac{4}{x} I_0(x) I_1(x) - I_1^2(x)}{2I_0^2(x)}. \quad (70)$$

In the limit that $x \rightarrow 0$, corresponding to $\lambda \rightarrow \infty$, $k(x) \rightarrow 1/16$, and $b(x) \rightarrow 0$, and we recover the prefactor $\delta\omega_{\text{kin}}$ [Eq. (64)] as we should.

The vector potential, Eq. (65), is finite in the origin. This is not the case for the magnetic field, Eq. (66), which diverges as $\ln(R)$ when $R \rightarrow 0$ in the origin.^{32,33} This divergence leads to that

superconductivity must vanish locally in the vortex core (see, e.g., Fig. 4). The suppression of superconductivity happens over a couple of coherence lengths, and we determine the order-parameter profile using SuperConga. The functional form of the free energy, Eqs. (62) and (63), holds also for finite screening but with $\delta\omega_{\text{kin}}(\lambda)$ and with a modified coefficient for linear-in-field term in Eq. (63) [see also Eq. (54)]. In Fig. 16, we show how the free-energy of a single vortex core, i.e., $\delta\omega_{\text{kin}}(\lambda)\epsilon_{\text{core}}$, depends on temperature. The penetration depth is considered for the values $\kappa = (2, 20, \infty)$, and the external field is zero. The temperature dependence of ξ and λ gives that close to T_c , in the regime of validity of Ginzburg–Landau theory, the physical size of the disk will be the smallest length-scale of the system. In this limit, the relative contribution of the vortex core to the free energy dominates, as seen in Fig. 16. Moving to lower temperatures Ginzburg–Landau theory is strictly speaking no longer applicable and we need to resort to the quasiclassical theory. Below $T = 0.8T_c$, the vortex-core contribution for this geometry becomes less dominant and settles to be less than 5% of the total free energy. Instead, the free energy mainly depends on how well the supercurrents originating from the phase-winding around the vortex core are screened, i.e., focus is primarily on the ratio λ_0/\mathcal{R} . We consider two cases of screening: a typical type-II superconductor with $\kappa = 20.0$ for which the penetration depth compare to the size of the disk, and a marginal type II superconductor with $\kappa = 2.0$ with $\lambda_0 \ll \mathcal{R}$. In Fig. 16, we see that strong screening ($\kappa = 2$) reduces the energy cost of a single vortex compared to the case of moderate screening ($\kappa = 20$) so that we can extract $\delta\omega_{\text{kin}}\epsilon_{\text{core}}(\kappa = 2) \approx 0.5\delta\omega_{\text{kin}}\epsilon_{\text{core}}(\kappa = 20)$.

In Fig. 17, we show a large set of computed free energies as a function of applied external magnetic field. The magnetic field $B = \alpha\Phi_0/\mathcal{A}$ ranges from $\alpha = 0$ to 200 so that we pass B_{c1} , where the first vortex enters, to B_{c2} where the superconducting state is no longer stable and the disk becomes normal. Starting with the first critical field, B_{c1} , we lift results from the literature.^{113,129} For a superconducting disk of radius \mathcal{R} where $\lambda_0/\mathcal{R} \sim 1$, the first critical field is given as

$$B_{c1}^{\text{disc}} = \frac{\Phi_0}{\mathcal{A}} \left(\ln \frac{\mathcal{R}}{\mathcal{R}_c} + \epsilon_0 \right), \quad (71)$$

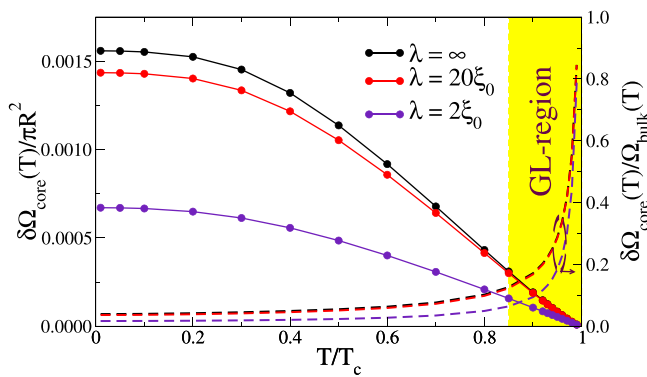


FIG. 16. The vortex-core energy $\delta\Omega_{\text{core}} = \delta\omega_{\text{kin}}\epsilon_{\text{core}}$, extracted as the difference in free energy between the one-vortex and the zero-vortex case, at zero-field ($\alpha = 0$). Here, $\delta\Omega_{\text{core}}(T)$ is computed with SuperConga and shown for the full temperature range from $T \approx 0.0$ to $T \leq T_c$. Solid and dashed lines correspond to the left and right axes, respectively, and line colors denote the penetration depth as indicated by the label.

with ϵ_0 due to the gradient terms. For a *pancake vortex* in the extreme 2D-case with $\lambda_0/\mathcal{R} \ll 1$, the first critical field for a vortex in the 2D superconducting sheet is given as^{60,134}

$$B_{c1}^{\text{sheet}} = \frac{\Phi_0}{4\pi\lambda^2} (\ln \kappa + \epsilon_0). \quad (72)$$

The lower critical field, B_{c1} , that we extract correspond well to earlier theory if we associate the case $\kappa = 20$ to Eq. (71) and $\kappa = 2$ to Eq. (72). Extracting from the data in Fig. 17, we find

$$B_{c1}(\kappa = 20) = 4.4 \frac{\Phi_0}{\mathcal{A}}, \quad (73)$$

$$B_{c1}(\kappa = 2) = 63.3 \frac{\Phi_0}{\mathcal{A}} = 1.62 \frac{\Phi_0}{4\pi\lambda_0^2}. \quad (74)$$

The numerical prefactor $4.4 = \ln(\mathcal{R}/\mathcal{R}_c) + \epsilon_0$ contains two unknown parameters \mathcal{R}_c and ϵ_0 for $\kappa = 20$. On the other hand the prefactor in the case $\kappa = 2$, $1.62 = \ln \kappa + \epsilon_0$ contains only one unknown and we extract $\epsilon_0 = 1.62 - \ln \kappa \approx 0.93$. Taking the results from Fig. 16, we can determine the nominal vortex-core size \mathcal{R}_c as $\ln \mathcal{R}_c = \ln \mathcal{R} + 2\epsilon_0(\kappa = 2) - 4.4 \approx 0.68$ or $\mathcal{R}_c = 2.7\xi_0$. This value is quite reasonable estimate for a vortex-core size if we compare with the order-parameter field displayed in Fig. 4 [or in the panels (b)–(e) in Fig. 17].

For large external fields, the upper critical field, B_{c2} , is given by the field scale when vortex cores start to overlap,³³ i.e.,

$$B_{c2} = \eta \frac{\Phi_0}{2\pi\xi_0^2}. \quad (75)$$

We add a prefactor, η , as in our finite geometry, there is always a vortex-free surface layer around the perimeter of the sample that reduces the effective vortex-lattice area. The vortex-free ring at the perimeter of our disk has a width $\sim 5\xi_0$ and, thus, we can estimate $\eta = (20/25)^2 = 0.64$. The upper critical field for our disk would be at $\alpha = \frac{1}{2}(\mathcal{R}/\xi_0)^2 = 312.5$ but we find that superconductivity vanishes at $\alpha \approx 200$. This reduction of B_{c2} captured (by construction) by the smaller effective area of the vortex as we described above.

For intermediate fields, $B_{c1} < B < B_{c2}$, a vortex lattice emerges. In Fig. 17, we follow how the vortex lattice is formed for the two cases, $\kappa = 2, 20$. It is a very rich system with several local free-energy minima corresponding to different metastable configurations of vortices in the disk. For a stronger type-II superconductor, $\kappa = 20$, the lattice structure is less pronounced and only at high fields can we see a square lattice evolving in the center of the disk. On the route toward higher fields, one can find transitions between states comprising of distinct number of vortices. In the following example below, we come back to this and connect to recent experiments.

For the marginal type-II superconductor, the vortex lattice is established quite rapidly in increasing field above B_{c1} . The vortex lattice is triangular with the outer most layer of vortices creating an ever denser necklace of vortices following the perimeter.

C. Spectroscopy on superconducting nanostructures

As we have seen above, the vortex physics in a finite geometry might be highly modified compared to a bulk system. In particular, the number of vortices and their separation do not only depend on the

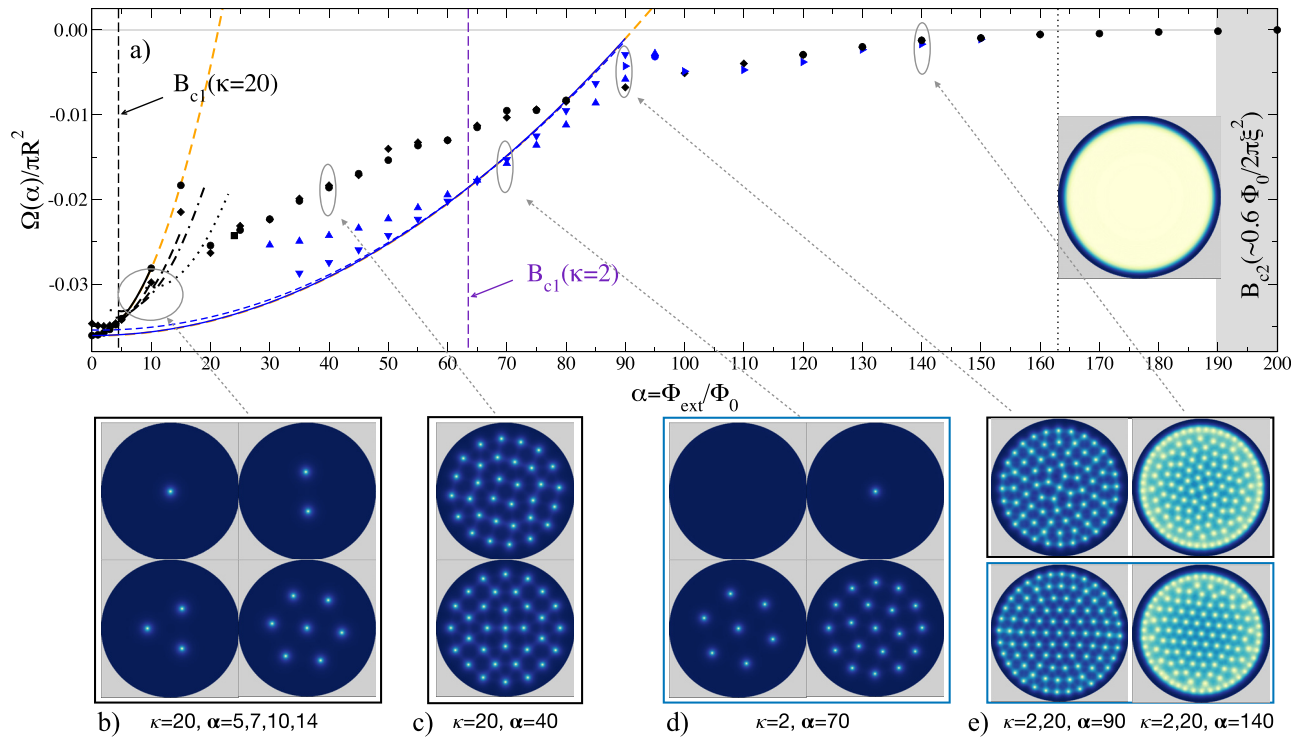


FIG. 17. (a) The free energy as a function of externally applied field for an s-wave superconducting disk of radius $25\xi_0$. The temperature is set to $0.2T_c$. We consider two values for the penetration depth $\lambda_0 = 20\xi_0 \sim \mathcal{R}$ and $\lambda_0 = 2\xi_0 \ll \mathcal{R}$ and follow the route from the zero-field case via the first critical field B_{c1} , where the first vortex enters the disk, to the critical field B_{c2} where the (bulk) superconductivity is killed by the magnetic field. Black lines and symbols refer to $\kappa = 20$, while the same in indigo refers to $\kappa = 2$. The two orange dashed curves are given using the analytical expression Eq. (68) for the two respective values of κ . The lower panels (b)–(e) show possible vortex configurations at different fields. In panel (b), we show the four vortex states that live on the parabolas at low fields. As the field is increased above $B_{c1}(\kappa = 20)$, the stable configuration contains more and more vortices. Panel (c) shows that at higher fields, different vortex configurations are possible at the same field with very little difference in free energy between the configurations. In panel (d), we show four vortex configurations just above $B_{c1}(\kappa = 2)$. This indicates that for marginal type-II superconductors, the vortex lattice is established just above B_{c1} . In panel (e), we show configurations for both values of κ . The inset in panel (a) shows the extreme high-field configuration with only a surface layer having a sizable order-parameter amplitude. It sustains a large circulating supercurrent by a phase that winds rapidly around the disk perimeter. This surface superconductivity vanishes at an even higher field B_{c3} .

external field strength, but also on the geometry, the nature of boundaries, and the relation between system size and superconducting length scales, i.e., the coherence length and the penetration depth. This problem has attracted considerable interest both theoretically and experimentally during the last few years.^{3,19–21,55,88–96}

Figure 18 shows the arrangement of 1–6 vortices in a finite and irregular-shaped superconducting grain, inspired by the experiment of Timmermans *et al.*⁵⁵ Finding the arrangement and number of vortices for a given magnetic field is a difficult task due to the complicated magnetic field dependence of the free energy. For this reason, Fig. 18(a) shows the local density of states at the minimum of the free energy. The magnetic field dependence of the free energy is shown in Fig. 18(b) for every number of a vortex. Each vortex follows a parabolic dependence on the magnetic field with a minimum marked as a triangle. The theoretically obtained vortex positions are in qualitative agreement with the experiment results in Ref. 55.

The transition between different vortices entering the grain can be visualized by measuring the zero-bias conductance close to the edge of the grain [Fig. 2(b) in Ref. 55]. Increasing the magnetic field starting from zero, screening currents prevent the magnetic field to penetrate

the superconductor. The screening currents are responsible for a reduction of the superconducting gap and an increase in the density of states. The magnetic-field dependence of the density of states close to the edge is shown in Fig. 18(c) for each number of a vortex. The entrance of a vortex in the grain is accompanied by a reduction of the density of states at zero energy. These abrupt reduction of the local density of states are related to the reduction in the zero-bias conductance measurements in Ref. 55.

VI. SUMMARY

SuperConga is an open-source framework for simulating mesoscopic superconducting grains within the quasiclassical theory of superconductivity. Its main strengths are ease-of-use, speed, and modularity. A Python frontend enables the user to quickly set up simulations of multi-component singlet superconducting grains in general 2D-geometries and in an applied magnetic field. The framework solves for the order parameters self-consistently and includes the back-coupling of the vector potential due to induced supercurrents. Real-time visualization is provided during simulations, and the Python

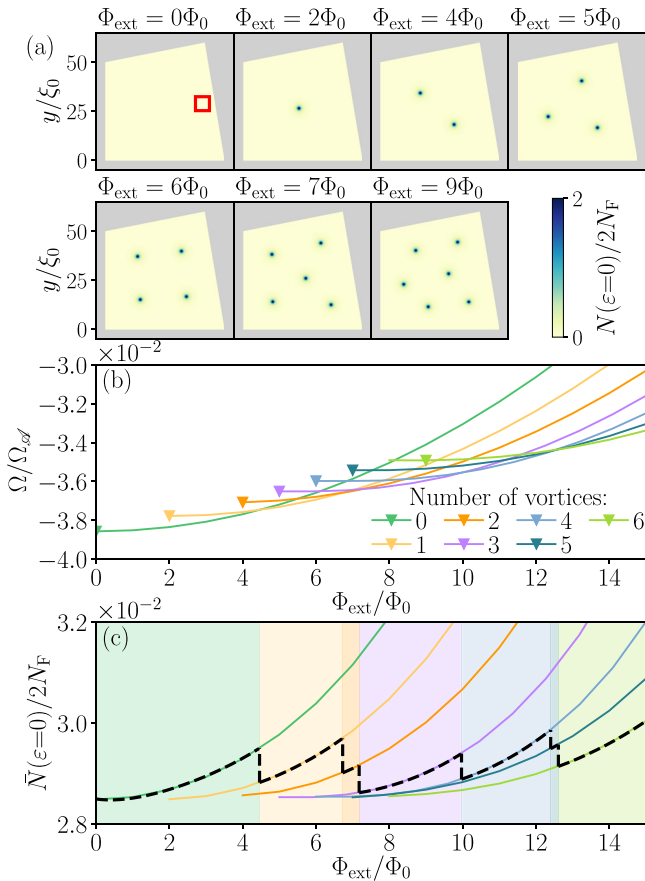


FIG. 18. Simulations of conventional superconductors shaped like an irregular square, inspired by experiments in Ref. 55, with similar values of the side length $S \sim 50\xi_0$, Ginzburg–Landau coefficient $\kappa = 83$, and temperature $T = 0.1T_c$. (a) Zero-energy local density of states at the minimum of the free energy with the specific number of vortices shown. (b) Magnetic field dependence of the free energy with the minimal value for each number of vortex marked with a triangle. (c) Average density of states as function of magnetic field. The average is over the surface region marked as red square in the first panel of (a). The smearing is $\delta = 0.008 \cdot 2\pi k_B T_c$.

frontend includes functionality for additional data analysis and visualization, as well as interactive spectroscopy of the local density of states.

The SuperConga framework is free to download under the GNU LGPL v3 license or higher, from its GitLab repository,⁸³ <https://gitlab.com/superconga/superconga>. An extensive user manual has been published online,⁸⁰ containing numerous pedagogical examples, tutorials, and guides. The framework has been developed and tested for Unix-based environments and generally runs on most modern laptops, desktops, as well as in cluster environments. The framework relies on the high-performance capabilities offered by GPU acceleration and CUDA^{78,79} and, therefore, requires a CUDA-capable device (i.e., NVIDIA GPU)¹³⁵ to run. To the best of our knowledge, SuperConga is the only open-source code that uses quasiclassical theory to describe mesoscopic superconducting grains and ballistic devices. It therefore adds the missing “mesoscopic link” between more phenomenological methods, such as London–Maxwell¹³⁶ and

Ginzburg–Landau,^{117,137,138} and fully microscopic methods, such as density-functional theory^{139–143} and tight-binding approaches.^{144–147}

This paper outlines the functionality of version 1.0 of SuperConga. We foresee that the project website will evolve continuously and that a community of developers can be formed to enable further improvements that increase the functionality and scope of the framework to include, e.g., impurity scattering, spin degrees of freedom, multiband physics, more general boundary conditions, and non-equilibrium phenomena.

ACKNOWLEDGMENTS

We thank Anton B. Vorontsov and Kevin Marc Seja for valuable discussions. We thank the Swedish Research Council (VR) and the European Research Council (ERC) for financial support. The computations were enabled by resources provided by the National Academic Infrastructure for Supercomputing in Sweden (NAISS) at the computing centers C3SE, HPC2N, and NSC, funded by the Swedish Research Council through Grant Agreement No. 2022-06725.

AUTHOR DECLARATIONS

Conflict of Interest

The authors have no conflicts to disclose.

Author Contributions

Patric Holmvall and Niclas Wall Wennerdal contributed equally to this work.

Patric Holmvall: Conceptualization (equal); Formal analysis (equal); Investigation (equal); Methodology (equal); Software (equal); Validation (equal); Visualization (equal); Writing – original draft (equal); Writing – review & editing (equal). **Niclas Wall Wennerdal:** Conceptualization (equal); Formal analysis (equal); Investigation (equal); Project administration (equal); Software (lead); Validation (equal); Visualization (equal); Writing – original draft (equal); Writing – review & editing (equal). **Mikael Håkansson:** Conceptualization (equal); Methodology (equal); Software (lead); Visualization (equal). **Pascal Stadler:** Investigation (equal); Software (equal); Validation (equal); Visualization (equal). **Oleksii Shevtsov:** Conceptualization (equal); Formal analysis (supporting); Methodology (supporting); Software (supporting). **Tomas Löfwander:** Conceptualization (equal); Formal analysis (equal); Investigation (equal); Software (equal); Supervision (equal); Validation (equal); Writing – original draft (equal); Writing – review & editing (equal). **Mikael Fogelström:** Conceptualization (equal); Formal analysis (equal); Funding acquisition (equal); Methodology (equal); Software (equal); Supervision (equal); Validation (equal); Writing – original draft (equal); Writing – review & editing (equal).

DATA AVAILABILITY

The data that supports the findings of this study are available within the article and its supplementary material.

APPENDIX A: DEPENDENCIES

This appendix lists external dependencies and packages that SuperConga offloads functionality to. For a more detailed

description, refer to the installation guide in the main README of the repository.⁸³ SuperConga uses git as a version control system, with the central repository being hosted on GitLab.⁸³ The SuperConga backend is written in C++, with most of the heavy lifting being done by CUDA^{78,79} kernels and Thrust^{148–150} algorithms running on the GPU. This functionality is provided by the CUDA toolkit, which also provides the NVCC compiler for compiling the CUDA code. Alternatively, Clang can be used for compilation. To run the compiled code, a CUDA-enabled device (an NVIDIA GPU) is required. Note that we use CMake¹⁵¹ as the build-system generator, and Ninja¹⁵² as the build system, but that it is possible to replace either of these with another choice. Linear algebra on the CPU uses Armadillo,^{153,154} and all tests are implemented in the doctest¹⁵⁵ framework. The real-time plotting is done using OpenGL^{156,157} and ArrayFire-Forge.¹⁵⁸ The package JsonCpp¹⁵⁹ is used to handle JSON¹⁶⁰ objects on the backend side, and HighFive¹⁶¹ for HDF5¹⁶² support. The SuperConga frontend requires a Python3 environment, in particular with the modules NumPy,¹⁶³ Matplotlib,¹⁶⁴ h5py,¹⁶⁵ and pandas.¹⁶⁶

APPENDIX B: THE SIMULATION-PARAMETER FILE

The simulation configuration is specified in a `simulation_config.json` file. It is a JSON¹⁶⁰ file, which consists of a number of key-value pairs on the form “key”: value. The value of a key can be more key-value pairs, numbers, strings, etc. Below is the JSON file with the parameters used in the demonstration example in Sec. IV.

LISTING 10: examples/swave_disc_vortex/simulation_config.json.

```
{
  "physics": {
    "temperature": 0.5,
    "external_flux_quanta": 1.5,
    "penetration_depth": 5.0,
    "crystal_axes_rotation": 0.0,
    "gauge": "symmetric",
    "charge_sign": -1,
    "order_parameter": {
      "s": {
        "critical_temperature": 1.0,
        "initial_phase_shift": 0.0,
        "initial_noise_stddev": 0.0,
        "vortices": [
          {
            "center_x": 0.0,
            "center_y": 0.0,
            "winding_number": -1.0
          }
        ]
      }
    }
  },
}
```

```
  "geometry": [
    {
      "disc": {
        "add": true,
        "center_x": 0.0,
        "center_y": 0.0,
        "radius": 15.0
      }
    }
  ],
  "numerics": {
    "convergence_criterion": 1e-05,
    "energy_cutoff": 16.0,
    "norm": "l2",
    "num_fermi_momenta": 32,
    "num_iterations_burnin": 0,
    "num_iterations_max": 10000,
    "num_iterations_min": 0,
    "points_per_coherence_length": 20.0,
    "vector_potential_error": 1e-06
  },
  "accelerator": {
    "name": "congacc"
  },
  "misc": {
    "data_format": "h5",
    "load_path": "",
    "save_frequency": -1,
    "save_path": "data/examples/swave_disc_vortex",
    "verbose": true,
    "visualize": true
  }
}
```

Under `physics` are all parameters directly related to the physics of the simulation, except the geometry which is specified separately. The temperature is given in units of the highest critical temperature of the order-parameter components, $T_c \equiv \max\{T_{c,\Gamma}\}$. The `external_flux_quanta` is given in units of Φ_0 . The `penetration_depth` is given in units of ξ_0 . If the penetration depth is zero or negative, it is considered to be infinite and the induced vector potential will be zero. The `crystal_axes_rotation`, i.e., the rotation of the crystal *ab*-axes relative to the simulation coordinate system, is given in units of a full turn, i.e., 2π . The choice for the gauge is either `symmetric`, `landau` or `solenoid`. The sign of the carrier charge is controlled by `charge_sign`.

The field `order_parameter` is a list of components. The supported order-parameter components are $d_{x^2-y^2}$, d_{xy} , $g_{xy}(x^2-y^2)$, and *s*. The `critical_temperature` is only meaningful with multiple components as it is the *relative* $T_{c,\Gamma}$ that is relevant. Each component is initialized as

$$\Delta_{\Gamma}(\mathbf{R}) = \Delta_{\Gamma}^{(\text{bulk})} e^{i\chi_{\Gamma}(\mathbf{R})} + Z, \quad (\text{B1})$$

$$\chi_{\Gamma}(\mathbf{R}) = 2\pi\delta_{\Gamma} + \sum_v w_{\Gamma}^{(v)} \varepsilon(\mathbf{R} - \mathbf{C}_{\Gamma}^{(v)}), \quad (\text{B2})$$

$$Z \sim \mathcal{N}(0, \sigma_{\Gamma}^2), \quad (\text{B3})$$

where δ_{Γ} is controlled by `initial_phase_shift`, and Z is complex additive Gaussian noise with a standard deviation σ_{Γ} controlled by `initial_noise_stddev` in units of $2\pi k_B T_c$. The field `vortices` is list of vortices, with each vortex being specified by its center position, $\mathbf{C}_{\Gamma} = (\text{center_x}, \text{center_y})$, and phase winding $w_{\Gamma} = \text{phase_winding}$. Furthermore, $\varepsilon(\mathbf{R}) = \text{atan2}(y, x)$ is the polar angle of the coordinate \mathbf{R} . We may add additional components by adding entries to the “`order_parameter`” field in the above JSON file. For example, we may add a subdominant d_{xy} -wave component initialized with a phase shift of $\pi/2$, some Gaussian noise, and no vortices.

LISTING 11: Example of a multi-component order parameter.

```

“order_parameter”: {
  “s”: {
    “critical_temperature”: 1.0,
    “initial_phase_shift”: 0.0,
    “initial_noise_stddev”: 0.0,
    “vortices”: [
      {
        “center_x”: 0.0,
        “center_y”: 0.0,
        “winding_number”: -1.0
      }
    ]
  },
  “dxy”: {
    “critical_temperature”: 0.1,
    “initial_phase_shift”: 0.25,
    “initial_noise_stddev”: 0.01,
    “vortices”: [
      {
        “center_x”: 0.0,
        “center_y”: 0.0,
        “winding_number”: -1.0
      }
    ]
  }
}

```

The geometry is built of three different kinds of components; `disc`, `regular_polygon`, and `polygon`. A disk is specified by its center which is set by `center_x` and `center_y`, and its radius. All parameters are in units of ξ_0 . A regular polygon is specified by its center coordinates `center_x` and `center_y`, the number of edges `num_edges`, its rotation, and the `side_length`. The side length and center is given in units of ξ_0 , and the rotation in units of a full turn, i.e., 2π . A

polygon is specified by the x - and y -coordinates of its vertices, `vertices_x` and `vertices_y` respectively. The vertices should be given in counterclockwise order. All components have a parameter `add`, which determines if the area of the component should be added (`true`), or subtracted (`false`), from the geometry. For more general geometries it is possible to add and subtract several objects by adding to the `geometry` field. For example, we may put a square hole of side length $5\xi_0$ in the disk at $10\xi_0$ to the right of the center:

LISTING 12: Example of a compound geometry.

```

“geometry”: [
  {
    “disc”: {
      “add”: true,
      “center_x”: 0.0,
      “center_y”: 0.0,
      “radius”: 15.0
    }
  },
  {
    “regular_polygon”: {
      “add”: false,
      “center_x”: 10.0,
      “center_y”: 0.0,
      “num_edges”: 4,
      “rotation”: 0,
      “side_length”: 5.0
    }
  }
]

```

Note that the order of the components is important if any component is subtracted.

Under `numerics` are parameters controlling the numerics of the simulation. The `convergence_criterion` determines when the simulation is considered converged. When all computed quantities; each order-parameter component, the charge current density, the induced vector potential, and the free energy, has a residual [as given by Eq. (27)] smaller than `convergence_criterion` the simulation should stop. The `energy_cutoff` is the largest energy, in units of $2\pi k_B T_c$, to include in the energy sums. The p -norm to use when computing the residuals is controlled by `norm`, with $p \in \{1, 2, \infty\}$. The `num_fermi_momenta` is the number of discrete momenta to use when approximating the Fermi-surface averages $\langle \cdots \rangle_{p_F}$. Note that the terms in the Fermi-surface averages are computed sequentially in contrast to the terms in the energy sum which are computed in parallel on the GPU.

The field `num_iterations_burnin` controls the number of iterations during which the order-parameter and vector potential are kept constant, while the coherence functions are converged and updated self-consistently, in particular on the boundary, i.e., $\gamma_{\partial D}$

and $\tilde{\gamma}_{\partial D}$. Once the burn-in period is over, all quantities are updated self-consistently again. This is particularly useful when resuming a previous simulation from file, since the order parameter and vector potentials are stored, but the coherence functions are not (due to occupying too much disk space). If the number of burn-in iterations is negative, it will run until convergence. The maximum and minimum of iterations to run is controlled by `num_iterations_max` and `num_iterations_min`, respectively. When `num_iterations_max` iterations has been run the simulation stops regardless of whether convergence has been achieved. Similarly, the simulation does *not* stop until `num_iterations_min` iterations has been run even if the simulation is considered converged. This is useful for avoiding endless simulations, or prematurely stopping a simulation that is moving slowly on a plateau, respectively. The spatial resolution of the simulation is set by `points_per_coherence_length`. The `vector_potential_error` is explained in [Appendix D](#).

All parameters of the accelerator are explained in [Appendix E](#).

Finally, under `misc` are miscellaneous parameters not affecting the simulation *per se*. `data_format` specifies which data format to use when saving files. The options are `h5` and `csv` yielding compressed HDF5¹⁶² files or plain-text CSV files, respectively. Both data formats are standard and have excellent support in, e.g., MATLAB¹⁶⁷ and Python (via `h5py`¹⁶⁵ and `pandas`¹⁶⁶). By setting `load_path` SuperConga will read the files located there and use them as the initial values for the order parameter and the vector potential. How often to save the results are controlled by `save_frequency`. If `save_frequency` is negative the results will only be saved at the end of the simulation, and if it is zero the results will not be saved at all, otherwise it will save every n th iteration. The results are saved to `save_path`. If `verbose` is true the progress of the simulation will be written to the terminal, and if `visualize` is true the simulation will be visualized live.

APPENDIX C: SOLVING THE RICCATI EQUATIONS

Using the trajectory coordinate, s , Eq. (25) in standard form is

$$\frac{\partial \gamma}{\partial s} = f[s, \gamma(s)], \quad (C1)$$

$$f[s, \gamma(s)] = \frac{i}{v_F} [\Delta^*(s)\gamma^2(s) + 2z(s)\gamma(s) + \Delta(s)], \quad (C2)$$

with the dependence on p_F and ε dropped, and

$$z(s) \equiv \varepsilon + \frac{e}{c} \mathbf{v}_F \cdot \mathbf{A}(s), \quad (C3)$$

for brevity. Instead of using the stepping method,^{70,168} i.e., using an analytical formula for $\gamma(s)$ but approximating $\Delta(s)$ and $z(s)$ as piecewise constant, we solve Eq. (C1) along a trajectory using the implicit mid-point method, which we find to be both faster and more accurate. The implicit mid-point method has the following update rule:

$$\gamma_{j+1} = \gamma_j + hf \left[s_{j+\frac{1}{2}}, \frac{1}{2}(\gamma_j + \gamma_{j+1}) \right], \quad (C4)$$

with the discretized trajectory coordinate $s_j = jh$, where j is an integer and h is the length between points. This yields a quadratic equation in γ_{j+1} to solve for each step along the trajectory;

$$0 = c_2 \gamma_{j+1}^2 + c_1 \gamma_{j+1} + c_0, \quad (C5)$$

$$c_2 = \frac{ih}{4v_F} \Delta_{j+\frac{1}{2}}^*, \quad (C6)$$

$$c_1 = 2c_2 \gamma_j + \frac{ih}{v_F} z_{j+\frac{1}{2}} - 1, \quad (C7)$$

$$c_0 = (c_1 - c_2 \gamma_j + 2) \gamma_j - 4c_2^*, \quad (C8)$$

with the solution,

$$\gamma_{j+1} = \frac{2c_0}{-c_1 + \sqrt{c_1^2 - 4c_2c_0}}. \quad (C9)$$

Half-step values, $\Delta_{j+\frac{1}{2}}$ and $z_{j+\frac{1}{2}}$, are calculated by linear interpolation. Nearest-neighbor extrapolation is used at the boundary. If we have $\gamma(s_j)$ at one point, we obtain $\gamma(s_{j+1})$ via Eq. (C9). The solution along the whole trajectory is obtained by starting with the initial boundary value $\gamma(s_0 = \mathbf{R}_{\min})$ and stepping along the trajectory to obtain $\gamma(s_j)$ with $j > 0$. The stepping proceeds until a second boundary is reached (denoted \mathbf{R}_{\max} , see [Fig. 1](#)). We compute $\tilde{\gamma}(s)$ by solving Eq. (26) in the same manner but stepping in the opposite direction from \mathbf{R}_{\max} to \mathbf{R}_{\min} .

APPENDIX D: COMPUTING THE VECTOR POTENTIAL

The induced vector potential is obtained by solving Eq. (7), in the Coulomb gauge, using the Green's function of the 2D Laplacian operator. The current is computed on a $N \times N$ grid, with every grid cell having the area h^2 . We treat the current as being piecewise constant within each grid cell. With our choice of units, this yields

$$\mathbf{A}_{m,n}^{\text{ind}} = -\frac{1}{K^2} \sum_{m'=0}^{N-1} \sum_{n'=0}^{N-1} G_{|m'-m|, |n'-n|} \mathbf{j}_{m'n'}, \quad (D1)$$

$$G_{m,n} = \frac{1}{4\pi} \int_{-\frac{1}{2}h}^{+\frac{1}{2}h} \int_{-\frac{1}{2}h}^{+\frac{1}{2}h} \ln \left(\frac{(x' + hm)^2 + (y' + hn)^2}{\xi_0^2} \right) dx' dy' \quad (D2)$$

$$= \frac{1}{4\pi} \left(\frac{h}{\xi_0} \right)^2 \left(2 \ln \left(\frac{h}{\xi_0} \right) - \ln(2) - 3 + \frac{1}{4} \sum_{a=\pm} \sum_{b=\pm} \Gamma(am, bn) \right), \quad (D3)$$

$$\Gamma(m, n) = \alpha(m, n) + \beta(m, n) + \beta(n, m), \quad (D4)$$

$$\alpha(m, n) = (1 + 2m)(1 + 2n) \ln \left(\frac{1}{2} [1 + 2m]^2 + \frac{1}{2} [1 + 2n]^2 \right), \quad (D5)$$

$$\beta(m, n) = (-2m^2 + 2n^2 + 4n + 1) \tan^{-1} \left(\frac{1 + 2m}{1 + 2n} \right). \quad (D6)$$

G is a normal matrix, so it can be eigendecomposed as $G = V\Lambda V^\top$, where V is a unitary matrix with (normalized) eigenvectors as its columns, and Λ is a diagonal matrix with the eigenvalues as diagonal elements. Equivalently,

$$G_{m,n} = \sum_{k=0}^{N-1} \lambda_k V_m^{(k)} V_n^{(k)}, \quad (D7)$$

where $V_m^{(k)}$ is the eigenvector corresponding to the eigenvalue λ_k . Thus

$$\mathbf{A}_{m,n}^{\text{ind}} = -\frac{1}{\kappa^2} \sum_{k=0}^{N-1} \sum_{m'=0}^{N-1} \sum_{n'=0}^{N-1} \lambda_k V_{|m'-m|}^{(k)} V_{|n'-n|}^{(k)} \mathbf{j}_{m'n'}, \quad (\text{D8})$$

and the cross correlation with the eigenvectors can be done separately. The eigendecomposition is done with Armadillo.^{153,154} In order to increase performance, only N^* eigenvalues are kept. It is computed as the smallest integer obeying

$$\sqrt{1 - \frac{\sum_{n=0}^{N^*-1} \lambda_n^2}{\sum_{n=0}^{N-1} \lambda_n^2}} < \varepsilon, \quad (\text{D9})$$

with the eigenvalues sorted $|\lambda_0| \geq |\lambda_1| \geq \dots \geq |\lambda_{N-1}|$, and $\varepsilon > 0$ being a tolerance chosen by the user via the parameter vector `_potential_error` in the configuration file. If $\varepsilon = 0$, no eigendecomposition is done, and the current density is cross-correlated with the full 2D Green's function.

Finally, the mean of the charge-current density is, in general, not zero due to numerical errors or bad initialization. This has the undesired effect of introducing a constant term in the induced vector potential, which gives rise to an overall phase gradient through the grain and an associated current. For small penetration depths, this can prevent the simulation from converging. In order to remedy this, the mean of the charge-current density is subtracted before solving Eq. (7).

APPENDIX E: CONVERGENCE ACCELERATORS

SuperConga solves Eqs. (7) and (20) self-consistently. With our choice of units, the solution to Eq. (7) is given by

$$\kappa^2 \mathbf{A}_{\text{ind}}(\mathbf{R}) = -\int_{\mathcal{A}} d\mathbf{R}' G(\mathbf{R}, \mathbf{R}') \mathbf{j}(\mathbf{R}'), \quad (\text{E1})$$

in the Coloumb gauge, where $G(\mathbf{R}, \mathbf{R}')$ is the Green's function of the 2D Laplacian. In order to simplify the notation we concatenate the left- and right-hand sides of Eqs. (20) and (E1), yielding

$$\mathbf{x} = \mathbf{g}(\mathbf{x}), \quad (\text{E2})$$

$$\mathbf{x} \equiv \left(\text{Re}[\Delta_{\Gamma_1}], \text{Im}[\Delta_{\Gamma_1}], \dots, \kappa^2 A_x^{\text{ind}}, \kappa^2 A_y^{\text{ind}} \right)^{\top}. \quad (\text{E3})$$

Note that the vector potential is scaled by κ^2 in order to have similar magnitudes of all elements in \mathbf{x} . SuperConga provides several different methods of solving Eq. (E2). Namely, basic Picard iterations, Polyak's⁸¹ “small heavy sphere,” a variant of the Barzilai–Borwein method,⁸² and a custom method. The user controls which method to use by setting the name in the accelerator in Listing 10. By only setting the name, all internal parameters of the accelerator will be set to default values. The user can change the internal parameters by specifying them in the configuration file, or via the CLI.

In the following discussion of different accelerators, it is convenient to define the difference between the right- and left-hand sides of Eq. (E2),

$$\mathbf{d} \equiv \mathbf{g}(\mathbf{x}) - \mathbf{x}, \quad (\text{E4})$$

which enters all accelerator methods.

1. Picard

The simplest way of solving Eq. (E2) is by using Picard iterations,

$$\mathbf{x}_{i+1} = \mathbf{x}_i + \alpha \mathbf{d}_i, \quad (\text{E5})$$

where $\alpha > 0$ is the step size. Picard iterations can converge very slowly, or not at all. Hence, the need for a method accelerating the iterations, and preferably stabilizing them. Calling the Picard method an accelerator is a misnomer as it is the baseline. The other methods should improve on this method. How to use this accelerator is shown in Listing 13.

LISTING 13: Picard accelerator with default values.

```
“accelerator”: {
  “name”: “picard”,
  “step_size”: 1.0
}
```

2. Polyak

In Polyak's method, \mathbf{x} behaves similarly to the position of a particle moving through a viscous fluid, with a low Reynold's number, and a potential field [Eq. (E4)] taking the role of the negative potential gradient],

$$\mathbf{v}_{i+1} = (1 - \beta) \mathbf{v}_i + \alpha \mathbf{d}_i, \quad (\text{E6})$$

$$\mathbf{x}_{i+1} = \mathbf{x}_i + \mathbf{v}_{i+1}, \quad (\text{E7})$$

where $\beta \in (0, 1)$ is the drag, and $\alpha > 0$ is the step size. The velocity is initialized to zero, $\mathbf{v}_0 = \mathbf{0}$. How to use this accelerator is shown in Listing 14.

LISTING 14. Polyak accelerator with default values.

```
“accelerator”: {
  “name”: “polyak”,
  “step_size”: 2.0,
  “drag”: 0.5
}
```

3. Barzilai–Borwein

The Barzilai–Borwein (BB) method⁸² is similar to the Picard method but with an adaptive step size,

$$s_i^{\text{BB1}} = \frac{\|\mathbf{d}_{i-1}\|_2^2}{|\mathbf{d}_{i-1} \cdot (\mathbf{d}_i - \mathbf{d}_{i-1})|}, \quad (\text{E8})$$

$$s_i^{\text{BB2}} = \frac{|\mathbf{d}_{i-1} \cdot (\mathbf{d}_i - \mathbf{d}_{i-1})|}{\|\mathbf{d}_i - \mathbf{d}_{i-1}\|_2^2}, \quad (\text{E9})$$

$$\alpha_{i+1} = \alpha_i \cdot \begin{cases} s_i^{\text{BB1}} & \text{if } i \text{ odd,} \\ s_i^{\text{BB2}} & \text{if } i \text{ even,} \end{cases} \quad (\text{E10})$$

$$\mathbf{x}_{i+1} = \mathbf{x}_i + \alpha_{i+1} \mathbf{d}_i, \quad (\text{E11})$$

where s^{BB1} and s^{BB2} are two different variants of how to scale the step size. Note that we force the step size to be positive. Alternating between the variants gives better performance on the examples provided by SuperConga than using either one of them on its own. What step size to use during the first iteration is a free parameter and can be set by the user. How to use this accelerator is shown in Listing 15.

LISTING 15. Barzilai–Borwein accelerator with default values.

```

“accelerator”: {
  “name”: “barzilai-borwein”,
  “step_size”: 1.0,
  “step_size_max”: 100.0,
  “step_size_min”: 0.001
}

```

4. CongAcc

CongAcc, short for (Super)Conga Accelerator, is our custom made accelerator. Introducing the *component* index, c , where a component is either the real or imaginary part of an order-parameter component, or the x - or y -component of the induced vector potential, in Eq. (E3), the update rule is as follows:

$$s_{i,c} = S_C(\mathbf{m}_{i,c}, \mathbf{d}_{i,c}), \quad (\text{E12})$$

$$\alpha_{i+1,c} = \alpha_{i,c} k^{s_{i,c}}, \quad (\text{E13})$$

$$w_{i+1,c} = \begin{cases} w_{i,c} & \text{if } s_{i,c} \geq s_{\min}, \\ w^* & \text{otherwise,} \end{cases} \quad (\text{E14})$$

$$\mathbf{m}_{i+1,c} = w_{i+1,c} \mathbf{m}_{i,c} + (1 - w_{i+1,c}) \mathbf{d}_{i,c}, \quad (\text{E15})$$

$$\mathbf{x}_{i+1,c} = \mathbf{x}_{i,c} + \alpha_{i+1,c} \mathbf{m}_{i+1,c}, \quad (\text{E16})$$

where S_C is the cosine similarity, $\mathbf{m}_{i,c}$ is an exponential moving average of the component difference $\mathbf{d}_{i,c}$, and k is a constant

determining how much the step size should maximally increase (decrease), which can be set by the user. If $s_{i,c} < s_{\min}$, where s_{\min} is the minimum tolerated similarity, the weight w^* is obtained by solving

$$S_C(w^* \mathbf{m}_{i,c} + (1 - w^*) \mathbf{d}_{i,c}, \mathbf{d}_{i,c}) = s_{\min}, \quad (\text{E17})$$

ensuring that the step will be similar to $\mathbf{d}_{i,c}$. Equation (E17) is solved approximately using the bisection method. Note that we set $w_{0,c} = 0$ during the first iteration.

The reasoning is that if $\mathbf{m}_{i,c}$ and $\mathbf{d}_{i,c}$ roughly point in the same direction, then we should have taken a larger step the previous iteration, and thus, the step size is increased. Analogously, if they roughly point in opposite directions, the step size should decrease.

How to use this accelerator is shown in Listing 16.

LISTING 16: CongAcc accelerator with default values.

```

“accelerator”: {
  “name”: “congaacc”,
  “step_size”: 0.5,
  “step_size_factor”: 1.234,
  “cos_similarity_min”: 0.7,
  “step_size_max”: 100.0,
  “step_size_min”: 0.001
}

```

This accelerator should be regarded as experimental as no thorough analysis of it has been done.

5. Comparing the accelerators

Table I shows the number of iterations needed to reach convergence in all the examples, (a)–(g), for each of the different accelerators available in SuperConga (using default parameters). The

TABLE I. How many iterations each accelerator, with default parameters, require in order to reach convergence on the examples provided with SuperConga. An F means that it did not converge during the 10^4 iterations the simulation was run. The convergence criterion is set to 10^{-5} for all examples. In the geometry column, a radius is denoted \mathcal{R} , and a polygon side length \mathcal{S} .

	Name	T/T_c	Φ/Φ_0	κ	Symmetry	Initialization	Geometry	Picard	Polyak	BB	CongAcc
(a)	dwave_chiral	0.5	0	∞	$d_{xy} + id_{x^2-y^2}$	Bulk	Disk: $\mathcal{R} = 12.5\xi_0$	80	34	25	37
(b)	dwave_octagon	0.5	0	∞	$d_{x^2-y^2}$	Bulk	Octagon: $\mathcal{S} = 10\xi_0$	71	33	25	32
(c)	dwave_plus_swave	0.5	0	∞	$d_{xy} + s$	Bulk	Disk: $\mathcal{R} = 12.5\xi_0$	69	36	30	33
(d)	dwave_phase_crystal	0.1 ^a	0	100	$d_{x^2-y^2}$	Bulk + vortex ^b	Irregular polygon ^c	3027	809	1054	439
(e)	swave_abrikosov_lattice	0.2	20	10	s	Bulk + giant vortex ^d	Square: $\mathcal{S} = 25\xi_0$	1001	313	211	133
(f)	swave_disc_meissner	0.5	0.5	5	s	Bulk	Disk: $\mathcal{R} = 15\xi_0$	F	F	20	43
(g)	swave_disc_vortex	0.5	1.5		s	Bulk + vortex ^e	Disk: $\mathcal{R} = 15\xi_0$	F	F	45	42

^aThe temperature is low, $T < T^*$, so the solution is a phase crystal.^{57,58}

^bThe vortex has a phase winding $n = -1$ and is positioned outside of the grain, yielding a soft phase-gradient through the grain.

^cThe shape of the grain is a square, $\mathcal{S} = 25\xi_0$, with one corner removed, yielding four [100]-interfaces, and one pair-breaking [110]-interface.⁶¹

^dThe giant vortex in the middle has a phase winding $n = -13$. It is unstable and decays into a lattice of singly quantized Abrikosov vortices.

^eThe vortex in the middle has a phase winding $n = -1$.

examples (a)–(c) are fairly easy as even Picard iterations converge quickly. It is examples (d)–(g) that really show the strength of the more advanced methods. In (d) Polyak, BB and CongAcc provide roughly a speedup of 3–7 compared to Picard. Both Picard and Polyak fail to converge in example (f) and (g). This is due to the small penetration depth in those examples, making the default step sizes too large. They can be made to converge by manually changing their parameters, however. The adaptive methods, BB and CongAcc, have no problem with bad initial step sizes; they will quickly adjust the step size to something appropriate.

Listing 17 shows how to run one of the available examples with a specific accelerator, where `<example>` and `<accelerator>` should be replaced with the example and accelerator names, respectively.

LISTING 17: Running an examples with one of the accelerators.

```
python superconga.py simulate -C examples/  
  <example> -A <accelerator>
```

APPENDIX F: UNITS AND BASIS FUNCTIONS

We summarize in Table II our choice of units and the resulting natural scales for observables and other quantities. The available order parameter basis functions are listed in Table III.

TABLE II. Natural units and scales for observables used in this paper and in the framework SuperConga. We use the convention $e = -|e|$ for the charge of the electron but allow for simulations with either positive or negative charge carriers. Often, we let $\hbar = k_B = c = 1$.

Input to quasiclassical theory	
v_F	Fermi velocity
N_F	Normal-state density of states per spin at the Fermi level
T_c	Superconducting transition temperature (critical temperature)
Geometry	
\mathcal{A}	Total superconducting computational area ^a
$\mathcal{V} = N_l d \mathcal{A}$	Volume of a stack of N_l layers with inter-layer distance d
Natural units	
T_c	Temperature in units of critical temperature
$2\pi k_B T_c$	Energy in units of critical temperature
$\xi_0 = \frac{\hbar v_F}{2\pi k_B T_c}$	Length in units of superconducting coherence length
$\Phi_0 = \frac{hc}{2 e }$	Magnetic flux in units of superconducting flux quantum

TABLE II. (Continued.)

Derived units and observables		
λ_0	$\frac{1}{\lambda_0^2} = \frac{4\pi e^2}{c^2} N_F v_F^2$	Penetration depth. Here we use Gaussian units. In SI units $4\pi/c^2 \rightarrow \mu_0$ with $\mu_0 \approx 4\pi \times 10^{-7} \text{J/A}^2\text{m}$
κ	$\frac{\lambda_0}{\xi_0}$	Penetration depth in units of coherence length
α	$\frac{B_{\text{ext}} \mathcal{A}}{\Phi_0}$	Symmetric gauge: external magnetic flux penetrating the superconducting computational area in units of the flux quantum solenoid gauge: external magnetic flux penetrating a circular hole centered at origo in units of the flux quantum
A_0	$\frac{\Phi_0}{\pi \xi_0^2}$	Vector potential
B_0	$\frac{\Phi_0}{\pi \xi_0^2}$	Magnetic flux density
j_0	$= 2\pi k_B T_c e v_F N_F$	Charge-current density
m_0	$= j_0 \xi_0 \mathcal{V}$	Magnetic moment
$\Omega_{\mathcal{A}}$	$= (2\pi k_B T_c)^2 N_F \mathcal{A}$	Free energy

^aWe use calligraphic capital letters for quantities related to geometry, such as area \mathcal{A} or disk radius \mathcal{R} .

TABLE III. The available symmetries of the order parameter that can be chosen in SuperConga. Note that any linear combination forming a multi-component order parameter is also allowed.

Order-parameter basis functions		
Symmetry class for D_{4h}	Order parameter $\Delta(\mathbf{k})$	Basis function $\eta(\varphi)$ for circular Fermi surface
A_{1g} (s)	1	1
A_{2g} ($g_{xy}(x^2 - y^2)$)	$k_x k_y (k_x^2 - k_y^2)$	$\sqrt{2} \sin(4\varphi)$
B_{1g} ($d_{x^2 - y^2}$)	$k_x^2 - k_y^2$	$\sqrt{2} \cos(2\varphi)$
B_{2g} (d_{xy})	$k_x k_y$	$\sqrt{2} \sin(2\varphi)$

REFERENCES

- C. C. Tsuei and J. R. Kirtley, "Pairing symmetry in cuprate superconductors," *Rev. Mod. Phys.* **72**, 969–1016 (2000).
- J. R. Kirtley, C. Tsuei, C. Ariando, S. Harkema, and H. Hilgenkamp, "Angle-resolved phase-sensitive determination of the in-plane gap symmetry in $\text{YBa}_2\text{Cu}_3\text{O}_{7-\delta}$," *Nat. Phys.* **2**, 190–194 (2005).
- N. Kokubo, S. Okayasu, A. Kanda, and B. Shinozaki, "Scanning squid microscope study of vortex polygons and shells in weak-pinning disks of an amorphous superconducting film," *Phys. Rev. B* **82**, 014501 (2010).
- J. A. Bert, B. Kalisky, C. Bell, M. Kim, Y. Hikita, H. Y. Hwang, and K. A. Moler, "Direct imaging of the coexistence of ferromagnetism and superconductivity at the $\text{LaAlO}_3/\text{SrTiO}_3$ interface," *Nat. Phys.* **7**, 767 (2011).
- D. Vasyukov, Y. Anahory, L. Embon, D. Halbertal, J. Cuppens, L. Neeman, A. Finkler, Y. Segev, Y. Myasoedov, M. L. Rappaport, M. E. Huber, and E. Zeldov, "A scanning superconducting quantum interference device with single electron spin sensitivity," *Nat. Nanotechnol.* **8**, 639 (2013).

- ⁶H. Hess, R. Robinson, R. Dynes, J. M. J. Valles, and J. V. Waszczak, "Scanning-tunneling-microscope observation of the Abrikosov flux lattice and the density of states near and inside a fluxoid," *Phys. Rev. Lett.* **62**, 214–216 (1989).
- ⁷C. Renner, A. D. Kent, P. Niedermann, Ø. Fischer, and F. Lévy, "Scanning tunneling spectroscopy of a vortex core from the clean to the dirty limit," *Phys. Rev. Lett.* **67**, 1650–1652 (1991).
- ⁸I. Maggio-Aprile, C. Renner, A. Erb, E. Walker, and O. Fischer, "Direct vortex lattice imaging and tunneling spectroscopy of flux lines on $\text{YBa}_2\text{Cu}_3\text{O}_{7-\delta}$," *Phys. Rev. Lett.* **75**, 2754–2757 (1995).
- ⁹A. Yazdani, B. Jones, C. Lutz, M. F. Crommie, and D. M. Eigler, "Probing the local effects of magnetic impurities on superconductivity," *Science* **275**, 1767–1770 (1997).
- ¹⁰S. Pan, E. Hudson, A. Gupta, K. Ng, H. Eisaki, S. Uchida, and J. Davis, "STM studies of the electronic structure of vortex cores in $\text{Bi}_2\text{Sr}_2\text{CaCu}_2\text{O}_{8+\delta}$," *Phys. Rev. Lett.* **85**, 1536–1539 (2000).
- ¹¹S. Pan, E. Hudson, K. Lang, H. Eisaki, S. Uchida, and J. Davis, "Imaging the effects of individual zinc impurity atoms on superconductivity in $\text{Bi}_2\text{Sr}_2\text{CaCu}_2\text{O}_{8+\delta}$," *Nature* **403**, 746–750 (2000).
- ¹²J. E. Hoffman, K. McElroy, D.-H. Lee, K. M. Lang, H. Eisaki, S. Uchida, and J. C. Davis, "Imaging quasiparticle interference in $\text{Bi}_2\text{Sr}_2\text{CaCu}_2\text{O}_{8+\delta}$," *Science* **297**, 1148–1151 (2002).
- ¹³I. Guillamón, H. Suderow, S. Vieira, L. Cario, P. Diener, and P. Rodière, "Superconducting density of states and vortex cores of 2H-NbS_2 ," *Phys. Rev. Lett.* **101**, 166407 (2008).
- ¹⁴D. Roditchev, C. Brun, L. Serrier-Garcia, J. C. Cuevas, V. H. L. Bessa, M. V. Milošević, F. Debontridder, V. Stolyarov, and T. Cren, "Direct observation of Josephson vortex cores," *Nat. Phys.* **11**, 332–337 (2015).
- ¹⁵C. Berthod, I. Maggio-Aprile, J. Bruér, A. Erb, and C. Renner, "Observation of Caroli-de Gennes–Matricon vortex states in $\text{YBa}_2\text{Cu}_3\text{O}_{7-\delta}$," *Phys. Rev. Lett.* **119**, 237001 (2017).
- ¹⁶C. A. Bolle, V. Aksyuk, F. Pardo, P. L. Gammel, E. Zeldov, E. Bucher, R. Boie, D. J. Bishop, and D. R. Nelson, "Observation of mesoscopic vortex physics using micromechanical oscillators," *Nature* **399**, 43–46 (1999).
- ¹⁷A. C. Bleszynski-Jayich, W. E. Shanks, B. Peaudecerf, E. Ginossar, F. v Oppen, L. Glazman, and J. G. E. Harris, "Persistent currents in normal metal rings," *Science* **326**, 272–275 (2009).
- ¹⁸J. Jang, D. G. Ferguson, V. Vakaryuk, R. Budakian, S. B. Chung, P. M. Goldbart, and Y. Maeno, "Observation of half-height magnetization steps in Sr_2RuO_4 ," *Science* **331**, 186–188 (2011).
- ¹⁹A. K. Geim, I. V. Grigorieva, S. V. Dubonos, J. G. S. Lok, J. C. Maan, A. E. Filippov, and F. M. Peeters, "Phase transitions in individual sub-micrometre superconductors," *Nature* **390**, 259–262 (1997).
- ²⁰M. Morelle, J. Bekaert, and V. V. Moshchalkov, "Influence of sample geometry on vortex matter in superconducting microstructures," *Phys. Rev. B* **70**, 094503 (2004).
- ²¹I. V. Grigorieva, W. Escoffier, J. Richardson, L. Y. Vinnikov, S. Dubonos, and V. Oboznov, "Direct observation of vortex shells and magic numbers in mesoscopic superconducting disks," *Phys. Rev. Lett.* **96**, 077005 (2006).
- ²²V. V. Khotkevych, M. V. Milošević, and S. J. Bending, "A scanning hall probe microscope for high resolution magnetic imaging down to 300 mK," *Rev. Sci. Instrum.* **79**, 123708 (2008).
- ²³P. J. Curran, S. J. Bending, W. M. Desoky, A. S. Gibbs, S. L. Lee, and A. P. Mackenzie, "Search for spontaneous edge currents and vortex imaging in Sr_2RuO_4 mesostructures," *Phys. Rev. B* **89**, 144504 (2014).
- ²⁴J.-Y. Ge, V. N. Gladilin, J. Tempere, V. S. Zharinov, J. V. d Vondel, J. T. Devreese, and V. V. Moshchalkov, "Direct visualization of vortex ice in a nanostructured superconductor," *Phys. Rev. B* **96**, 134515–134517 (2017).
- ²⁵M. T. Tuominen, J. M. Hergenrother, T. S. Tighe, and M. Tinkham, "Experimental evidence for parity-based 2e periodicity in a superconducting single-electron tunneling transistor," *Phys. Rev. Lett.* **69**, 1997–2000 (1992).
- ²⁶P. Lafarge, P. Joyez, D. Esteve, C. Urbina, and M. H. Devoret, "Measurement of the even-odd free-energy difference of an isolated superconductor," *Phys. Rev. Lett.* **70**, 994–997 (1993).
- ²⁷D. C. Ralph, C. T. Black, and M. Tinkham, "Spectroscopic measurements of discrete electronic states in single metal particles," *Phys. Rev. Lett.* **74**, 3241–3244 (1995).
- ²⁸S. E. Kubatkin, A. Y. Tzalenchuk, Z. G. Ivanov, P. Delsing, R. I. Shekhter, and T. Claeson, "Coulomb blockade electrometer with a high- T_c island," *JETP Lett.* **63**, 126–132 (1996).
- ²⁹D. Gustafsson, D. Golubev, M. Fogelström, T. Claeson, S. Kubatkin, T. Bauch, and F. Lombardi, "Fully gapped superconductivity in a nanometre-size $\text{YBa}_2\text{Cu}_3\text{O}_{7-\delta}$ island enhanced by a magnetic field," *Nat. Nanotechnol.* **8**, 25–30 (2013).
- ³⁰J. Bardeen, L. N. Cooper, and J. R. Schrieffer, "Theory of superconductivity," *Phys. Rev.* **108**, 1175–1204 (1957).
- ³¹J. R. Schrieffer, *Theory of Superconductivity, Advanced Book Classics* (Perseus Books, 1999).
- ³²P. G. de Gennes, *Superconductivity of Metals and Alloys*, Advanced Book Classics (Addison-Wesley Publishing Company, 1989).
- ³³M. Tinkham, *Introduction to Superconductivity: Second Edition*, Dover Books on Physics (Dover Publications, 2004).
- ³⁴V. P. Mineev and K. V. Samokhin, *Introduction to Unconventional Superconductivity* (Gordon and Breach Science Publishers, 1999).
- ³⁵J. Serene and D. Rainer, "The quasiclassical approach to superfluid ^3He ," *Phys. Rep.* **101**, 221–311 (1983).
- ³⁶G. Eilenberger, "Transformation of Gorkov's equation for type II superconductors into transport-like equations," *Z. Phys. A: Hadrons Nucl.* **214**, 195–213 (1968).
- ³⁷A. I. Larkin and Y. N. Ovchinnikov, "Quasiclassical method in the theory of superconductivity," *Zh. Eksp. Teor. Fiz.* **55**, 2262 (1968) [*Sov. Phys. JETP* **28**, 1200 (1969)].
- ³⁸G. M. Eliashberg, "Inelastic electron collisions and nonequilibrium stationary states in superconductors," *Zh. Eksp. Teor. Fiz.* **61**, 1254 (1971) [*Sov. Phys. JETP* **34**, 668 (1972)].
- ³⁹A. A. Abrikosov, L. P. Gorkov, and I. E. Dzyaloshinski, *Methods of Quantum Field Theory in Statistical Physics* (Dover Publications, Inc., 1975).
- ⁴⁰D. Rainer and J. A. Sauls, "Strong-coupling theory of superconductivity," in *Superconductivity: From Basic Physics to the Latest Developments* (World Scientific, 1995), pp. 45–78.
- ⁴¹N. Kopnin, *Theory of Nonequilibrium Superconductivity* (Clarendon Press, 2001).
- ⁴²M. Fogelström and J. Kurkijärvi, "Quasiclassical theory of vortices in $^3\text{He-B}$," *J. Low Temp. Phys.* **98**, 195–226 (1995).
- ⁴³J. Sauls and M. Eschrig, "Vortices in chiral, spin-triplet superconductors and superfluids," *New J. Phys.* **11**, 075008 (2009).
- ⁴⁴M. A. Silaev, E. V. Thuneberg, and M. Fogelström, "Lifshitz transition in the double-core vortex in $^3\text{He-B}$," *Phys. Rev. Lett.* **115**, 235301 (2015).
- ⁴⁵K. M. Seja and T. Löfwander, "Finite element method for the quasiclassical theory of superconductivity," *Phys. Rev. B* **106**, 144511 (2022).
- ⁴⁶A. A. Andreev, "The thermal conductivity of the intermediate state in superconductors," *Zh. Eksp. Teor. Fiz.* **46**, 1823 (1964) [*Sov. Phys. JETP* **19**, 1228 (1964)].
- ⁴⁷A. Shelankov and M. Ozana, "Quasiclassical theory of superconductivity: A multiple-interface geometry," *Phys. Rev. B* **61**, 7077–7100 (2000).
- ⁴⁸K. D. Usadel, "Generalized diffusion equation for superconducting alloys," *Phys. Rev. Lett.* **25**, 507–509 (1970).
- ⁴⁹J. A. X. Alexander, T. P. Orlando, D. Rainer, and P. M. Tedrow, "Theory of Fermi-liquid effects in high-field tunneling," *Phys. Rev. B* **31**, 5811–5825 (1985).
- ⁵⁰W. Belzig, F. K. Wilhelm, C. Bruder, G. Schön, and A. D. Zaikin, "Quasiclassical Green's function approach to mesoscopic superconductivity," *Superlattices Microstruct.* **25**, 1251–1288 (1999).
- ⁵¹F. S. Bergeret, M. Silaev, P. Virtanen, and T. T. Heikkilä, "Colloquium: Nonequilibrium effects in superconductors with a spin-splitting field," *Rev. Mod. Phys.* **90**, 041001 (2018).
- ⁵²J. A. Sauls, "Theory of disordered superconductors with applications to non-linear current response," *Prog. Theor. Exp. Phys.* **2022**, 033103.
- ⁵³Y. V. Nazarov, "Novel circuit theory of Andreev reflection," *Superlattices Microstruct.* **25**, 1221–1231 (1999).
- ⁵⁴M. Amundsen and J. Linder, "General solution of 2D and 3D superconducting quasiclassical systems: Coalescing vortices and nanoisland geometries," *Sci. Rep.* **6**, 22765 (2016).

- ⁵⁵M. Timmermans, L. Serrier-Garcia, M. Perini, J. Van de Vondel, and V. V. Moshchalkov, "Direct observation of condensate and vortex confinement in nanostructured superconductors," *Phys. Rev. B* **93**, 054514 (2016).
- ⁵⁶A. B. Vorontsov and J. Sauls, "Thermodynamic properties of thin films of superfluid $^3\text{He-A}$," *Phys. Rev. B* **68**, 064508 (2003).
- ⁵⁷M. Håkansson, T. Löfwander, and M. Fogelström, "Spontaneously broken time-reversal symmetry in high-temperature superconductors," *Nat. Phys.* **11**, 755 (2015).
- ⁵⁸P. Holmval, M. Fogelström, T. Löfwander, and A. B. Vorontsov, "Phase crystals," *Phys. Rev. Res.* **2**, 013104 (2020).
- ⁵⁹S. Yip and A. Garg, "Superconducting states of reduced symmetry: General order parameters and physical implications," *Phys. Rev. B* **48**, 3304–3308 (1993).
- ⁶⁰M. Graf, D. Rainer, and J. Sauls, "Coupled two-dimensional Fermi liquids as a model for layered superconductors: Basic equations and elementary results," *Phys. Rev. B* **47**, 12089–12098 (1993).
- ⁶¹N. W. Wennnerdal, A. Ask, P. Holmval, T. Löfwander, and M. Fogelström, "Breaking time-reversal and translational symmetry at edges of d-wave superconductors: Microscopic theory and comparison with quasiclassical theory," *Phys. Rev. Res.* **2**, 043198 (2020).
- ⁶²M. Eschrig, "Distribution functions in nonequilibrium theory of superconductivity and Andreev spectroscopy in unconventional superconductors," *Phys. Rev. B* **61**, 9061–9076 (2000).
- ⁶³J. M. Luttinger and J. C. Ward, "Ground-state energy of a many-fermion system. II," *Phys. Rev.* **118**, 1417–1427 (1960).
- ⁶⁴E. V. Thuneberg, J. Kurkijärvi, and D. Rainer, "Elementary-flux-pinning potential in type-II superconductors," *Phys. Rev. B* **29**, 3913–3923 (1984).
- ⁶⁵A. V. Zaitsev, "Quasiclassical equation of the theory of superconductivity for contiguous metals and the properties of constricted microstructures," *Zh. Eksp. Teor. Fiz.* **86**, 1742 (1984).
- ⁶⁶G. Kieselmann, "Self-consistent calculations of the pair potential and the tunneling density of states in proximity contacts," *Phys. Rev. B* **35**, 6762–6770 (1987).
- ⁶⁷A. Millis, D. Rainer, and J. A. Sauls, "Quasiclassical theory of superconductivity near magnetically active interfaces," *Phys. Rev. B* **38**, 4504–4515 (1988).
- ⁶⁸M. Fogelström, "Josephson currents through spin-active interfaces," *Phys. Rev. B* **62**, 11812–11819 (2000).
- ⁶⁹E. Zhao, T. Löfwander, and J. A. Sauls, "Nonequilibrium superconductivity near spin-active interfaces," *Phys. Rev. B* **70**, 134510 (2004).
- ⁷⁰M. Eschrig, "Scattering problem in nonequilibrium quasiclassical theory of metals and superconductors: General boundary conditions and applications," *Phys. Rev. B* **80**, 134511 (2009).
- ⁷¹M. Ozana and A. Shelankov, "Quasiclassical theory of superconductivity: Interfering paths," *Phys. Rev. B* **65**, 014510 (2001).
- ⁷²A. Samoilenska and E. Babaev, "Boundary states with elevated critical temperatures in Bardeen-Cooper-Schrieffer superconductors," *Phys. Rev. B* **101**, 134512 (2020).
- ⁷³C. Hainzl, B. Roos, and R. Seiringer, "Boundary superconductivity in the BCS model," *arXiv:2201.08090* (2022).
- ⁷⁴Y. Nagato, K. Nagai, and J. Hara, "Theory of the Andreev reflection and the density of states in proximity contact normal-superconducting infinite double-layer," *J. Low Temp. Phys.* **93**, 33–56 (1993).
- ⁷⁵N. Schopohl and K. Maki, "Quasiparticle spectrum around a vortex line in a d-wave superconductor," *Phys. Rev. B* **52**, 490–493 (1995).
- ⁷⁶N. Schopohl, "Transformation of the eilenberger equations of superconductivity to a scalar Riccati equation," *arXiv:cond-mat/9804064* (1998).
- ⁷⁷T. Ozaki, "Continued fraction representation of the Fermi-Dirac function for large-scale electronic structure calculations," *Phys. Rev. B* **75**, 035123 (2007).
- ⁷⁸NVIDIA Corporation, *NVIDIA CUDA C Programming Guide Version 11.2.0* (NVIDIA Corporation, 2021).
- ⁷⁹J. Nickolls, I. Buck, M. Garland, and K. Skadron, "Scalable parallel programming with CUDA," *Queue* **6**, 40–53 (2008).
- ⁸⁰SuperConga Team (2022). "SuperConga online documentation and user manual," GitHub, <https://superconga.gitlab.io/superconga-doc/>
- ⁸¹B. Polyak, "Some methods of speeding up the convergence of iteration methods," *USSR Comput. Math. Math. Phys.* **4**, 1–17 (1964).
- ⁸²J. Barzilai and J. M. Borwein, "Two-point step size gradient methods," *IMA J. Numer. Anal.* **8**, 141–148 (1988).
- ⁸³SuperConga Team (2022). "SuperConga," GitHub, <https://gitlab.com/superconga/superconga>
- ⁸⁴P. Holmval, A. B. Vorontsov, M. Fogelström, and T. Löfwander, "Spontaneous symmetry breaking at surfaces of d-wave superconductors: Influence of geometry and surface ruggedness," *Phys. Rev. B* **99**, 184511 (2019).
- ⁸⁵C. Caroli, P. De Gennes, and J. Matricon, "Bound fermion states on a vortex line in a type II superconductor," *Phys. Lett.* **9**, 307–309 (1964).
- ⁸⁶H. Kim, Y. Nagai, L. Rózsa, D. Schreyer, and R. Wiesendanger, "Anisotropic non-split zero-energy vortex bound states in a conventional superconductor," *Appl. Phys. Rev.* **8**, 031417 (2021).
- ⁸⁷C. P. Bean and J. D. Livingston, "Surface barrier in type-II superconductors," *Phys. Rev. Lett.* **12**, 14–16 (1964).
- ⁸⁸H. J. Zhao, V. R. Misko, F. M. Peeters, V. Oboznov, S. V. Dubonos, and I. V. Grigorieva, "Vortex states in mesoscopic superconducting squares: Formation of vortex shells," *Phys. Rev. B* **78**, 104517 (2008).
- ⁸⁹V. R. Misko, H. J. Zhao, F. M. Peeters, V. Oboznov, S. V. Dubonos, and I. V. Grigorieva, "Formation of vortex shells in mesoscopic superconducting squares," *Superconductor Sci. Technol.* **22**, 034001 (2009).
- ⁹⁰L.-F. Zhang, L. Covaci, M. V. Milošević, G. R. Berdysorov, and F. M. Peeters, "Unconventional vortex states in nanoscale superconductors due to shape-induced resonances in the inhomogeneous cooper-pair condensate," *Phys. Rev. Lett.* **109**, 107001 (2012).
- ⁹¹L.-F. Zhang, L. Covaci, M. V. Milošević, G. R. Berdysorov, and F. M. Peeters, "Vortex states in nanoscale superconducting squares: The influence of quantum confinement," *Phys. Rev. B* **88**, 144501 (2013).
- ⁹²H. J. Zhao, V. R. Misko, F. M. Peeters, S. Dubonos, V. Oboznov, and I. V. Grigorieva, "Vortex configurations in mesoscopic superconducting triangles: Finite-size and shape effects," *Europhys. Lett.* **83**, 17008 (2008).
- ⁹³L. R. E. Cabral and J. A. Aguiar, "Vortex configurations in thin superconducting equilateral triangles," *Phys. Rev. B* **80**, 214533 (2009).
- ⁹⁴N. Kokubo, H. Miyahara, S. Okayasu, and T. Nojima, "Vortex shells in mesoscopic triangles of amorphous superconducting thin films," *Physica C* **530**, 42–45 (2016).
- ⁹⁵W. Wu, H. J. Zhao, B. Li, and Z. X. Shi, "Finite-size and shape effects on vortex configurations in superconducting isosceles right triangles," *Supercond. Sci. Technol.* **30**, 125004 (2017).
- ⁹⁶H. T. Huy, M. Kato, and T. Ishida, "Vortex states in de facto mesoscopic Mo80Ge20pentagon plates," *Superconductor Sci. Technol.* **26**, 065001 (2013).
- ⁹⁷H. J. Fink and A. G. Presson, "Magnetic irreversible solution of the Ginzburg-Landau equations," *Phys. Rev.* **151**, 219–228 (1966).
- ⁹⁸V. V. Moshchalkov, M. Baert, V. V. Metlushko, E. Rosseel, M. J. Van Bael, K. Temst, R. Jonckheere, and Y. Bruynseraede, "Magnetization of multiple-quanta vortex lattices," *Phys. Rev. B* **54**, 7385–7393 (1996).
- ⁹⁹V. A. Schweigert, F. M. Peeters, and P. S. Deo, "Vortex phase diagram for mesoscopic superconducting disks," *Phys. Rev. Lett.* **81**, 2783–2786 (1998).
- ¹⁰⁰A. Kanda, B. J. Baelus, F. M. Peeters, K. Kadowaki, and Y. Ootuka, "Experimental evidence for giant vortex states in a mesoscopic superconducting disk," *Phys. Rev. Lett.* **93**, 257002 (2004).
- ¹⁰¹D. S. Golubović, M. V. Milošević, F. M. Peeters, and V. V. Moshchalkov, "Magnetically induced splitting of a giant vortex state in a mesoscopic superconducting disk," *Phys. Rev. B* **71**, 180502 (2005).
- ¹⁰²T. Cren, L. Serrier-Garcia, F. Debontridder, and D. Roditchev, "Vortex fusion and giant vortex states in confined superconducting condensates," *Phys. Rev. Lett.* **107**, 097202 (2011).
- ¹⁰³L. F. Chibotaru, A. Ceulemans, V. Bruynndoncx, and V. V. Moshchalkov, "Symmetry-induced formation of antivortices in mesoscopic superconductors," *Nature* **408**, 833–835 (2000).
- ¹⁰⁴M. Iavarone, A. Scarfato, F. Bobba, M. Longobardi, G. Karapetrov, V. Novosad, V. Yefremenko, F. Giubileo, and A. M. Cucolo, "Imaging the spontaneous formation of vortex-antivortex pairs in planar superconductor/ferromagnet hybrid structures," *Phys. Rev. B* **84**, 024506 (2011).
- ¹⁰⁵C. D. Giorgio, D. D'Agostino, A. M. Cucolo, M. Iavarone, A. Scarfato, G. Karapetrov, S. A. Moore, M. Polichetti, D. Mancusi, S. Pace, V. Novosad, V. Yefremenko, and F. Bobba, "Superconducting vortex-antivortex pairs:

- Nucleation and confinement in magnetically coupled superconductor ferromagnet hybrids," in *Vortex Dynamics and Optical Vortices*, edited by H. Perez-de Tejada (InTech, 2017), Chap. 03.
- ¹⁰⁶J. Simmendinger, M. Bihler, A. M. Ionescu, M. Weigand, G. Schütz, and J. Albrecht, "Bound and stable vortex-antivortex pairs in high- T_c superconductors," *New J. Phys.* **22**, 123035 (2020).
 - ¹⁰⁷A. B. Vorontsov, "Broken translational and time-reversal symmetry in unconventional superconducting films," *Phys. Rev. Lett.* **102**, 177001 (2009).
 - ¹⁰⁸P. Holmval, "Modeling mesoscopic unconventional superconductors," licentiate thesis (Chalmers University of Technology, 2017).
 - ¹⁰⁹P. Holmval, T. Löfwander, and M. Fogelström, "Spontaneous generation of fractional vortex-antivortex pairs at single edges of high- T_c superconductors," *J. Phys.: Conf. Ser.* **969**, 012037 (2018).
 - ¹¹⁰P. Holmval, A. B. I. Vorontsov, M. Fogelström, and T. Löfwander, "Broken translational symmetry at edges of high-temperature superconductors," *Nat. Commun.* **9**, 2190 (2018).
 - ¹¹¹P. Holmval, "Crystallization of the superconducting phase in unconventional superconductors," Ph.D. thesis (Chalmers University of Technology, 2019).
 - ¹¹²A. A. Abrikosov, "On the magnetic properties of superconductors of the second group," *Sov. Phys. JETP* **5**, 1174–1182 (1957).
 - ¹¹³J. Pearl, "Current distribution in superconducting films carrying quantized fluxoids," *Appl. Phys. Lett.* **5**, 65–66 (1964).
 - ¹¹⁴D. Xu, S. K. Yip, and J. A. Sauls, "Nonlinear Meissner effect in unconventional superconductors," *Phys. Rev. B* **51**, 16233–16253 (1995).
 - ¹¹⁵J. Sakurai and J. Napolitano, *Modern Quantum Mechanics* (Addison-Wesley, 2011).
 - ¹¹⁶Y. Aharonov and D. Bohm, "Significance of electromagnetic potentials in the quantum theory," *Phys. Rev.* **115**, 485–491 (1959).
 - ¹¹⁷F. N. Rybakov and E. Babaev, "Flux qubit based on hybrid ferromagnetic-superconducting device," [arXiv:2203.08752](https://arxiv.org/abs/2203.08752) [cond-mat.supr-con] (2022).
 - ¹¹⁸B. S. Deaver and W. M. Fairbank, "Experimental evidence for quantized flux in superconducting cylinders," *Phys. Rev. Lett.* **7**, 43–46 (1961).
 - ¹¹⁹R. Doll and M. Näbauer, "Experimental proof of magnetic flux quantization in a superconducting ring," *Phys. Rev. Lett.* **7**, 51–52 (1961).
 - ¹²⁰L. Onsager, "Magnetic flux through a superconducting ring," *Phys. Rev. Lett.* **7**, 50–50 (1961).
 - ¹²¹N. Byers and C. N. Yang, "Theoretical considerations concerning quantized magnetic flux in superconducting cylinders," *Phys. Rev. Lett.* **7**, 46–49 (1961).
 - ¹²²W. Brenig, "Remark concerning quantized magnetic flux in superconductors," *Phys. Rev. Lett.* **7**, 337–339 (1961).
 - ¹²³W. A. Little and R. D. Parks, "Observation of quantum periodicity in the transition temperature of a superconducting cylinder," *Phys. Rev. Lett.* **9**, 9–12 (1962).
 - ¹²⁴R. P. Groff and R. D. Parks, "Fluxoid quantization and field-induced depairing in a hollow superconducting microcylinder," *Phys. Rev.* **176**, 567–580 (1968).
 - ¹²⁵O. Buisson, P. Gandit, R. Rammal, Y. Wang, and B. Pannetier, "Magnetization oscillations of a superconducting disk," *Phys. Lett. A* **150**, 36–42 (1990).
 - ¹²⁶C.-R. Hu, "Midgap surface states as a novel signature for $d_{x^2-y^2}^2$ -wave superconductivity," *Phys. Rev. Lett.* **72**, 1526–1529 (1994).
 - ¹²⁷S.-I. Suzuki and Y. Asano, "Paramagnetic instability of small topological superconductors," *Phys. Rev. B* **89**, 184508 (2014).
 - ¹²⁸D. Chakraborty, T. Löfwander, M. Fogelström, and A. M. Black-Schaffer, "Disorder-robust phase crystal in high-temperature superconductors stabilized by strong correlations," *npj Quantum Mater.* **7**, 44 (2022).
 - ¹²⁹A. L. Fetter, "Flux-penetration in a thin superconducting disk," *Phys. Rev. B* **22**, 1200–1213 (1980).
 - ¹³⁰V. G. Kogan, J. R. Clem, and R. G. Mints, "Properties of mesoscopic superconducting thin-film rings: London approach," *Phys. Rev. B* **69**, 064516 (2004).
 - ¹³¹S. Higashitani, "Mechanism of paramagnetic Meissner effect in high-temperature superconductors," *J. Soc. Phys. Jpn.* **66**, 2556 (1997).
 - ¹³²M. Fogelström, D. Rainer, and J. A. Sauls, "Tunneling into current-carrying surface states of high- T_c superconductors," *Phys. Rev. Lett.* **79**, 281–284 (1997).
 - ¹³³M. Fogelström, D. Rainer, and J. Sauls, "Nonlinear magnetic field dependence of the conductance in d-wave normal-metal/insulator/superconductor tunnel junctions," *Phys. Rev. B* **70**, 012503 (2004).
 - ¹³⁴J. R. Clem, M. W. Coffey, and Z. Hao, "Lower critical field of a Josephson-coupled layer model of high- T_c superconductors," *Phys. Rev. B* **44**, 2732–2738 (1991).
 - ¹³⁵NVIDIA Corporation, *NVIDIA CUDA Enabled Devices* (NVIDIA Corporation, 2021).
 - ¹³⁶L. B.-V. Horn and K. A. Moler, "Superscreen: An open-source package for simulating the magnetic response of two-dimensional superconducting devices," *Comput. Phys. Commun.* **280**, 108464 (2022).
 - ¹³⁷I. Sadovskyy, A. Koshelev, C. Phillips, D. Karpeyev, and A. Glatz, "Stable large-scale solver for Ginzburg–Landau equations for superconductors," *J. Comput. Phys.* **294**, 639–654 (2015).
 - ¹³⁸F. N. Rybakov and E. Babaev, <http://quantumandclassical.com/excalibur/> for "Excalibur software," 2022.
 - ¹³⁹G. Kresse and J. Furthmüller, "Efficiency of *ab initio* total energy calculations for metals and semiconductors using a plane-wave basis set," *Comput. Mater. Sci.* **6**, 15–50 (1996).
 - ¹⁴⁰G. Kresse and J. Furthmüller, "Efficient iterative schemes for *ab initio* total-energy calculations using a plane-wave basis set," *Phys. Rev. B* **54**, 11169–11186 (1996).
 - ¹⁴¹P. Giannozzi, S. Baroni, N. Bonini, M. Calandra, R. Car, C. Cavazzoni, D. Ceresoli, G. L. Chiarotti, M. Cococcioni, I. Dabo, A. D. Corso, S. de Gironcoli, S. Fabris, G. Fratesi, R. Gebauer, U. Gerstmann, C. Gougousis, A. Kokalj, M. Lazzeri, L. Martin-Samos, N. Marzari, F. Mauri, R. Mazzarello, S. Paolini, A. Pasquarello, L. Paulatto, C. Sbraccia, S. Scandolo, G. Sclauzero, A. P. Seitsonen, A. Smogunov, P. Umari, and R. M. Wentzcovitch, "QUANTUM ESPRESSO: A modular and open-source software project for quantum simulations of materials," *J. Phys.: Condens. Matter* **21**, 395502 (2009).
 - ¹⁴²P. Giannozzi, O. Andreussi, T. Brumme, O. Bunau, M. B. Nardelli, M. Calandra, R. Car, C. Cavazzoni, D. Ceresoli, M. Cococcioni, N. Colonna, I. Carnimeo, A. D. Corso, S. de Gironcoli, P. Delugas, R. A. DiStasio, A. Ferretti, A. Floris, G. Fratesi, G. Fugallo, R. Gebauer, U. Gerstmann, F. Giustino, T. Gorni, J. Jia, M. Kawamura, H.-Y. Ko, A. Kokalj, E. Küçükbenli, M. Lazzeri, M. Marsili, N. Marzari, F. Mauri, N. L. Nguyen, H.-V. Nguyen, A. O. de-la Roza, L. Paulatto, S. Poncè, D. Rocca, R. Sabatini, B. Santra, M. Schlipf, A. P. Seitsonen, A. Smogunov, I. Timrov, T. Thonhauser, P. Umari, N. Vast, X. Wu, and S. Baroni, "Advanced capabilities for materials modelling with quantum ESPRESSO," *J. Phys.: Condens. Matter* **29**, 465901 (2017).
 - ¹⁴³M. Nakhaee, S. A. Ketabi, and F. M. Peeters, "Tight-binding studio: A technical software package to find the parameters of tight-binding Hamiltonian," *Comput. Phys. Commun.* **254**, 107379 (2020).
 - ¹⁴⁴C. W. Groth, M. Wimmer, A. R. Akhmerov, and X. Waintal, "Kwant: A software package for quantum transport," *New J. Phys.* **16**, 063065 (2014).
 - ¹⁴⁵K. Björnson, "Tbtk: A quantum mechanics software development kit," *SoftwareX* **9**, 205–210 (2019).
 - ¹⁴⁶T. Löthman, C. Triola, J. Cayao, and A. M. Black-Schaffer, "Efficient numerical method for evaluating normal and anomalous time-domain equilibrium green's functions in inhomogeneous systems," *Phys. Rev. B* **104**, 125405 (2021).
 - ¹⁴⁷Y. Nagai, "N-independent localized Krylov–Bogoliubov–de Gennes method: Ultra-fast numerical approach to large-scale inhomogeneous superconductors," *J. Phys. Soc. Jpn.* **89**, 074703 (2020).
 - ¹⁴⁸N. Bell and J. Hoberock, "Thrust: A productivity-oriented library for CUDA," in *GPU Computing Gems Jade Edition*, Applications of GPU Computing Series, edited by W. Mei and W. Hwu (Morgan Kaufmann, Boston, 2012), Chap. 26, pp. 359–371.
 - ¹⁴⁹K. Kaczmarek and P. Rżakowski, "Thrust and CUDA in data intensive algorithms," in *New Trends in Databases and Information Systems*, edited by M. Pechenizkiy and M. Wojciechowski (Springer Berlin Heidelberg, 2013), pp. 37–46.
 - ¹⁵⁰E. Wynters, "Examples from computational geometry that demonstrate the potential of using the thrust library to implement parallel processing on GPUs," *J. Comput. Sci. Coll.* **28**, 148–155 (2013).
 - ¹⁵¹K. Martin and B. Hoffman (2021). "CMake," GitHub, Version 3.21.3, <https://cmake.org/>.
 - ¹⁵²E. Martin (2020). "Ninja," GitHub, Version 1.10.2, <https://ninja-build.org/>.
 - ¹⁵³C. Sanderson and R. Curtin, "Armadillo: A template-based C++ library for linear algebra," *J. Open Source Software* **1**, 26 (2016).

- ¹⁵⁴C. Sanderson and R. Curtin, “A user-friendly hybrid sparse matrix class in C++,” in *Mathematical Software–ICMS 2018*, edited by J. H. Davenport, M. Kauers, G. Labahn, and J. Urban (Springer International Publishing, 2018), pp. 422–430.
- ¹⁵⁵V. Kirilov (2021). “doctest,” GitHub, <https://github.com/onqtam/doctest>
- ¹⁵⁶Khronos Group (2021). “OpenGL overview,” GitHub, Version 4.6, <https://github.com/KhronosGroup/OpenGL>
- ¹⁵⁷M. Woo, J. Neider, T. Davis, and D. Shreiner, *OpenGL Programming Guide: The Official Guide to Learning OpenGL, Version 1.2* (Addison-Wesley Longman Publishing Co., Inc., 1999).
- ¹⁵⁸P. Yalamanchili, U. Arshad, Z. Mohammed, P. Garigipati, P. Entschew, B. Kloppenborg, J. Malcolm, and J. Melonakos, “ArrayFire: A high performance software library for parallel computing with an easy-to-use API,” in *Proceedings of the Symposium on Applied Computing*, 2015.
- ¹⁵⁹C. Dunn (2020). “Jsoncpp,” GitHub, <https://github.com/open-source-parsers/jsoncpp>
- ¹⁶⁰F. Pezoa, J. L. Reutter, F. Suarez, M. Ugarte, and D. Vrgoč, “Foundations of json schema,” in *Proceedings of the 25th International Conference on World Wide Web* (International World Wide Web Conferences Steering Committee, 2016), pp. 263–273.
- ¹⁶¹BlueBrain (2021). “HighFive: Header-only C++ HDF5 interface,” GitHub, Version 2.3.1 <https://bluebrain.github.io/HighFive/>
- ¹⁶²The HDF Group, <https://www.hdfgroup.org/HDF5/> for “Hierarchical Data Format, Version 5,” 1997.
- ¹⁶³C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. F. del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, and T. E. Oliphant, “Array programming with NumPy,” *Nature* **585**, 357–362 (2020).
- ¹⁶⁴J. D. Hunter, “Matplotlib: A 2D graphics environment,” *Comput. Sci. Eng.* **9**, 90–95 (2007).
- ¹⁶⁵A. Collette, *Python and HDF5* (O’Reilly, 2013).
- ¹⁶⁶W. McKinney *et al.*, “Data structures for statistical computing in python,” in *Proceedings of the 9th Python in Science Conference*, 2010, Vol. 445, pp. 51–56.
- ¹⁶⁷MATLAB Version 9.11.0.1769968 (R2021b) (MathWorks Inc., 2021).
- ¹⁶⁸R. Grein, T. Löfwander, and M. Eschrig, “Inverse proximity effect and influence of disorder on triplet supercurrents in strongly spin-polarized ferromagnets,” *Phys. Rev. B* **88**, 054502 (2013).