SuperConga: an open-source framework for mesoscopic superconductivity

P. Holmwall,1,2 1) N. Wall Wennerdal,1 1) M. Håkansson, P. Stadler,1 O. Shevtsov,1 T. Löfwander,1 and M. Fogelström1
1) Department of Microtechnology and Nanoscience - MC2, Chalmers University of Technology, SE-412 96 Göteborg, Sweden
2) Department of Physics and Astronomy, Uppsala University, Box 516, S-751 20, Uppsala, Sweden

(Dated: 16 February 2023)

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 temperature. 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 GPUs. 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, as well as 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.

I. INTRODUCTION

Over the last decades there has been a tremendous improvement in 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 to 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, or make detailed spatial maps of low-lying quasiparticle states and coherences using scanning tunneling spectroscopy (STS). By fabricating nano-scale cantilevers, the intrinsic size of a superconducting island can be monitored as changes in the cantilever-oscillation frequency. Other techniques of detecting and following the superconducting state in nanoscale devices include Hall magnetometry, scanning electron microscopy (SEM), and scanning Hall probe microscopy. 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. Experimental data gathered using the above-mentioned methods give rich and complex information of the strongly correlated material system under study. To fully understand this data, one needs to compare and contrast them to theoretical predictions and modelling.

Superconductivity in metals is explained by the Bardeen-Cooper-Schrieffer (BCS) theory, Its extension and generalisation to unconventional superconductivity and superfluids makes the theory a cornerstone 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, Larkin and Ovchinnikov, and Eliashberg. 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 characterised by the Fermi energy, $E_F$, Fermi temperature, $T_F$, and inverse Fermi wave number, $1/k_F$, while the superconducting state is characterised by the superconducting order parameter, $\Delta$, transition temperature, $T_c$, and coherence length, $\xi_0$. Quasiclassical theory is then a controlled expansion in small parameters like $\Delta/E_F$, $T_c/T_F$, or $1/(\xi_0 k_F)$, which are usually of order $10^{-2} - 10^{-3}$ in metals. As a Green’s function based theory, it is very powerful and include in its most general form material effects such as impurity scattering, Fermi-liquid effects, and...
electron-phonon interaction, and non-equilibrium situations for instance imposed by external fields or potentials. At the same time, it is versatile enough that it can be adapted to realistic device sizes and geometries. In its simplest form, quasiclassical theory is equivalent to the Andreev approximation of the Bogoliubov-deGennes equation for the ballistic case.

When the mean free path, \( \ell \), due to elastic impurity scattering is much smaller than the superconducting coherence length, \( \xi_0 \), it is possible to derive diffusion equations from the more general Eilenberger-Larkin-Ovchinnikov equations. These diffusion equations, first derived by Usadel, are more easy to handle and is widely used to describe conventional diffusive s-wave samples and devices. Solution methods in the diffusive case include Nazarov’s circuit theory and finite-element methods. 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. To develop 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 take 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 into 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 it in the future can be extended to include more aspects of quasiclassical theory.

This paper is organized in the following way. In section 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 section 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 section 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 \( B_{41} \) where it becomes energetically favorable to have one vortex in a finite size disc. 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 in a grain studied experimentally by Timmermans et al. Finally, we show that in a d-wave superconducting annulus, the free-energy parabolas are conventional at high temperatures, while at lower 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 can coexist at high external magnetic fields with superconducting phase windings \( n2\pi (n \text{ integer}) \) around the annulus and integer flux \( n\Phi_0 \) threading the inner circular hole. Section VI contains a summary of the paper. The appendices 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 characterised by an order parameter

\[
\Delta(p_r, R) = \sum_n |\Gamma(R)| e^{i2\pi n(R)} \eta_c(p_r),
\]

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 \( |\Delta(R)| \) and a phase \( \chi(R) \), both in general dependent on the spatial coordinates \( R \). The phase is directly coupled to the superfluid momentum, \( p_s(R) \), and the electromagnetic gauge field, \( A(R) \), via the gauge invariant expression

\[
p_s(R) = -\frac{\hbar}{2} \nabla \chi(R) - \frac{e}{c} A(R),
\]

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 \( \eta_c(p_r) \) which depends on the Fermi momentum, \( p_F \). The index \( \Gamma \) denotes the irreducible representation of the crystallographic point group that the basis function belongs to. 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 \( \Delta_{\Gamma\delta}(p_r, R) \). 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:

\[
\Delta(p_r, R) = N_f T \sum_n \langle V(p_r, p'_r) f(p'_r, R; \epsilon_n) \rangle p'_r,
\]
where $V(p_e, p'_e)$ is the effective superconducting pairing interaction. We use the approximation that the interaction can be separated into symmetry channels of the crystal point group

$$V(p_e, p'_e) = \sum_{\Gamma} V_\Gamma \eta^e(p_e) \eta^{e'}(p'_e),$$  \hspace{1cm} (4)

where $V_\Gamma$ is the pairing strength in the symmetry channel $\Gamma$. In this case, the dependencies on position and momentum direction of the order parameter separate as in Eq. (1). The angle bracket $\langle \ldots \rangle_{p_e}$ denotes a Fermi-surface average, which in 2D is a line integral around the Fermi surface according to [3] and Fermi velocity respectively and $N_e$ the total normal-state density of states per spin at the Fermi level is

$$N_e = \frac{1}{N_\Gamma \langle p_e \rangle} \int_{FS} \frac{dp_e}{(2\pi \hbar)^2 |v_\perp(p_e)|} \langle \ldots \rangle_{p_e},$$ \hspace{1cm} (5)

where the factor 2 accounts for spin degeneracy. The Fermi velocity at $p_e$ on the Fermi surface is $v_\perp(p_e) = \nabla p \varepsilon(p)|_{p=p_e}$, where $\varepsilon(p)$ is the dispersion.

The two functions $g(p_e, R; \varepsilon_n)$ and $f(p_e, R; \varepsilon_n)$ appearing in Eq. (3) and Eq. (8) are components of the quasiclassical Green’s function, or propagator,

$$\hat{g}(p_e, R; \varepsilon) = \left( \begin{array}{cc} g(p_e, R; \varepsilon) & f(p_e, R; \varepsilon) \\ -f(p_e, R; \varepsilon) & g(p_e, R; \varepsilon) \end{array} \right).$$ \hspace{1cm} (9)

It is a 2x2 matrix in electron-hole (Nambu) space, as indicated by the “-”-symbol. The propagator is determined from the quasiclassical counterpart of the Gorkov-Dyson equation, the transport-like Eilenberger equation:

$$0 = iv_\perp \varepsilon v_\perp \hat{g}(p_e, R; \varepsilon) + \left[ \varepsilon + \frac{\pi}{c} v_\perp(p_e) \cdot A(R) \hat{\tau}_3 - \hat{\Delta}(p_e, R), \hat{g}(p_e, R; \varepsilon) \right],$$ \hspace{1cm} (10)

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

$$\hat{g}^2(p_e, R; \varepsilon) = -\pi^2 \mathbb{1},$$ \hspace{1cm} (11)

where $\mathbb{1}$ is the identity matrix.

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

$$\hat{\Delta}(p_e, R) = \left( \begin{array}{cc} 0 & \hat{\Delta}(p_e, R) \\ -\hat{\Delta}(p_e, R) & 0 \end{array} \right).$$ \hspace{1cm} (12)

There is some redundancy in the parametrisation of the propagator, and the following symmetry:

$$\hat{x}(p_e, R; \varepsilon) = x(-p_e, R; -\varepsilon^*),$$ \hspace{1cm} (13)

between tilded ($\hat{}$) and un-tilded ($x$) quantities holds. The propagator may be evaluated at imaginary frequencies $\varepsilon \rightarrow i\varepsilon = i\pi T (2n+1)$ 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, $\delta$ 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

$$N(p_e, R; \varepsilon) = -\frac{N_e}{\pi} \text{Im} \hat{g}^R(p_e, R; \varepsilon),$$ \hspace{1cm} (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 defined as the transition temperature $T_c$, namely $2\pi k_B T_c$, and a natural length scale, the coherence length $\xi_0 = \hbar v_\perp / 2\pi k_B T_c$. We will use the natural units $\hbar = k_B = 1$. The normal state density of states at the Fermi level $N_e$ 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 $\lambda_0$, defined as

$$\frac{1}{\lambda_0^2} = \frac{4\pi^2}{c^2 v_\perp^2 N_e}. \hspace{1cm} (15)$$

We use the Ginzburg-Landau parameter $\kappa$, 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, $B = \nabla \times A$, 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 in units of $j_0 = 2\pi k_B T_c |v_\perp N_e|$. A summary of these units and natural scales are given in Table [11] in Appendix [F].

The quasiclassical theory is a conserving theory in the sense that the equations for $\hat{g}$, $\hat{\Delta}$, and $A$ above may be derived as stationarity conditions from the Luttinger-Ward free-energy functional. For the theory to be conserving, the stationarity equations are to be solved to self consistency. At convergence, we solve Eq. (10) with $\Delta(p_e, R)$ and $A(R)$, recalculate Eq. (3) and Eq. (7), and get back the same $\Delta(p_e, R)$ and $A(R)$ 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. This functional always gives a measure of the free energy with respect to the normal state.\footnote{35-36,64} Over to the left hand side, and using the fact that the basis functions are orthonormal, \( \langle \eta_i'(p) \eta_j(p) \rangle_{pV} = \delta_{ij} \), we can divide down the resulting prefactor of \( \Delta_{\Gamma}(\mathbf{R}) \), obtaining the gap equation in a fix point form:

\[
\Delta_{\Gamma}(\mathbf{R}) \ln \frac{T}{T_c} = T \sum_n \langle \eta_i'(p) \rangle \left[ f(p, R; \varepsilon_n) - \frac{\pi \Delta(p, R)}{|\varepsilon_n|} \right]_{pV},
\]

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

\[
\hat{\lambda}(p, R; \varepsilon) \text{ in Eq. (10), subject to the normalization condition in Eq. (11). The gradient term } i \nu \delta \hat{\lambda}(p, R; \varepsilon) \text{ couples the momentum direction and spatial coordinates. Quasiparticle trajectories are then defined by this directional derivative and we may introduce a trajectory coordinate } s \text{ as }
\]

\[
\dot{R}(s) = R_{\text{min}} + s \hat{v}_p,
\]

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

\[
\mathcal{I}(\mathbf{R}) = \pi N_c k_B T \sum_n \left\langle \frac{\Delta(p, R)}{\pi} f(p, R; \varepsilon_n) + \Delta(p, R) \tilde{f}(p, R; \varepsilon_n) \right\rangle_{pV}.
\]
tions satisfy a set of Riccati equations:

\[ \dot{\gamma} = -i \frac{\pi}{1 + \gamma^2} \begin{pmatrix} 1 - \gamma^2 & 2 \gamma \\ 2 \gamma & 1 + \gamma^2 \end{pmatrix}, \tag{24} \]

FIG. 1. (a) The system domain is denoted \( D \), and has a boundary \( \partial D \). A quasiparticle trajectory parameterized by the coordinate \( s \) starts at one boundary point \( R_{\text{min}} \) and ends at another boundary point \( R_{\text{max}} \). 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 the following sections for more details.

ducting state. The quasiclassical Green’s function can be written in terms of them as

\[ \hat{g} = \frac{-i \pi}{1 + \gamma^2} \begin{pmatrix} 1 - \gamma^2 & 2 \gamma \\ 2 \gamma & 1 + \gamma^2 \end{pmatrix}, \tag{24} \]

where we for brevity dropped the explicit dependences on \( p_F, R, \) and \( \epsilon \). The normalization of the Green’s function in Eq. (11) is now automatically satisfied. The coherence functions satisfy a set of Riccati equations:

\[ iv_\gamma \cdot \nabla \gamma + 2(\epsilon + \frac{\pi}{2} v_\gamma \cdot A) \gamma + \hat{A} \gamma^2 + \Delta = 0, \tag{25} \]

\[ iv_\gamma \cdot \nabla \hat{\gamma} - 2(\epsilon + \frac{\pi}{2} v_\gamma \cdot A) \hat{\gamma} + \Delta \hat{\gamma}^2 + \hat{\Delta} = 0. \tag{26} \]

The equation for \( \gamma(p_F, R, \epsilon) \) is stable to integrate in the direction parallel to \( v_\gamma \), while the equation for \( \hat{\gamma}(p_F, R, \epsilon) \) is stable to be integrated in the opposite direction. After introducing the trajectory coordinate, we may let \( v_\gamma \cdot \nabla \gamma \rightarrow v_\gamma \partial_s \gamma \).

### 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 B) along trajectories through the grain. An example of a trajectory is shown in Fig. 1. Given an initial boundary value, \( \gamma(R_{\text{min}}) \), we solve Eq. (25) until we reach \( R_{\text{max}} \). Similarly, given an initial boundary value, \( \hat{\gamma}(R_{\text{max}}) \), we solve Eq. (26) until we reach \( R_{\text{min}} \).

Clearly, the boundaries are special points in that we need a starting value for \( \gamma(R_{\text{min}}) \) and we obtain after stepping along the trajectory an end value \( \gamma(v_N) \) 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 compute the boundary values by repeatedly stepping along all trajectories and apply the boundary condition. Since the order parameter \( \Delta(R) \) 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 to 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: \( \Delta(R) = \Delta(x, y) \rightarrow \Delta(i, j) \) and \( A(R) = A(x, y) \rightarrow A(i, j) \), 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 artefacts. 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 \( \varphi_k \) on the Fermi surface, we obtain a Fermi velocity \( v_\gamma(\varphi_k) \), 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 \( \varphi_k \) 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 \( j(i, j) \), and also the vector potential \( A(i, j) \) can be updated through Eqs. (3)-(7).

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 rota-
FIG. 2. Visualization of the implementation and main algorithm. (a)–(b) The Fermi surface is parametrized by the discrete angle \( \phi_k \) and the Fermi velocity \( v_k(\phi_k) \), which together define quasiparticle trajectories in real space. As in Fig. 1, each trajectory has a well-defined start point \( R_{\text{min}} \) and end point \( R_{\text{max}} \), 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.

Algorithm 1 A sketch of the SuperConga main loop

Define domain \( \mathcal{D} \) and input parameters
Initial \( \Delta \) and \( A_{\text{ind}} \)
Initial \( T_0 \) and \( \tilde{T}_0 \)

while not converged do
for all \( p_i, e_i \) do
Rotate \( \Delta \) and \( A \) so that \( \vec{v} \parallel \vec{v}_i \)
Compute \( \gamma \) and \( \rho \) along the \( y \)-axis
Update \( T_{iD} \) and \( \tilde{T}_{iD} \)
Update \( \Delta, j, \) and \( \Omega \)
Compute \( A_{\text{ind}} \)
Check convergence
\( \Delta, A_{\text{ind}} \leftarrow \) Accelerator
Update \( T_{iD} \) and \( \tilde{T}_{iD} \)

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 \( \Delta \phi \) in momentum space, and an energy cut-off 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 Ozaki has been implemented. In addition, a convergence criterion for the self-consistency must be set. The global error is defined as

\[
\varepsilon_G = \frac{\| \Delta^i - \Delta^{i-1} \|_p}{\| \Delta^{i-1} \|_p},
\]  

(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 \( \varepsilon_G \) 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, \( \{ \Delta^i, \Delta^{i-1}, \Delta^{i-2}, \ldots \} \), and make a more educated guess for \( \Delta^{i+1} \). 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. But 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. 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 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, 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, disc- 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. discs 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\(^{31}\) "small heavy sphere", a variant of the Barzilai-Borwein method\(^{32}\) and CongAcc which is an adaptive method developed for SuperConga. See App.\(^{33}\) for details and comparisons.

**Order parameter:** Class for setting up the order parameter, possibly with multiple components like \(d + i s\) 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\(^{34}\).

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 is 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 are 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 the next section 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 an example). The user can control what to plot, in what order, and which settings to use for e.g. colormaps 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 is read and visualized in an interactive plot (see Fig.\(^{5}\) for an example).

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

The eight 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.

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. One of the examples will now be shown, namely the 
simulation of a single Abrikosov vortex in a conventional superconducting disc. 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.

The simulation configuration for the vortex example can be found in examples/swave_disc_vortex/ 
in the SuperConga main folder. Appendix 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 $\Phi_{ext} = 1.5\Phi_0$, the 
penetration depth $\lambda = 5\xi_0$, and an s-wave order parameter with a phase 
winding of $-2\pi$ 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 guide), 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 disc.

```
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 specifies 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 towards speed 
rather than producing production-ready plots. For the latter, 
we instead recommend to visualize the fully converged re-

```
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. Various properties of the plot can be controlled via command-line 
arguments (as described by the help message), like fonts, col-
```
 or maps, which quantities are plotted, and if saving the plot di-
rectly 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.
FIG. 4. Plot generated with the SuperConga “plot-simulation” command, visualizing the example with a single Abrikosov vortex in a conventional superconducting disc, by following Listing 2 then 3. Here, $T = 0.5T_c$, $B_{ext} = 1.5\Phi_0/A$, 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.

FIG. 5. Screenshot of the SuperConga interactive spectroscopy command, “plot-postprocess”. The plots show the local density of states (LDOS), $N(R, \varepsilon)$, for a single vortex at the center of a conventional superconducting disc, after following Listings 2, 4, and then 6 in order. Here, $T = 0.5T_c$, $B_{ext} = 1.5\Phi_0/A$, 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 versus 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).
Listing 5. Configuration file used for computing (post-processing) the LDOS: postprocess_config.json in examples/swave_disc_vortex/.

```json
{
    "spectroscopy": {
        "energy_max": 0.5,
        "energy_min": 0.0,
        "energy_broadening": 0.008,
        "num_energies": 128
    },
    "num_iterations_min": 0,
    "num_iterations_max": 1000,
    "num_iterations_burnin": -1,
    "num_energies_per_block": 32,
    "num_fermi_momenta": 720,
    "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 is 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:

```
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 versus coordinates), and which point in space to plot the LDOS versus 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 \( \kappa \), throughout the whole Type-II range \( \kappa \in [1, \infty) \):

Listing 8. Performing vortex spectroscopy.

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

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

```
python examples/parameter_sweeps/plot_vortex_kappa_sweep.py
```

Figure 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 disc 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. and 5). Different lines correspond to different values of \( \kappa \), 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 \( R/\lambda \) will increase the region of full screening.

So far, the demonstration has dealt with a superconducting disc. SuperConga allows the user to specify more general composite geometries by successively adding and removing discs and polygons. Both regular and free-form polygons are implemented, enabling quite intricate systems that e.g. have holes, are multiply connected, or even disconnected but coupled via induction. For example, Fig. 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 disc from the central region:

Listing 9. Example setup of a composite geometry.

```json
"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": {
```

This file specifies e.g. which energies and resolution to use in the LDOS calculation, and where the order-parameter data is 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:

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

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 \( \kappa \), throughout the whole Type-II range \( \kappa \in [1, \infty) \):

Listing 8. Performing vortex spectroscopy.

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

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

```
python examples/parameter_sweeps/plot_vortex_kappa_sweep.py
```

Figure 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 disc 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. and 5). Different lines correspond to different values of \( \kappa \), 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 \( R/\lambda \) will increase the region of full screening.

So far, the demonstration has dealt with a superconducting disc. SuperConga allows the user to specify more general composite geometries by successively adding and removing discs and polygons. Both regular and free-form polygons are implemented, enabling quite intricate systems that e.g. have holes, are multiply connected, or even disconnected but coupled via induction. For example, Fig. 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 disc from the central region:

Listing 9. Example setup of a composite geometry.

```json
"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": {
```
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 $\xi$, of edges, rotation (in units of 2 and radius, a regular polygon by center coordinates, number whichever origin they prefer, as it does not influence 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 $\xi$, of edges, rotation (in units of 2 and radius, a regular polygon by center coordinates, number whichever origin they prefer, as it does not influence the physics. The disc is uniquely defined via its center coordinates and radius, a regular polygon by center coordinates, number of edges, rotation (in units of $2\pi$), and side-length. For free-range polygons, a list of x- and y-coordinates for the vertices are specified (in counter-clockwise 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. 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. 2 plots the zero-energy LDOS in nanoscale grains of various geometries exposed to high external fields. Vortex cores show up as dot-like patterns, 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 discs, squares, triangles and pentagons. 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 as well as vortex-antivortex pairs. Careful treatment and use of the adaptive convergence accelerators show, however, that these structures are generally unstable in non-heterostructures that are sufficiently clean, where they 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. Second, we study a superconducting disc 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 and in particular to that of Pearl considering superconducting discs. For the cases of zero or one vortex at the center of the disc, 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. on spectroscopy on superconducting nanostructures assembled by small squares of Mo$_7$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 $\Delta(p_v, R) = \Delta(R)\eta(p_v)\exp[i\chi(R)]$, which in general is complex valued. The order-parameter matrix can then be decomposed as

$$\tilde{\Delta}(p_v, R) = \tilde{U}(\chi)\tilde{\Delta}_0(p_v, R)\tilde{U}^\dagger(\chi),$$

with the transformation matrix

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

and $\Delta_0(R)$ a purely real amplitude (i.e. $\chi_0(R) \equiv 0$ while the basis function $\eta(p_v)$ 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

$$i\nu_0 \nabla_R \hat{g} + [(i\epsilon_n + \xi \nu_0 \cdot \mathbf{A}) \hat{\tau}_3 - \hat{\Delta}, \hat{g}] = 0,$$

(30)

with $\hat{\Delta} = \hat{U} \Delta_0 \hat{U}^\dagger$, then $\hat{g}_0 = \hat{U}^\dagger \hat{g}_0 \hat{U}$ is a solution to

$$i\nu_0 \nabla_R \hat{g}_0 + [(i\epsilon_n + \xi \nu_0 \cdot \mathbf{A} - \frac{1}{2} \xi \nu_0 \cdot \nabla \chi) \hat{\tau}_3 - \hat{\Delta}_0, \hat{g}_0] = 0.$$

(31)

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

$$i\nu_0 \nabla_R \hat{g}_0 + [(i\epsilon_n - \nu_0 \cdot \mathbf{p}_s) \hat{\tau}_3 - \hat{\Delta}_0, \hat{g}_0] = 0,$$

(32)

with the purely real order-parameter matrix.

Let us look at the linear response to a small and spatially homogeneous superfluid momentum, $p_\parallel \ll \Delta/\nu_0$. To do this, we write the propagator as a perturbation expansion

$$\hat{g} = \hat{g}_0 + \delta \hat{g} + O[p_\parallel^2].$$

Using the normalisation condition on the propagator $\hat{g}^2 = -\pi^2 \hat{\nu}$ we get

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

Since $\hat{g}_0 \propto i\epsilon_n \hat{\tau}_3 - \hat{\Delta}_0(p_\parallel)$, we get via the Eilenberger equation

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 $T = 0.1 T_c$, and the grain side-lengths ($S$) was chosen such that the area is the same as that of the disc, $A = \pi R^2$. Colors indicate the LDOS at zero energy, with localised Caroli-de Gennes-Matricon states in each vortex core. To give the vortex cores a visible extent, a smearing of $\delta = 0.008 \cdot 2\pi k_T$ was used, and an arbitrary cutoff of $2N_c$ 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 from high fluxes to low fluxes again (right-most column). For each new value of the flux, the converged results from the previous simulation is used as a start guess. The 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 disc (blue). Full lines are for sweeping towards higher fields and dashed for sweeps in the opposite direction. This asymmetry in the free energy versus applied field, between the two sweep directions, can be interpreted as a Bean-Livingston barrier that makes vortex entry energetically harder than vortex exit. (d) LDOS at a finite energy $\epsilon = 0.9 k_F T$, at external flux $\Phi_{ext} = 55\Phi_0$. 


the solution
\[
\delta \tilde{g}(p_r; \epsilon_n) = \pi \frac{v_r \cdot p_r}{\Lambda_n^2(p_r)} \left( \frac{\vert \Delta(p_r) \vert^2 - i\epsilon_n \Delta(p_r)}{i\epsilon_n \Delta'(p_r) - \vert \Delta(p_r) \vert^2} \right),
\]
(33)
defining \( \Lambda_n(p_r) = \sqrt{\vert \Delta(p_r) \vert^2 + \epsilon_n^2} \). From the form of \( \delta \tilde{g} \) we can directly read out that the linear correction to the anomalous Green’s function component, \( f(p_r; \epsilon_n) \), is odd in frequency and will not give a correction in leading order in \( p_i \) to the order parameter \( \Delta(p_r) \). The diagonal part of \( \delta \tilde{g} \) allows us to calculate the current to linear order in \( p_i \) as
\[
\delta j = 2eN_c \langle v_r \cdot (p_r) Y_{3/2} (p_r) v_r \rangle p_r / p_F,
\]
(34)
with the angle-dependent Yoshida function
\[
Y_{3/2}(p_r) = \pi \frac{T}{\mathcal{A}} \sum_n \frac{\vert \Delta(p_r) \vert^2}{\Lambda_n(p_r)},
\]
Equation (34) is usually written in the well-known form
\[
\tilde{J} = e \langle \tilde{\rho}_s(p_r) \cdot p_r / p_F, \n\]
defining
\[
\tilde{\rho}_s(p_r) = 2N_c Y_{3/2}(p_r) v_r \tau^T(p_r),
\]
(36)
the angle-dependent superfluid-density, or superfluid stiffness tensor \( \mathcal{A} \).

As a final step we write the change in the free energy in linear response as
\[
\Omega = \Omega_{\text{bulk}} + \delta \Omega_{\text{kin}} + \delta \Omega_{\text{magn}},
\]
(37)
where
\[
\Omega_{\text{bulk}} = AN_c \left( \frac{\vert \Delta(p_r) \vert^2}{\mathcal{A}_n(p_r)} \ln \frac{T}{T_c} \right) T + \pi T \sum_n \frac{\vert \Delta(p_r) \vert^4}{\mathcal{A}_n(p_r) + \epsilon_n^2} p_r
\]
(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
\[
\delta \Omega_{\text{kin}} = \frac{1}{2} \int dR \left\langle \tau^T(R) \tilde{\rho}_s(p_r) p_r / p_F \right\rangle,
\]
(39)
and a magnetic contribution
\[
\delta \Omega_{\text{magn}} = \frac{1}{8\pi} \int dR \mathcal{A}(R) - B_{\text{ext}},
\]
(40)
where \( B_{\text{ext}} \) 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 \( R_\circ \) (\( R_\times \)), as illustrated in Fig. 8c. The area of the annulus is \( \mathcal{A} = \pi (R_\circ^2 - R_\times^2) \). Let us first consider

![FIG. 8. The magnetic flux dependence of the free energy of an s-wave (a) and a d-wave (b) superconducting annulus using the solenoid gauge Eq. (41). The outer (inner) radius of the annulus is \( R_\circ = 25 R_\times \) (\( R_\circ = 10 R_\times \)), and the temperature is set to \( T = 0.5 T_c \). 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 is the free energy given by Eq. (38) and Eq. (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 \( n = 1 \).](image)

the following text-book vector potential \( \mathcal{A}_{\text{ext}}(R) \) used to illustrate the Aharonov-Bohm effect
\[
\mathcal{A}_{\text{ext}}(R) = \Phi_{\text{ext}} \phi \tau^T(R) \mathcal{A}(R) - B_{\text{ext}},
\]
(41)
where \( \phi \) is the unit vector along the azimuthal direction and \( \Phi_{\text{ext}} \) is an externally applied magnetic flux threading only the hole. We call this particular gauge the solenoid gauge. It can be seen as a mathematical idealisation 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

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 \( \chi \) in Eq. \( (28) \) must be chosen so that the total winding going around the void is a multiple \( n \) of \( 2\pi \), and we can write \( \chi = n\phi \) where \( n \) can take any integer value. If we now look at Eq. \( (2) \) we get, reinstating \( h \) and \( c 

\[
p_c = \frac{\hbar}{2} \nabla \chi - \frac{e}{c} \frac{\Phi_{\text{ext}}}{\hat{n}} = \frac{\hbar}{2R} \left( n + \frac{\Phi_{\text{ext}}}{\Phi_0} \right) \phi.
\]

We see that \( p_c \) 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 \( \Phi_0 \). 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 \( \bar{g}, \bar{A} \), at a given magnetic flux \( \Phi_{\text{ext}} \), threading the hole, we can, via an appropriate gauge transformation \( \bar{U} \), construct a solution at any other external flux that differ from \( \Phi_{\text{ext}} \) by an integer number of flux quanta. This setup nicely demonstrates the fundamental concept of flux quantization. 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 \( \Delta(p) = \Delta \), where \( \Delta \) is a complex scalar quantity and the basis function \( \eta(p) = 1 \). 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

\[
\delta \Omega_{\text{kin}} \left( \frac{2\pi T}{\eta} \right)^2 \frac{\hbar}{N_F} = \frac{\pi}{4} \xi_0^2 Y_{3/2}(T) \ln \left( \frac{\mathcal{R}_c}{\mathcal{R}_s} \right) \left( n + \frac{\Phi_{\text{ext}}}{\Phi_0} \right)^2,
\]

after integrating over the superconducting area and setting

\[
Y_{3/2}(T) = (\hat{\theta}_y(p\hat{v})) Y_{3/2}(p\hat{v}; T) \hat{\theta}_y(p\hat{v}) \cdot \hat{\theta}_y(p\hat{v})
\]

where \( \hat{\theta}_y(p\hat{v}) \) and \( \hat{\theta}_y(p\hat{v}) \) 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. \( 9(a) \), 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 \( \Omega_{\text{bulk}} \) when the external flux is close to \( (n + \frac{1}{2})\Phi_0 \). This is a manifestation of the Little-Parks effect \( [33,129,124] \) where one can induce a field-modulation of the critical temperature of a superconducting film enclosing a hole.

---

**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 \( \mathcal{R}_c > 25\xi_0 \) (\( \mathcal{R}_s < 10\xi_0 \)). 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 \( 2\pi \) 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)–(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 \( \lesssim 0.1J_0 \). The orange diamond in panel (a) is a special meta-stable state with no spontaneous currents.
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, $p_{\perp}$, via the basis function, either $d_{2-s}^{\alpha}$ or $d_{1}^{\alpha}$ 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. (22).

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. 3(b), neither $\Omega_{\text{bulk}}(T)$ nor $\delta\Omega_{\text{cin}}(T, n, \Phi_{\text{ext}})$, 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-wave superconducting annulus is shown in Fig. 3(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 localised in a region of a few coherence lengths width at the surface. These states modify low-temperature properties of the superconductor. A disc-shaped unconventional superconductor was studied by Suzuki and Asano finding a paramagnetic instability at low temperatures as a response to a small magnetic field. We find similar paramagnetic behavior for the annular superconductor at low temperatures and relate the effects to spontaneous generation of phase gradients.

In Fig. 2 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 $2\pi$ around the annulus. Below $T \approx 0.2T_c$, the existence of zero-energy Andreev states start to drive the superconductor to generate a finite $p_{\perp}$ by spontaneous phase gradients. This finite superfluid momentum gives rise to a Doppler shift, $v_{\perp} \cdot p_{\perp}$, 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. 2 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. 2.

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 $B_{\text{ext}} = B_{\text{ext}} \hat{z}$ is applied in a homogeneous fashion and penetrates also the superconductor. We incorporate this case through a symmetrical gauge $A_{\text{ext}}(R) = \frac{1}{2}B_{\text{ext}}(-\gamma \hat{x} + x \hat{y})$ subject to the condition $\nabla \cdot A = 0$.

Let us study the London-Maxwell equation for the vector potential $A$ in the symmetrical gauge

$$-\nabla^2 A = \frac{4\pi e}{c} \langle \hat{n}_{p_{\perp}}(p_{\perp}) \rangle_{p_{\perp}} = -\frac{1}{\lambda^2(T)} \left( \Phi_0 \right) \frac{\partial}{\partial n} \Phi + A \right),$$

(45)

for the same annulus geometry as in the previous section. We allow for a phase winding $2\pi n$ of the order parameter going around the annulus, and include screening of the externally applied field. The temperature-dependent penetration depth $\lambda(T)$ entering Eq. (45) is defined as

$$\frac{1}{\lambda^2(T)} = \frac{1}{\Phi_0^2} Y_{3/2}(\Omega(\Omega_{T})),$$

(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

$$\alpha = \frac{\Phi_{\text{ext}}}{\Phi_0},$$

(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 $A = A(\alpha)\hat{\phi}$, the general solution, in the external magnetic field $B_{\text{ext}} = \alpha \Phi_0 \hat{z}$, may be written as

$$A(\alpha) = \Phi_0 \lambda \left[ a \lambda_1 \left( \frac{R}{\lambda} \right) + b \lambda_1 \left( \frac{R}{\lambda} \right) \right]$$

$$+ \frac{\alpha \Phi_0}{2\pi} R \in [R_{<}, R_{>}]$$

(48)

$$A(\alpha) = \beta \left( \frac{\Phi_0 \bar{R}}{\lambda_0} \right)^2 R \in [0, R_{<})$$

(49)

where $A = \pi(\mathcal{R}_{>}^2 - \mathcal{R}_{<}^2)$ and $A_{\lambda} = \pi\mathcal{R}_{<}^2$ 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 $(a, b, \beta)$ are determined so that $A$ is bounded for $R < R_{<}$, and continuous at $R = R_{<}$ and that the magnetic field $B = \nabla \times A$ is continuous at $R = R_{>}$. We state them for completeness

$$a = \frac{1}{M} \left( \frac{K_{<}}{\lambda^2} \alpha + \frac{K_{0}^{<}}{\lambda_0^2} n \right),$$

(50)

$$b = \frac{1}{M} \left( \frac{I_{<}}{\lambda^2} \alpha + \frac{I_{0}^{<}}{\lambda_0^2} n \right),$$

(51)

$$\beta = \frac{1}{M} \left( \frac{2\pi \mathcal{R}_{>}}{\mathcal{R}_{<}} \alpha - \frac{I_{0}^{<}}{\lambda_0^2} \right) \langle n \rangle$$

(52)

with $M = I_{0}^{<} K_{2}^{<} - I_{2}^{<} K_{0}^{<}$ and $I_{\mu}^{<} = I_{\mu}^{<} (R_{<}^2)$ and the same for $K_{\mu}^{<}$. The corresponding magnetic field, $B = B(\alpha) \hat{z}$ in and
of the free-energy correction due to an external field (40), and the results compare well with results obtained by perconducting annulus can thus be evaluated using Eqs. (39)–(43). 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 \( \kappa / \xi_0 \), which is (a) \( \kappa = 100 \), (b) \( \kappa = 20 \), and (c) \( \kappa = 2 \).

The free energy for an \( s \)-wave superconductor in an annulus, with an outer (inner) radius \( R_\text{\textgreater} = 25 \xi_0 \) \((R_\text{\textless} = 10 \xi_0)\), in an external magnetic field in symmetric gauge, at \( T = 0.2 T_c \). The magnetic flux-density is given by \( B_\text{\textgreater} = \Phi_{\text{ext}} / \mathcal{A} \), where \( \mathcal{A} = \pi (R_\text{\textgreater}^2 - R_\text{\textless}^2) \). 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 \( \kappa / \xi_0 \), which is (a) \( \kappa = 100 \), (b) \( \kappa = 20 \), and (c) \( \kappa = 2 \).

FIG. 10. The free energy for an \( s \)-wave superconducting annulus: (a) \( n = 0 \), (b) \( n = -1 \), (c) \( n = -2 \), and (d) \( n = -3 \).

FIG. 11. The free energy as function of external magnetic field for a \( d \)-wave superconducting annulus, with an outer (inner) radius \( R_\text{\textgreater} = 25 \xi_0 \) \((R_\text{\textless} = 10 \xi_0)\), 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 \( 2 \pi \) around the annulus. From top to bottom we lower the temperature from \( T = 0.3 T_c \) down to \( T = 0.1 T_c \) as in Fig. 9. The dotted vertical lines are a guide to the eye.

- An \( s \)-wave superconducting annulus: For the \( s \)-wave annulus, the equations (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. (49)–(50), 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 \( \alpha \) and a possible phase winding \( n \) will be

\[
\delta \Omega_{\text{kin}} + \delta \Omega_{\text{magn}} = \delta \Omega_0 (\omega_\chi n^2 + (\omega_\theta n - \alpha)^2).
\]

We determine the factors \( \delta \Omega_0 \), \( \omega_\chi \) and \( \omega_\theta \) 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 \( T = 0.2 T_c \) 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 \( \lambda \) 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 cut-off in the frequency sums.

- A \( d \)-wave superconducting annulus: For the \( d \)-wave superconductor we solve Eqs. (71)–(80) and Eq. (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 \( T \approx 0.2 T_c \) 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, signalling that it is energetically favorable to form spontaneous currents as discussed above using the solenoid gauge.

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

\[
m = \frac{1}{2} \int_V \mathbf{d} \mathbf{R} \times j_{\text{3D}}(\mathbf{R}).
\]

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

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

i.e. the magnetic moment is given as

\[
m = \frac{N d}{2} \int_A \mathbf{d} \mathbf{R} \times j(\mathbf{R}),
\]
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 \( m_0 = j_0 \bar{\phi}_0 \mathcal{V} \) where

\[
    j_0 \bar{\phi}_0 = \frac{c^2 B_0}{4\pi \kappa^2} \text{ (cgs)} = \frac{1}{\mu_0} \frac{B_0}{\kappa^2} \text{ (SI)},
\]

with \( B_0 = \Phi_0 / (\pi \bar{\xi}_0^2) \) and \( \mathcal{V} = N_l d A \). 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 \( \mu_B = |e|/2m_e \) is the Bohr magneton and \( n = v_F p_F N_e \) is the number density of charge carriers. Inserting numerical values for \( \Phi_0 \) and \( \mu_0 \) and setting the length scale to \( n m_s \) for \( \lambda_0 \) and \( \mathcal{V} \) gives

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

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

\[
    m(\alpha, n) = m_0 (d_\alpha d + d_n n) \mathbf{\hat{z}}, \tag{59}
\]

where

\[
    d_\alpha = \frac{\pi \lambda_0^2}{A} \left( I_\alpha^2 K_\alpha^2 - I_\alpha^2 K_\alpha^2 \right), \tag{60}
\]

and

\[
    d_n = \frac{\pi \lambda_0^2}{A} \left( 2\pi \lambda_0^2 - M A_0 \right)^{-1} \tag{61}
\]

How \( d_\alpha \) 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).

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. 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. \( R_\infty = 25 \xi_0 \) and \( R_\infty = 10 \xi_0 \), 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 disc

As another example we consider an s-wave superconductor in the form of a 2D-disc, with radius \( \mathcal{R} \), in an external magnetic field. We will allow for arbitrary strength of the applied magnetic field so that multiple vortices can enter the disc. The general solution can and will break the continuous rotational symmetry of the disc 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 favourable for the superconducting material to host...
one vortex 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 disc. 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.[129] 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. $\xi_0$, $\lambda_0$, 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 disc with a radius fixed to $25\xi_0$ at a temperature set to $0.2T_c$ if not otherwise stated. The analytic results 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 A. Fetter in Ref.[129] for a thin-film superconducting disc. 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

$$p_s(R) = \frac{\hbar}{2} \left( \frac{n}{R} + \alpha \frac{R}{\Omega^2} \right) \hat{\phi}. \tag{60}$$

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

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

$$n = -1 : (\Omega(\alpha) - \Omega_{\text{bulk}})/A = \delta \phi_{\text{kin}} (\epsilon_{\text{core}} - 4\alpha + \alpha^2). \tag{63}$$

where $\Omega_{\text{bulk}}$ is the bulk free-energy contribution Eq. (38) and

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

gives the free-energy contribution from the induced supercurrent due to the combined effect of the external field and to the phase-winding of $2\pi R$ around the vortex core. The term $\epsilon_{\text{core}}$ in principle consists of two terms. The first is proportional to $\ln(R/R_c)$, with the vortex-core radius $R_c$, due to the $1/R$ dependence of $p_s$. The second, $\delta \phi_{\text{kin}}$, is due to the gradient-energy terms that come form 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 $\xi_0$ in the calculations. In Fig. 15 we show calculations made with SuperConga for
\[ \lambda_0 = \infty \text{.} \]

As seen, numerical results fall straight on the theoretical prediction for \( (\Omega(\alpha) - \Omega_{\text{bulk}})/\alpha \) given by Eqs. (62)-(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 disc at this temperature \( B_{c1} \) occurs at \( \alpha \approx 3.8 \).

When the penetration depth is comparable to or smaller than the size of the disc we need to analyze Eqs. (49)-(53) in the limit that \( R_c \to 0 \). The result is

\[
A_0(R) = \lambda B_{\text{eff}}(\alpha, n) I_1 \left( \frac{R}{\lambda} \right) + \frac{\Phi_0}{2\pi} \left[ \frac{R}{\lambda} - \frac{1}{\lambda} K_1 \left( \frac{R}{\lambda} \right) \right], \tag{65}
\]

\[
B_c(R) = B_{\text{eff}}(\alpha, n) [I_0 \left( \frac{R}{\lambda} \right) + \frac{\Phi_0}{2\pi\lambda^2 K_0} \frac{R}{\lambda}]. \tag{66}
\]

with

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

The functional form of Eqs. (65)-(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 \( \lambda \) in the following way

\[
\delta \Omega_{\text{kin}}(\lambda) = \left( k(R) + b(R) \right) \left( \frac{\xi_0}{\lambda} \right)^2 Y_{3/2}(T), \tag{68}
\]

with

\[
k(x) = \frac{1}{x^2} I_1^2(x) + 2 I_0(x)I_1(x) - I_0^2(x), \tag{69}
\]

\[
b(x) = \frac{2 I_0^2(x) - \frac{4}{x} I_0(x)I_1(x) - I_1^2(x)}{2 I_0^2(x)}. \tag{70}
\]

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

The vector potential, Eq. (55), is finite in the origin. This is not the case for the magnetic field, Eq. (66), which diverges as \( \ln(R) \) when \( R \to 0 \) in the origin. 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)-(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. \( \delta \omega_{\text{kin}}(\lambda) \), 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 \( \xi \) and \( \lambda \) gives that close to \( T_c \), in the regime of validity of Ginzburg-Landau theory, the physical size of the disc 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 sets to be less than 5% of the total free energy. Instead, the free energy mainly depends on how well the supercurrents are screened, i.e. focus is primarily on the ratio \( \lambda_0/R \). 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 disc, and a marginal type II superconductor with \( \kappa = 2.0 \) with \( \lambda_0 \ll R \). 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 \( \delta \omega_{\text{kin}}(\lambda_\text{core}(\kappa = 2) \approx 0.5\delta \omega_{\text{kin}}(\kappa = 20) \).

In Fig. 17 we show a large set of computed free energies as a function of applied external magnetic field. The magnetic field \( B = \alpha\Phi_0/A \) 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 disc becomes normal. Starting with the first critical field, \( B_{c1} \), we lift results from literature.\[13, 129\]

For a superconducting disc of radius \( R \) where \( \lambda_0/R \sim 1 \) the first critical field is given as

\[
B_{c1}^\text{disc} = \frac{\Phi_0}{A} \left( \ln \frac{R}{R_c} + \varepsilon_0 \right), \tag{71}
\]

with \( \varepsilon_0 \) due to the gradient terms. For a pancake vortex in the extreme 2D-case with \( \lambda_0/R \ll 1 \), the first critical field for a vortex in the 2D superconducting sheet is given as\[60, 134\]

\[
B_{c1}^\text{sheet} = \frac{\Phi_0}{4\pi\lambda^2} \left( \ln \kappa + \varepsilon_0 \right). \tag{72}
\]

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
field with very little difference in free energy between the configurations. In panel (d) we show 4 vortex configurations just above configuration contain more and more vortices. Panel (c) shows that at higher fields, different vortex configurations are possible at the same

to 0

vortex-core size

\( \varepsilon \)

one unknown and we extract

\( \kappa \)

the prefactor in the case

\( R \)

unknown parameters.

The numerical prefactor 4.4 = \( \ln(R/R_c) + \varepsilon_0 \) contains two unknown parameters \( R_c \) and \( \varepsilon_0 \) for \( \kappa = 20 \). On the other hand the prefactor in the case \( \kappa = 2 \), 1.62 = \( \ln \kappa + \varepsilon_0 \) contains only one unknown and we extract \( \varepsilon_0 = 1.62 - \ln \kappa \approx 0.93 \). Taking the results from Fig. [16] we can determine the nominal vortex-core size \( R_c \) as \( R_c = \ln R_c = \ln 2 \varepsilon_0/(\kappa = 2) - 4.4 \approx 0.68 \) or \( R_c = 2.7 \xi_0 \). 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\(^{[13]}\)

\[ B_{c1}(\kappa = 20) = 4.4 \frac{\Phi_0}{A} \]  
\[ B_{c1}(\kappa = 2) = 63.3 \frac{\Phi_0}{A} = 1.62 \frac{\Phi_0}{4\pi \xi_0}. \]  

The numerical prefactor 4.4 = \( \ln(R/R_c) + \varepsilon_0 \) contains two unknown parameters \( R_c \) and \( \varepsilon_0 \) for \( \kappa = 20 \). On the other hand the prefactor in the case \( \kappa = 2 \), 1.62 = \( \ln \kappa + \varepsilon_0 \) contains only one unknown and we extract \( \varepsilon_0 = 1.62 - \ln \kappa \approx 0.93 \). Taking the results from Fig. [16] we can determine the nominal vortex-core size \( R_c \) as \( R_c = \ln R_c = \ln 2 \varepsilon_0/(\kappa = 2) - 4.4 \approx 0.68 \) or \( R_c = 2.7 \xi_0 \). 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\(^{[13]}\)

\[ B_{c2} = \eta \frac{\Phi_0}{2\pi \xi_0}. \]  

We add a prefactor, \( \eta \), 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 disc has a width \( \sim 5 \xi_0 \) and thus we can estimate \( \eta = (20/25)^2 = 0.64 \). The upper critical field for our disc would be at \( \alpha = \frac{1}{3} (R/\xi_0)^2 = 312.5 \) 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 meta-stable configurations of vortices in the disc. For a stronger type-II superconductor, \( \kappa = 20 \), the lattice struc-
In particular, the number of vortices and their separation do not only depend on the 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 years.

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.\textsuperscript{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. \textsuperscript{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. \textsuperscript{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 of 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. \textsuperscript{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 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\textsuperscript{83} https://gitlab.com/superconga/superconga. An extensive user manual has been published online\textsuperscript{89} 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\textsuperscript{78,79} and therefore requires a CUDA-capable device (i.e. NVIDIA GPU\textsuperscript{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-Maxwell\textsuperscript{136} and Ginzburg-Landau\textsuperscript{117,137,138} and fully microscopic methods, such as density-functional theory\textsuperscript{139–143} and tight-binding approaches\textsuperscript{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

N.W-W. and P.H. have contributed equally to this work. We thank Anton B. Vorontsov and Kevin Marc Seja for valuable discussions. We thank the Swedish Research Council (VR) for financial support. The computations were enabled by resources provided by the Swedish National Infrastructure for Computing (SNIC) at the computing centers C3SE, HPC2N and NSC, partially funded by the Swedish Research Council through grant agreement no. 2018-05973.
Appendix A: Dependencies

This appendix lists external dependencies and packages that SuperConga offloads functionality to. For a more detailed description, please refer to the installation guide in the main README of the repository. SuperConga uses git as a version control system, with the central repository being hosted on GitLab. The SuperConga backend is written in C++, with most of the heavy lifting being done by CUDA kernels and Thrust 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 CMake as the build-system generator, and Ninja as the build system, but that it is possible to replace either of these with another choice. Linear algebra on the CPU uses Armadillo, and tests are implemented in the doctest framework. The real-time plotting is done using OpenGl, and ArrayFire-Forgé. The package JsonCpp is used to handle JSON objects on the backend side, and Hdf5 support. The SuperConga frontend requires a Python3 environment, in particular with the modules Numpy, Matplotlib, h5py, and panda.

Appendix B: The simulation-parameter file

The simulation configuration is specified in a simulation_config.json file. It is a JSON 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 Section 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
        }
      }
    }
  }
}
```

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 \( \Phi_0 \). The penetration_depth is given in units of \( \xi \). 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. \( 2\pi \). The choices 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 \( d_{xy}, g_{xy}(x−y)^2 \), 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

\[
\Delta_t(R) = \Delta_t^{(bulk)} e^{i \chi_t (\vec{R})} + Z, \quad (B1)
\]

\[
\chi_t(R) = 2 \pi \delta_t + \sum_v w_v \phi(R - C_v), \quad (B2)
\]

\[
Z \sim \mathcal{N}(0, \sigma_t^2), \quad (B3)
\]

where \( \delta_t \) is controlled by initial_phase_shift, and \( Z \) is complex additive Gaussian noise with a standard deviation \( \sigma_t \) 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, \(C_i = (\text{center}_x, \text{center}_y)\), and phase winding \(w_i = \text{phase}_w\). Furthermore, \(\phi(R) = \text{atan2}(y,x)\) is the polar angle of the coordinate \(R\). 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 \(d_{2s}\)-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.

```json
"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
      }
    ]
  }
}
```

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 \(2\pi k_B T_s\), to include in the energy sums. The \(p\)-norm to use when computing the residuals is controlled by \(\text{norm}\), with \(p \in \{1, 2, \infty\}\). The num_fermi_momenta is the number of discrete momenta to use when approximating the Fermi-surface averages \(\langle \ldots \rangle_F\). 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; \(\gamma_{\partial D}\) and \(\hat{\gamma}_{\partial D}\). 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 App. D.

All parameters of the accelerator are explained in App. E. Lastly, 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 HDF5 files or plain-text CSV files, respectively. Both data formats are standard and have excellent support in e.g. MATLAB and Python (via h5py and pandas). 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 nth 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

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

\[ f(s, \gamma(s)) = \frac{i}{v_F} \left[ \Delta^s(\gamma^2(s) + 2\zeta(s)\gamma(s) + \Delta(s)) \right] , \tag{C1}
\]

with the dependence on \( p_F \) and \( \epsilon \) dropped, and

\[ z(s) \equiv \epsilon + \frac{x}{v_F} \cdot A(s) , \tag{C2}
\]

for brevity. Instead of using the stepping method, 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 midpoint method, which we find to be both faster and more accurate. The implicit midpoint method has the following update rule,

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

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 \( \gamma_{j+1} \) to solve for each step along the trajectory;

\[
0 = c_2 \gamma_{j+1}^2 + c_1 \gamma_{j+1} + c_0 \,, \tag{C5}
\]

\[
c_2 = \frac{ih}{4v_F} \Delta^s \gamma_j^2 \,, \tag{C6}
\]

\[
c_1 = 2c_2 \gamma_j + \frac{ih}{v_F} \gamma_j + 1 \,, \tag{C7}
\]

\[
c_0 = (c_1 - c_2 \gamma_j + 2) \gamma_j - 4c_2^2 \gamma_j^2 \,, \tag{C8}
\]

with the solution

\[
\gamma_{j+1} = \frac{-c_1 + \sqrt{c_1^2 - 4c_2c_0}}{2c_2} . \tag{C9}
\]

Half-step values, \( \Delta_{j+\frac{1}{2}} \) and \( z_{j+\frac{1}{2}} \), are calculated by linear interpolation. Nearest-neighbour 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 \( \gamma(s_0 = R_{\min}) \) and stepping along the trajectory to obtain \( \gamma(s_j) \) with \( j > 0 \). The stepping proceeds until a second boundary is reached (denoted \( R_{\max} \), see Fig 1). We compute \( \gamma(s) \) by solving Eq. (26) in the same manner, but stepping in the opposite direction from \( R_{\max} \) to \( R_{\min} \).

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

\[
A_{m,n}^{\text{ind}} = -\frac{1}{k^2} \sum_{n'} \sum_{n''=0}^{N-1} G_{m-n',n'-n,jm'n'} \,, \tag{D1}
\]

\[
G_{m,n} = \frac{1}{4\pi} \int_{-\frac{1}{2}h}^{\frac{1}{2}h} \int_{-\frac{1}{2}h}^{\frac{1}{2}h} \ln \left( \frac{(x'+hm)^2 + (y'+hn)^2}{s_0^2} \right) \, dx' \, dy' \,. \tag{D2}
\]

\[
= \frac{1}{4\pi} \left( \frac{h}{s_0} \right)^2 \left( 2 \ln \left( \frac{h}{s_0} \right) - \ln(2) - 3 + \frac{1}{2} \sum_{a=\pm} \sum_{b=\pm} \Gamma(\alpha m, \beta n) \right) \,, \tag{D3}
\]

\[
\Gamma(m,n) = \alpha(m,n) + \beta(m,n) + \beta(n,m) \,, \tag{D4}
\]

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

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

\( G \) is a normal matrix, so it can be eigendecomposed as \( G = VAV^\top \), where \( V \) is a unitary matrix with (normalized)
eigenvectors as its columns, and \( \Lambda \) is a diagonal matrix with the eigenvalues as diagonal elements. Equivalently, 
\[
G_{m,n} = \sum_{k=0}^{N-1} \lambda_k V_m^{(k)} V_n^{(k)},
\]
where \( V_m^{(k)} \) is the eigenvector corresponding to the eigenvalue \( \lambda_k \). Thus 
\[
A_{m,n}^{\text{ind}} = -\frac{1}{\kappa^2} \sum_{k=0}^{N-1} \sum_{m'=0}^{N-1} \sum_{n'=0}^{N-1} \lambda_k V_{m'-m}^{(k)} V_{n'-n}^{(k)} j_{m'n'},
\]
and the cross-correlation with the eigenvectors can be done separately. The eigendecomposition is done with Armadillo. In order to increase performance only \( N^* \) eigenvalues are kept. It is computed as the smallest integer obeying 
\[
\sqrt{1 - \frac{\sum_{n=0}^{N^*-1} \lambda_n^2}{\sum_{n=0}^{N-1} \lambda_n^2}} < \varepsilon,
\]
with the eigenvalues sorted \( |\lambda_0| \geq |\lambda_1| \geq \ldots \geq |\lambda_{N-1}| \), 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.

Lastly, the mean of the charge-current density is in general not zero due to numerical errors or bad initialization. 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. (20) and (7) self-consistently. With our choice of units, the solution to Eq. (7) is given by
\[
\kappa^2 A_{\text{ind}}( \mathbf{R} ) = -\int_{\mathcal{A}} d\mathbf{R}' G( \mathbf{R}, \mathbf{R}' ) j( \mathbf{R}' ),
\]
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
\[
x = g(x),
\]
\[
x = ( \text{Re}[\Delta g], \text{Im}[\Delta g], \ldots, \kappa^2 A_x^{\text{ind}}, \kappa^2 A_y^{\text{ind}} )^\top.
\]
Note that the vector potential is scaled by \( \kappa^2 \) in order to have similar magnitudes of all elements in \( x \). SuperConga provides several different methods of solving Eq. (E2). Namely, basic Picard iterations, Polyak’s “small heavy sphere”, a variant of the Barzilai-Borwein method, 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 the different accelerators, it is convenient to define the difference between the right- and left-hand sides of Eq. (E2),
\[
d \equiv g(x) - x,
\]
which enters in all accelerator methods.

1. **Picard**

The simplest way of solving Eq. (E2) is by using Picard iterations,
\[
x_{i+1} = x_i + \alpha d_i,
\]
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, \( x \) 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),
\[
v_{i+1} = (1-\beta)v_i + \alpha d_i,
\]
\[
x_{i+1} = x_i + v_{i+1},
\]
where \( \beta \in (0, 1) \) is the drag, and \( \alpha > 0 \) is the step size. The velocity is initialized to zero, \( v_0 = 0 \). 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) method is similar to the Picard method but with an adaptive step size,

\[ s_{BB1}^i = \frac{\|d_{i-1}\|^2}{\|d_{i-1} \cdot (d_i - d_{i-1})\|}, \quad (E8) \]

\[ s_{BB2}^i = \frac{\|d_{i-1} \cdot (d_i - d_{i-1})\|^2}{\|d_i - d_{i-1}\|^2}, \quad (E9) \]

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

\[ x_{i+1} = x_i + \alpha_{i+1} d_i, \quad (E11) \]

where \( s_{BB1}^i \) and \( s_{BB2}^i \) 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.

```json
"accelerator": { 
    "name": "barzilai−borwein",
    "step_size": 1.0,
    "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 a order-parameter component, or the \( x \)- or \( y \)-component of the induced vector potential, in Eq. [E3], the update rule is as follows,

\[ s_{i,c} = S_c(m_{i,c}, d_{i,c}), \quad (E12) \]

\[ \alpha_{i+1} = \alpha_i k^{w_{i,c}}, \quad (E13) \]

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

\[ m_{i+1,c} = w_{i+1,c} m_{i,c} + (1 - w_{i+1,c}) d_{i,c}, \quad (E15) \]

\[ x_{i+1,c} = x_{i,c} + \alpha_{i+1} m_{i+1,c}, \quad (E16) \]

where \( S_c \) is the cosine similarity, \( m_{i,c} \) is an exponential moving average of the component difference \( d_{i,c} \), and \( k \) is a constant determining how much the step size should maximally increase (decrease) which can be set by the user. If \( s_{i,c} < s_{\text{min}} \), where \( s_{\text{min}} \) is the minimum tolerated similarity, the weight \( w^* \) is obtained by solving

\[ S_c(w^* m_{i,c} + (1 - w^*) d_{i,c}, d_{i,c}) = s_{\text{min}} \quad (E17) \]

ensuring that the step will be similar to \( d_{i,c} \). 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 \( m_{i,c} \) and \( d_{i,c} \) roughly point in the same direction, then we should have taken a larger step in 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.

```json
"accelerator": { 
    "name": "congacc",
    "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 III 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 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 \( \sim \) 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.

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

Appendix F: Units and basis functions

We summarize in Table III 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.
SuperConga examples and accelerator comparison

| Name                            | T / Tc | Φ / Φ0 | α | Symmetry       | Initialization | Geometry                  | Picard | Polyak | BB | CongAcc |
|---------------------------------|--------|--------|---|----------------|----------------|---------------------------|--------|--------|----|---------|
| (a) dwave_chiral                | 0.5    | 0      | ∞ | d_3 y + id_2 y | Bulk           | Disc: $R = 12.5 \xi_0$   | 80     | 34     | 25 | 37      |
| (b) dwave_octagon               | 0.5    | 0      | ∞ | d_2 y + d_2 x | Bulk           | Octagon: $S = 10 \xi_0^2$| 71     | 35     | 25 | 32      |
| (c) dwave_plus_swave            | 0.5    | 0      | ∞ | d_3 y + s     | Bulk           | Disc: $R = 12.5 \xi_0$   | 69     | 36     | 30 | 33      |
| (d) dwave_phase_crystal         | 0.1    | 0      | 100| d_2 y + s     | Bulk + vortex  | Irregular polygon         | 3027   | 809    | 1054 | 439     |
| (e) swave_abrikosov_lattice     | 0.2    | 20     | 10| s             | Bulk + giant vortex | Square: $S = 25 \xi_0^2$ | 1001   | 313    | 211 | 133     |
| (f) dwave_disc_meissner         | 0.5    | 0.5    | 5 | s             | Bulk           | Disc: $R = 15 \xi_0^2$   | F      | F      | 20 | 43      |
| (g) dwave_disc_vortex           | 0.5    | 1.5    | 5 | s             | Bulk + vortex  | Disc: $R = 15 \xi_0^2$   | F      | F      | 45 | 42      |

* The temperature is low, $T < T_c$, so the solution is a phase crystal.113
* The vortex has a phase winding $n = -1$ and is positioned outside of the grain, yielding a soft phase-gradient through the grain.
* The shape of the grain is a square, $S = 25 \xi_0^2$, with one corner removed, yielding four [100]-interfaces, and one pair-breaking [110]-interface.51
* The giant vortex in the middle has a phase winding $n = -13$. It is unstable and decays into a lattice of singly-quantized Abrikosov vortices.
* The vortex in the middle has a phase winding $n = -1$.

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 $R$, and a polygon side length $S$.

Input to quasiclassical theory

$\nu_v$ Fermi velocity

$N_F$ normal-state density of states per spin at the Fermi level

$T_c$ superconducting transition temperature (critical temperature)

Geometry

$A$ total superconducting computational area

$V = N_l d A$ volume of a stack of $N_l$ layers with interlayer distance $d$

Natural units

$T_c$ temperature in units of critical temperature

$2 \pi k_B T_c$ energy in units of critical temperature

$\xi_0 = \frac{\hbar v F}{2 e c T_c}$ length in units of superconducting coherence length

$\Phi_0 = \frac{hc}{2 e c}$ magnetic flux in units of superconducting flux quantum

Derived units and observables

$\lambda_0 = \frac{1}{\xi_0^2} = \frac{4 \pi \xi_0^2}{c^2} N_F v_F^2$ penetration depth. Here we use Gaussian units. In SI units $4 \pi / e^2 \to \mu_0$ with $\mu_0 \approx 4 \pi \times 10^{-7} J / A^2 m$

$\kappa = \frac{\Phi_0}{\xi_0^2}$ penetration depth in units of coherence length

$\alpha = \frac{4 \pi \Phi_0 A}{\mu_0}$ symmetric gauge: external magnetic flux penetrating the superconducting computational area in units of the flux quantum

$\lambda_0 = \frac{\Phi_0}{\xi_0^2}$ vector potential

$B_0 = \frac{\Phi_0}{\xi_0^2}$ magnetic flux density

$j_0 = 2 \pi k_B T_c |e| v_F N_F$ charge-current density

$m_0 = \hbar \xi_0^2 v$ magnetic moment

$\Omega_A = (2 \pi k_B T_c)^2 N_F A$ free energy

* We use calligraphic capital letters for quantities related to geometry, such as area $A$ or disc radius $R$.

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 $\hbar = k_B = c = 1$. 
Order-parameter basis functions

| Symmetry class for $D_{4h}$ | Order parameter $\Delta(k)$ | Basis function $\eta(\phi)$ for circular Fermi surface |
|-----------------------------|-----------------------------|-----------------------------------------------------|
| $A_{1g}$ ($s$)              | $1$                         | $1$                                                 |
| $A_{2g}$ ($g_{xy}(e^{2}-e^{4})$) | $k_{x}k_{y}(k_{x}^{2} - k_{y}^{2})$ | $\sqrt{2}\sin(4\phi)$ |
| $B_{1g}$ ($d_{z^{2}}$)      | $k_{z}^{2} - k_{y}^{2}$     | $\sqrt{2}\cos(2\phi)$                             |
| $B_{2g}$ ($d_{xy}$)         | $k_{x}k_{y}$                | $\sqrt{2}\sin(2\phi)$                             |

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.
1C. C. Tsuei and J. R. Kirtley, “Pairing symmetry in cuprate superconductors,” Rev. Mod. Phys. 72, 969–1016 (2000).

2J. R. Kirtley, C. Tsuei, C. Ariando, S. Harkema, and H. Hilgenkamp, “Angle-resolved phase-sensitive determination of the in-plane gap symmetry in yb2cu3o7–δ,” Nature Physics 2, 190 – 194 (2006).

3N. Kokubo, S. Okayasu, A. Kanda, and B. Shinozaki, “Scanning tunnel microscopy study of vortex polygons and shells in weak-pinning disks of an amorphous superconducting film,” Phys. Rev. B 82, 014510 (2010).

4A. 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 LaAlO3/SrTiO3 interface,” Nature Physics 7, 1–5 (2011).

5D. Vasyyukov, V. Anahory, L. Embon, D. Halbertal, J. Cuppens, L. Neeman, A. Finkler, Y. Segev, Y. Myassedov, M. L. Rappaport, M. E. Huber, and E. Zeldov, “A scanning superconducting quantum interference device with single electron spin sensitivity,” Nature Nanotechnology 8, 1–6 (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, O. Fischer, and F. Lévy, “Scanning tunneling spectroscopy of a vortex core from the clean to the dirty limit,” Physical Review Letters 67, 1650–1652 (1991).

8I. V. Grigorieva, W. Escoffier, J. Richardson, L. Y. Vinnikov, S. Dubonos, and J. G. E. Harris, “Direct observation of mesoscopic vortex states in transition metal nanowires,” Physical Review Letters 101, 166407 (2008).

9I. Guillamón, H. Suderow, S. Vieira, L. Cario, P. Diener, and P. Rodière, “Superconducting Density of States and Vortex Cores of 2H-NbS2,” Phys. Rev. Lett. 75, 2754–2757 (1995).

10A. 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).

11S. Pan, E. Hudson, A. Gupta, K. Ng, H. Eisaki, S. Uchida, and J. Davis, “STM Studies of the Electronic Structure of Vortex Cores in Bi2Sr2CaCu2O8−δ,” Phys. Rev. Lett. 85, 1536 – 1539 (2000).

12S. Pan, E. Hudson, K. Lang, H. Eisaki, S. Uchida, and J. Davis, “Imaging the effects of individual zinc impurity atoms on superconductivity in Bi2Sr2CaCu2O8−δ,” Nature 403, 746 – 750 (2000).

13E. Hoffmann, K. McElroy, D.-H. Lee, K. M. Lang, H. Eisaki, S. Uchida, and J. C. Davis, “Imaging Quasiparticle Interference in Bi2Sr2CaCu2O8−δ,” Science 297, 1148–1151 (2002)-cond-mat/0209276.

14L. Guillamón, H. Suderow, S. Vieira, L. Cario, P. Diener, and P. Rodière, “Superconducting Density of States and Vortex Cores of 2H-NbS2,” Physical Review Letters 101, 166407 (2008).

15D. Roditchev, C. Brun, L. Serrier-Garcia, J. C. Cuevas, V. H. L. Bessa, M. Tinkham, D. Rainer, and J. A. Sauls, “Strong-coupling theory of superconductivity: From Basic Physics to the Latest Developments,” World Scientific, Oxford, 2001).

16N. Kopnin, Theory of Nonequilibrium Superconductivity (Clarendon Press, Oxford, 2001).

17I. 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).

18V. V. Khokhryakov, M. V. Milošević, and S. J. Bending, “A scanning hall probe microscope for high resolution magnetic imaging down to 300 mk,” Review of Scientific Instruments 79, 123708 (2008) https://doi.org/10.1063/1.3046285

19P. Lafarge, P. Joyce, 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).

20D. C. Ralph, C. T. Black, and M. Tinkham, “Spectroscopic Measurements of Discrete Electronic States in Single Metal Particles,” Physical Review Letters 74, 3241–3244 (1995).

21S. E. Kubatkin, A. Y. Tzalenchuk, Z. G. Ivanov, P. Delsing, R. I. Shekhter, and T. Claeson, “Coulomb blockade electrometer with a high-Tc island,” JETP Letters 63, 126–132 (1996).

22D. Gustafsson, D. Golubev, M. Fogelström, T. Claeson, S. Kubatkin, T. Bauch, and F. Lombardi, “Fully gapped superconductivity in a nanometre-size YBa2Cu3Oδ+δ island enhanced by a magnetic field,” Nature Nanotechnology 8, 25–30 (2013).

23H. Barean, L. N. Cooper, A. E. Filippov, and F. M. Peeters, “Phase transitions in individual sub-micrometre superconductors,” Nature 390, 259–262 (1997).

24M. T. Tuominen, J. M. Hergerth, Y. 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).

25D. Vasyukov, Y. Anahory, L. Embon, D. Halbertal, J. Cuppens, L. Neeman, A. Finkler, Y. Segev, Y. Myassedov, M. L. Rappaport, M. E. Huber, and E. Zeldov, “A scanning superconducting quantum interference device with single electron spin sensitivity,” Nature Nanotechnology 8, 1–6 (2013).

26H. 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).

27C. Renner, A. D. Kent, P. Niedermann, O. Fischer, and F. Lévy, “Scanning tunneling spectroscopy of a vortex core from the clean to the dirty limit,” Physical Review Letters 67, 1650–1652 (1991).

28I. V. Grigorieva, W. Escoffier, J. Richardson, L. Y. Vinnikov, S. Dubonos, and J. G. E. Harris, “Direct observation of mesoscopic vortex states in transition metal nanowires,” Physical Review Letters 101, 166407 (2008).

29D. Rainer and J. A. Sauls, “Strong-coupling theory of superconductivity: From Basic Physics to the Latest Developments,” World Scientific, Oxford, 2001).

30N. Kopnin, Theory of Nonequilibrium Superconductivity (Clarendon Press, Oxford, 2001).

31M. Fogelström and J. Kurkijärvi, “Quasiclassical theory of vortices in 3 He-B,” Journal of Low Temperature Physics 98, 195–226 (1995).

32J. Sauls and M. Eschrig, “Vortices in chiral, spin-triplet superconductors and superfluids,” New Journal of Physics 11, 075008 (2009).

33M. A. Sillavu, E. V. Thuneberg, and M. Fogelström, “Lifshitz Transition in the Double-Core Vortex in He–B3,” Physical Review Letters 115, 235301 (2015).

34K. M. Seja and T. Löfwander, “Finite element method for the quasiclassical theory of superconductivity,” Phys. Rev. B 106, 144511 (2022).

35J. A. Sauls and M. Eschrig, “Vortices in chiral, spin-triplet superconductors and superfluids,” New Journal of Physics 11, 075008 (2009).

36A. Shelankov and M. Ozana, “Quasiclassical theory of superconductivity: A multiple-interface geometry,” Phys. Rev. B 61, 7707–7710 (2000).

37K. D. Usadel, “Generalized Diffusion Equation for Superconducting Alloys,” Phys. Rev. Lett. 25, 507–509 (1970).
54. M. Amundsen and J. Linder, “General solution of 2d and 3d superconducting quasilocal systems: coalescing vortices and nanosolid geometries,” Scientific Reports 6, 22765 (2016).

55. M. Timmermans, L. Serrier-Garcia, M. Perini, J. Van de Vondel, and V. V. Moshchalkov, “Direct observation of condensate and vortex confinement in nanosteructured superconductors,” Phys. Rev. B 93, 054514 (2016).

56. A. B. Vorontsov and J. Sauls, “Thermodynamic properties of thin films of superfluid He-A,” Phys. Rev. B 68, 064508 (2003).

57. M. Håkansson, T. Löfwander, and M. Fogelström, “Spontaneously broken time-reversal symmetry in high-temperature superconductors,” Nature Physics 11, 755 (2015).

58. P. Holmwall, M. Fogelström, T. Löfwander, and A. B. Vorontsov, “Phase crystals,” Phys. Rev. Research 2, 013104 (2020).

59. S. Yip and A. Garg, “Superconducting states of reduced symmetry: General order parameters and physical implications,” Phys. Rev. B 48, 3304–3308 (1993).

60. M. Graf, D. Rainer, and J. Sauls, “Coupled two-dimensional Fermi liquids as a model for layered superconductors: Basic equations and elementary results.” Physical Review B (Condensed Matter) 47, 12089–12098 (1993).

61. N. W. Wennerdal, A. Ask, P. Holmwall, 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. Research 2, 043198 (2020).

62. M. Eschrig, “Distribution functions in nonequilibrium theory of superconductivity and andreev spectroscopy in unconventional superconductors,” Phys. Rev. B 61, 9061–9076 (2000).

63. J. M. Luttinger and J. C. Ward, “Ground-State Energy of a Many-Fermion System. II.” Physical Review 118, 1417–1427 (1960).

64. E. V. Thuneberg, J. Kurkijärvi, and D. Rainer, “Elementary-flux-pinning potential in type-II superconductors,” Phys. Rev. B 29, 3913–3923 (1984).

65. A. V. Zaitev, “Quasiclassical equation of the theory of superconductivity for contiguous metals and the properties of proximitized microstructures,” Zh. Eksp. Teor. Fiz. 86, 1742 (1984).

66. A. V. Zaitev, “Self-consistent calculations of the pair potential and the tunneling density of states in proximity contacts,” Phys. Rev. B 35, 6762–6770 (1987).

67. A. Millis, D. Rainer, and J. A. Sauls, “Quasiclassical theory of superconductivity near magnetically active interfaces,” Phys. Rev. B 38, 4504–4515 (1988).

68. M. Fogelström, “Josephson currents through spin-active interfaces,” Phys. Rev. B 62, 11812–11819 (2000).

69. E. Zhao, T. Löfwander, and J. A. Sauls, “Nonequilibrium superconductivity near spin-active interfaces,” Phys. Rev. B 70, 134510 (2004).

70. M. Eschrig, “Scattering problem in nonequilibrium quasiclassical theory of metals and superconductors: General boundary conditions and applications,” Phys. Rev. B 80, 134511 (2009).

71. M. Ozana and A. Shlank, “Quasiclassical theory of superconductivity: Interfering paths,” Phys. Rev. B 65, 014510 (2001).

72. A. Samolenkova and E. Babaev, “Boundary states with elevated critical temperatures in bardeen-cooper-schrieffer superconductors,” Phys. Rev. B 101, 134512 (2020).

73. C. Hainzl, B. Roos, and R. Seiringer, “Boundary superconductivity in the bcs model,” arXiv 2201.08090 (2022).

74. Y. Nagato, K. Nagai, and J. Harra, “Theory of the Andreev reflection and the density of states in proximity contact normal-superconducting infinite double-layer,” Journal of Low Temperature Physics 93, 33–56 (1993).

75. N. Schopohl and K. Maki, “Quasiparticle spectrum around a vortex line in a d-wave superconductor,” Phys. Rev. B 52, 490–493 (1995).

76. N. Schopohl, “Transformation of the eilenberger equation of superconductivity to a scalar riccati equation,” arXiv (1998), cond-mat/9804064.

77. T. Ozaki, “Continued fraction representation of the fermi-dirac function for large-scale electronic structure calculations,” Phys. Rev. B 55, 035123 (2007).

78. NVIDIA Corporation, “NVIDIA CUDA programming guide,” (2021), version 11.2.0.

79. J. Nickolls, I. Buck, M. Garland, and K. Skadron, “Scalable parallel programming with cuda,” Queue 6, 40–53 (2008).

80. SuperConga team, “SuperConga Online Documentation and User Manual,” https://superconga.github.io/superconga-doc/2022/.

81. B. Polvák, “Some methods of speeding up the convergence of iteration methods,” USSR Computational Mathematics and Mathematical Physics 4, 1–17 (1964).

82. J. Barzilai and J. M. Borwein, “Two-Point Step Size Gradient Methods,” IMA Journal of Numerical Analysis 8, 141–148 (1988). https://academic.oup.com/imajna/article-pdf/8/1/141/2402762/8-148.pdf

83. SuperConga team, “SuperConga GitLab repository,” https://gitlab.com/superconga/superconga (2022).

84. P. Holmwall, A. B. Vorontsov, M. Fogelström, and T. Löfwander, “Spataneous symmetry breaking at surfaces of d-wave superconductors: Influence of geometry and surface ruggedness,” Phys. Rev. B 99, 184511 (2019).

85. C. Caroli, P. De Gennes, and J. Matricon, “Bound fermion states on a vortex line in a type i superconductor,” Physics Letters 9, 307–309 (1964).

86. H. Kim, Y. Nagai, L. Rózsa, D. Schreyer, and R. Wiesendanger, “Anisotropic non-split zero-energy vortex bound states in a conventional superconductor,” (2021), arXiv:2105.01354 [cond-mat.superron]

87. C. P. Bean and J. D. Livingston, “Surface barrier in type-iii superconductors,” Phys. Rev. Lett. 12, 14–16 (1964).

88. H. 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).

89. V. 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 Science and Technology 22, 034001 (2009).

90. L.-F. Zhang, L. Covaci, M. V. Milošević, G. R. Berdyyov, 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).

91. L.-F. Zhang, L. Covaci, M. V. Milošević, G. R. Berdyyov, and F. M. Peeters, “Vortex states in nanoscale superconducting squares: The influence of quantum confinement,” Phys. Rev. B 88, 144501 (2013).

92. H. 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).

93. L. E. Cabral and J. A. Aguiar, “Vortex configurations in thin superconducting equilateral triangles,” Phys. Rev. B 80, 214533 (2009).

94. N. Kokubo, H. Miyahara, S. Okayasu, and T. Nojima, “Vortex shells in mesoscopic triangles of amorphous superconducting thin films,” Physica C: Superconductivity and its Applications 530, 42–45 (2016) 2nd International Symposium on Superconductivity.

95. W. Wu, H. J. Zhao, B. Li, and Z. X. Shi, “Finite-size and shape effects on vortex configurations in superconducting isocceles right triangles,” Supercond. Sci. Technol. 30, 125004 (2017).

96. H. T. Huy, M. Kato, and T. Ishida, “Vortex states in the femtoscopic mo80ge20pentagon plates,” Superconductor Science and Technology 26, 065001 (2013).

97. H. J. Fink and A. G. Presson, “Magnetic irreversible solution of the ginzburg-landau equations,” Phys. Rev. 151, 210–228 (1966).

98. V. Moshchalkov, M. Baert, V. V. Moshchalkov, E. Rosseel, M. J. Van Bael, K. Temst, R. Jonckheere, and Y. Bruynseraede, “Magnetization of multiple-quantum vortex lattices,” Phys. Rev. B 54, 7385–7393 (1996).
100. V. A. Schweigert, F. M. Peeters, and P. S. Deo, “Vortex phase diagram for mesoscopic superconducting disks,” Phys. Rev. Lett. 81, 2783–2786 (1998).
101. A. Kanda, B. J. Baels, 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).
102. D. S. Golubović, M. V. Milošević, F. M. Peeters, and V. V. Moschchalkov, “Magnetically induced splitting of a giant vortex state in a mesoscopic superconducting disk,” Phys. Rev. B 71, 180502 (2005).
103. T. Cren, L. Serrier-Garcia, F. Debontrider, and D. Roditchev, “Vortex fusion and giant vortex states in confined superconducting condensates,” Phys. Rev. Lett. 107, 097022 (2011).
104. F. Chibotaru, A. Ceulemans, V. Bruyndoncx, and V. V. Moschchalkov, “Symmetry-induced formation of antivortices in mesoscopic superconductors,” Nature 408, 833–835 (2000).
105. M. Iavarone, A. Scarfato, F. Bobba, M. Longobardi, G. Karapetrov, V. Novosad, V. Yefremenkov, F. Giubilei, and A. M. Cucolo, “Imaging the spontaneous formation of vortex-antivortex pairs in planar superconductor/ferromagnet hybrid structures,” Phys. Rev. B 84, 024506 (2011).
106. D. G. 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. Yefremenkov, and F. Bobba, “Superconducting vortex-antivortex pairs: Nucleation and confinement in magnetically coupled superconductor/ferron magnet hybrid nanomagnets,” in *Vortex Dynamics and Optical Vortices*, edited by H. Perez-de Tejada (InTech, Rijeka, 2017) Chap. 03.
107. J. Simmendinger, M. Bihler, A. M. Ionescu, M. Weigand, G. Schütz, and J. Albrecht, “Bound and stable vortex–antivortex pairs in high-first-critical-temperature superconductors,” New Journal of Physics 22, 123035 (2020).
108. A. B. Vorontsov, “Broken translational and time-reversal symmetry in unconventional superconducting films,” Phys. Rev. Lett. 102, 177001 (2009).
109. P. Holmvall, *Modeling mesoscopic unconventional superconductors* (Chalmers University of Technology, 2017) licentiate thesis.
110. P. Holmvall, T. Löfwander, and M. Fogelström, “Spontaneous generation of fractional vortex-antivortex pairs at single edges of high-Tc superconductors,” Journal of Physics: Conference Series 669, 012037 (2018).
111. P. Holmvall, A. B. I. Vorontsov, M. Fogelström, and T. Löfwander, “Broken translational symmetry at edges of high-temperature superconductors,” Nature Communications 9, 2190 (2018).
112. P. Holmvall, *Crystalization of the superconducting phase in unconventional superconductors* (Chalmers University of Technology, 2019) PhD thesis.
113. A. A. Abrikosov, “On the magnetic properties of superconductors of the second group,” Soviet Phys JETP 5, 1174–1182 (1957).
114. J. Pearl, “Current distribution in superconducting films carrying quantized fluxoids,” Applied Physics Letters 5, 65–66 (1964) [https://doi.org/10.1063/1.1754056].
115. D. Xu, S. K. Yip, and J. A. Sauls, “Nonlinear meissner effect in unconventional superconductors,” Phys. Rev. B 51, 16233–16253 (1995).
116. S. Igashitani, “Mechanism of paramagnetic meissner effect in high-temperature superconductors,” J. Soc. Phys. Jpn. 66, 2556 (1997).
117. M. Fogelström, D. Rainer, and J. A. Sauls, “Tunneling into current-carrying superconducting states of high-temperature superconductors,” Phys. Rev. Lett. 79, 281–284 (1997).
118. M. Fogelström, D. Rainer, and J. Sauls, “Nonlinear magnetic field dependence of the conductance in d-wave normal-metal/superconductor tunnel junctions,” Phys. Rev. B 70, 012503 (2004).
119. J. R. Clem, M. W. Coffey, and Z. Hao, “Lower critical field of a Josephson-coupled layer model of high-temperature superconductors,” Phys. Rev. B 44, 2732–2738 (1991).
120. P. Holmvall and J. Furrhmüller, “Efficiency of ab-initio total energy calculations for metals and semiconductors using a plane-wave basis set,” Computational Materials Science 6, 15–30 (1996).
121. G. Kresse and J. Furrhmüller, “Efficient iterative schemes for ab initio total-energy calculations using a plane-wave basis set,” Phys. Rev. B 54, 11169–11186 (1996).
122. P. Giannozzi, S. Baroni, N. Bonini, M. Calandra, R. Car, C. Cavazzoni, D. Ceresoli, G. L. Chiarotti, M. Cococcioni, I. Dabo, A. B. Calvino, S. de Gironcoli, S. Fabris, G. Fratesi, R. Gebauer, U. Gerstmann, C. Gougoussis, A. Koshevlev, C. Phillips, D. Karpeyev, and A. Glatz, “Stable large-scale solver for ginzburg–landau equations for superconductors,” Journal of Computational Physics 294, 639–654 (2015).
123. F. N. Rybakov and E. Babayev, “Excalibur software,” URL: http://quantumclassical.com/excalibur/ (2022).
124. P. Giannozzi, O. Andreussi, T. Brumme, O. Buxa, M. B. Nardelli, M. Calandra, R. Car, C. Cavazzoni, D. Ceresoli, M. Cococcioni, N. Colonna, I. Carmineo, A. D. Corso, S. de Gironcoli, P. Delugas, R. A. DiStasio, A. Ferrer, A. Floris, G. Fratesi, G. Falguero, R. Gebauer, U. Gerstmann, F. Giustino, T. Gorni, J. Jia, M. Kawanura, H.-Y. Ko, A. Kokalj, E. Kückenberger, M. Lazzeri, M. Marsili, N. Marzari, F. Mauri, R. Mazzarello, S. Paolini, A. Pasquarello, L. Paulatto, C. Sbraccia, S. Scandolo, G. Schlaaerzor, A. F. Seitzsonen, A. Smogunov, P. Umari, and R. M. Wentzcovitch, “QUANTUM ESPRESSO: a modular and open-source software package for quantum simulations of materials,” Journal of Physics: Condensed Matter 21, 395502 (2009).
125. P. Giannozzi, O. Andreucci, F. Mauri, G. Scandolo, G. Sclauzero, A. P. Seitsonen, A. Smogunov, P. Umari, and R. M. Wentzcovitch, “QUANTUM ESPRESSO: a modular and open-source project for quantum simulations of materials,” Journal of Physics: Condensed Matter 29, 465501 (2017).
126. M. Nakahara, S. A. Katabi, and F. M. Peeters, “Tight-binding studio: A technical software package to find the parameters of tight-binding hamiltonian,” Computer Physics Communications 254, 107379 (2020).
127. W. Groth, M. Wimmer, A. R. Ahmerow, and X. Wantal, “Kwant: a software package for quantum transport,” New J. Phys. 16, 063065 (201) publisher: IOP Publishing.
K. Björnson, “Tbtk: A quantum mechanics software development kit,” SoftwareX 9, 205–210 (2019).

T. 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).

Y. Nagai, “N-independent localized krylov–bogoliubov-de gennes method: Ultra-fast numerical approach to large-scale inhomogeneous superconductors,” Journal of the Physical Society of Japan 89, 074703 (2020), https://doi.org/10.7566/JPSJ.89.074703.

N. Bell and J. Hoberock, “Chapter 26 - thrust: A productivity-oriented library for cuda,” in GPU Computing Gems Jade Edition Applications of GPU Computing Series, edited by W. mei W. Hwu (Morgan Kaufmann, Boston, 2012) pp. 359 – 371.

K. Kaczynski and P. Rzakaczewski, “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, Berlin, Heidelberg, 2013) pp. 37–46.

E. 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).

Cedilnik, E. and Hoffman, B. and King, B. and Martin, K. and Neundorf, A., “CMake,” https://cmake.org/ (2021), version 3.21.3.

C. Sanderson and R. Curtin, “Armadillo: a template-based C++ library for linear algebra,” The Journal of Open Source Software 1, 26 (2016).

C. 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, Cham, 2018) pp. 422–430.

V. Kirilov, “doctest,” https://github.com/onqtam/doctest (2021).

Khoros Group, “Opengl overview,” (2021), version 4.6.

M. 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).

P. Yalamanchili, U. Arshad, Z. Mohammed, P. Garigipati, P. Entschev, B. Kloppenborg, J. Malcolm, and J. Melonakos, “ArrayFire - A high performance software library for parallel computing with an easy-to-use API,” (2015).

C. Dunn, “Jsoncpp,” https://github.com/open-source-parsers/jsoncpp (2020).