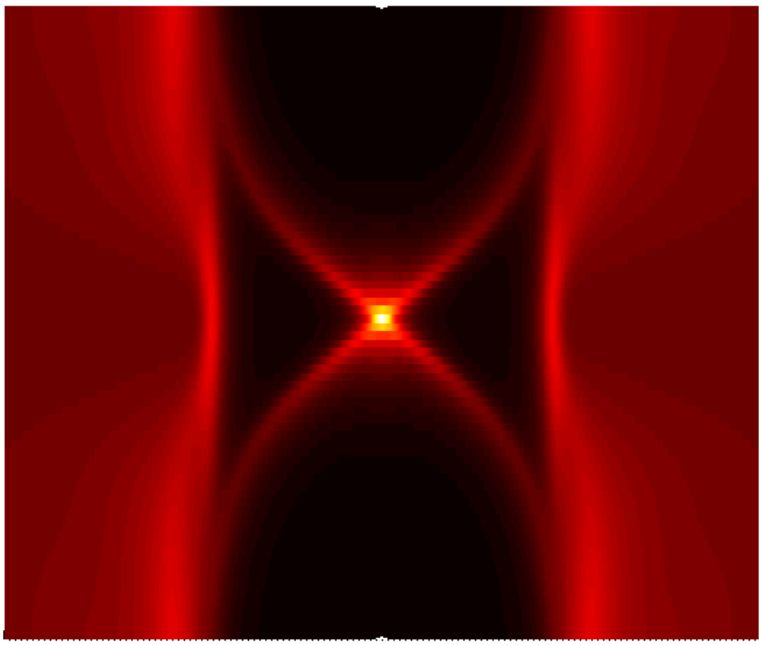
CHALMERS



Parallel Computations of Vortex Core Structures in Superconductors

Master of Science Thesis in Fundamental Physics

NICLAS E. WENNERDAL

Department of Microtechnology and Nanoscience
Applied Quantum Physics Laboratory
CHALMERS UNIVERSITY OF TECHNOLOGY
Göteborg, Sweden, 2011

Parallel Computations of Vortex Core Structures in Superconductors

NICLAS E. WENNERDAL

Department of Microtechnology and Nanoscience
CHALMERS UNIVERSITY OF TECHNOLOGY
Göteborg, Sweden 2011

Parallel Computations of Vortex Core Structurs in Superconductors
NICLAS E. WENNERDAL

© NICLAS E. WENNERDAL, 2011.

Department of Microtechnology and Nanoscience
Chalmers University of Technology
SE-412 96 Göteborg, Sweden
Telephone +46 (0)31-772 1000

Cover: A top view picture of the local density of states of a trajectory passing through the center of the vortex core of phase winding one.

Chalmers Reproservice
Göteborg, Sweden 2011

Abstract

In this master thesis the properties of an isolated vortex of flux quantum one in s-wave superconductors is investigated. This is done by solving the Eilenberger transport equation for the quasi-classical Matsubara propagator in the extreme clean limit. The local density of states (LDOS) in the vortex center is calculated and the Kramer-Pesch effect is demonstrated. Furthermore, the self-consistency equation for the order parameter, the LDOS and the current calculations have large amounts of data parallelism. By programming in CUDA C this parallelism is exploitable. Parallel computations are carried out on an NVIDIA graphics card giving speedups up to a factor of 100 for small systems and 10 for large ones, compared to serial code, thus showing the immense power of parallelizing calculations and moving them to the graphics card.

Keywords: Superconductivity, Vortex Core, Quasi-classical Method, Kramer-Pesch Effect, CUDA, Parallel Computing, NVIDIA

Contents

1 Introduction **1**
- 1.1 Vortices in Type II Superconductors . . . 1
- 1.2 High T_c Superconductors . . . 2
- 1.3 Goal and Structure of the Thesis . . . 3

2 Green's Function Formalism **4**
- 2.1 Propagators and Green's Functions . . . 5
- 2.2 Second Quantization . . . 7
- 2.3 Finite Temperatures . . . 9

3 The BCS Theory of Superconductivity **11**
- 3.1 The Energy Gap . . . 11
  - 3.1.1 Thermodynamic Properties of Bulk s-wave Superconductors 12
- 3.2 The Gor'kov Equation . . . 13
- 3.3 The Quasi-classical Method . . . 15
- 3.4 Δ as a Step Function . . . 17
- 3.5 Results . . . 18
  - 3.5.1 The Vortex Core Size . . . 19
  - 3.5.2 The Density of States . . . 19
  - 3.5.3 The Kramer-Pesch Effect . . . 20

4 CUDA and Parallel Computing **25**
- 4.1 CUDA C . . . 25
  - 4.1.1 Terminology . . . 25
- 4.2 The GPU Architecture . . . 26
  - 4.2.1 Invoking the Kernel . . . 27
- 4.3 Optimizing Performance . . . 28
  - 4.3.1 Parallelization . . . 29
  - 4.3.2 Memory Usage . . . 30
  - 4.3.3 Instruction Throughput . . . 30
- 4.4 Results . . . 31
  - 4.4.1 Exposing the Data Parallelism . . . 31
  - 4.4.2 Exploiting the Data Parallelism . . . 31
  - 4.4.3 Speedup . . . 33

5 Conclusions and outlook **35**
- 5.1 Acknowledgements . . . 36

Chapter 1

Introduction

In 1911 H. Kammerling Onnes discovered the peculiar phenomenon that when cooling metals to a few degrees above absolute zero the electrical resistance suddenly vanishes at a critical temperature, T_c, which depends on the material. This phenomenon is now called superconductivity, and the materials exhibiting it superconductors. A few years after his discovery it was shown that these materials not only exhibit perfect conductivity but also perfect diamagnetism, i.e. they expel magnetic fields even though they did not do this when being in the normal, non-superconducting, state. This could not be explained by classical physics and remained unexplained for quite some time. It was not until 1957 when Bardeen, Cooper and Schrieffer formulated their famous theory, now called the BCS theory of superconductivity, that it was understood at a microscopical level. A phonon mediated attraction between electrons makes the Fermi sea unstable at low temperatures allowing electrons to pair up and form so called Cooper pairs. Because these pairs are effectively bosons they can all condensate into the ground state. The particles in this condensate all have the same phase in the absence of a current. Thus superconductivity breaks $U(1)$ symmetry. If more symmetries than $U(1)$ are broken, the superconductors are called *unconventional*. The pairing in unconventional superconductors are often magnetic in nature, in contrast to the phonon-electron coupling exhibited in conventional ones. How to describe unconventional superconductors is an open question. Thus superconductivity is still, a century after its discovery, an active area of research. [1]

1.1 Vortices in Type II Superconductors

Superconductors are divided into two different kinds, with respect to their behaviour in a magnetic field; type I and type II. They have a first order and second order phase transition respectively. A type I superconductor exhibits a sudden change at the critical magnetic field strength H_c, being normal above it and superconducting below it, expelling the field completely[1]. In a type II superconductor however there exists penetrating flux lines creating vortices of

[1] Technically speaking the expulsion is not perfect, there is always an exponentially decaying penetration of flux lines.

shielding supercurrents between its two critical fields, $H_{c1} \leq H \leq H_{c2}$. Above $H \geq H_{c2}$ the material becomes normal. [2]

Using the phenomenological Ginzburg-Landau theory the division between the two kinds of superconductors is made by considering the dimensionless parameter $\kappa = \lambda_{\text{eff}}(T)/\xi(T)$, where λ_{eff} is the effective penetration depth and ξ is the Ginzburg-Landau coherence length. For type II superconductors $\kappa > 1/\sqrt{2}$ whereas the clean type I probed by Kammerling Onnes has $\kappa \ll 1$. In the extreme type II limit $\kappa \gg 1$ analytical analysis of the Ginzburg-Lanadu equations can be carried out showing how the magnetic field starts penetrating the material at $H = H_{c1}$. It gives that the magnetic flux entering the superconductor is quantized, with the flux quantum given by $\Phi_0 = hc/(2e)$. This is a direct consequence of the physical constraint on the order parameter being single-valued, i.e. a contour enclosing a vortex must return to itself, differing in phase only in integral values of 2π. Thus

$$\oint_C \mathrm{d}\boldsymbol{l} \cdot \nabla\varphi \quad = \quad 2\pi n, \tag{1.1}$$

where n is the winding number, which is equal to the number of flux quanta enclosed by the contour. These topological defects induce the screening supercurrents which form vortices around the defects and were first described by A. Abrikosov in 1957 which won him the Nobel Prize in physics in 2003. These vortices form a triangular lattice that moves due to the Lorentz force when a current is flowing through the specimen causing electrical resistance, this effect is called flux creep. However, the vortices can be pinned down in certain favourable locations with low energy due to inhomogeneities, thus allowing greater critical currents. This is used in, for example, the superconducting magnets in the accelerator LHC at CERN. In this thesis the study is restricted to a single isolated vortex. This approximation of treating the vortices separately is appropriate whenever the distance between them is large compared to the penetration depth λ, i.e. when the external magnetic field is close to H_{c1}.

1.2 High T_c Superconductors

In 1986 Bednorz and Müller, at IBM, discovered the first high T_c superconductor (HTS) for which they were rewarded the Nobel prize in 1987. Although it did not have an extraordinary high transition temperature it had higher T_c than the believed maximum predicted by the BCS theory. Bednorz and Muller also showed that layered copper oxides have peculiar and very interesting properties regarding superconductivity. After their discovery the highest known transition temperature quickly soared to over 130 K. This means that cooling using liquid helium is not necessary for these materials, liquid nitrogen is sufficient which means that it is much cheaper and simpler to reach T_c. HTSs also have the ability of carrying very large critical supercurrents, thus making them attractive for use in electro magnets. But the HTSs comes with a price. The high operational temperature implies large fluctuations due to thermal excitations. Furthermore the anisotropy makes the pinning of flux lines less effective. [1]

To be able to manufacture HTSs with high H_{c2} and eventually the holy grail - room temperature superconductors [3] - it is necessary to understand

the mechanism behind unconventional superconductivity, which is perhaps not phonon-electron interaction. [2]

1.3 Goal and Structure of the Thesis

One of the problems of modelling vortices in superconductors is that the calculations are very computationally demanding. Especially at low temperatures, where the number of so called Matsubara frequencies needed grows as the inverse temperature. However, a large part of the calculations are independent of each other, i.e. there are many nested for loops where the individual iterations do not depend on the previous ones. This means that there are tremendous amounts of data parallelism that can be exploited.

The ever growing demand for high definition real-time computer graphics by the gaming industry has led to a modern graphics card being a powerhouse capable of massively parallel computations. Its power, however, has been outside the general programmers reach until NVIDIA, in 2006, released their first general-purpose programmable graphics cards. By simply using their extension to the C programming language called CUDA one is able to do truly parallel computations, previously requiring a computer cluster. [4]

The goal of this thesis is pretty straightforward; investigate single vortex core structures utilizing parallel computation capability of NVIDIAs graphics cards. It is divided into three parts:

1. Green's Function Formalism: A quick overview of the definition and use of Green's functions in many-body quantum physics.

2. The BCS Theory of Superconductivity: Conventional superconductivity is described in terms of Green's functions and the Eilenberger equation for the quasi-classical matrix propagator is derived. The density of states, screening supercurrents etc. are calculated and compared to results found in literature.

3. CUDA and Parallel Computation: Here an overview of the architecture of a modern graphics card is presented together with guidelines for parallel programming using CUDA C. Furthermore, the resulting speedup using parallel computations compared to serial ones is discussed.

The chapters are written to be as independent as possible. For example, the parallelization of the calculations are done in the CUDA chapter, making the BCS chapter accessible even to those not interested in the computer science. The BCS chapter, however, assumes previous knowledge of Green's functions.

Chapter 2

Green's Function Formalism

Frequently in physics one encounters ordinary differential equations on the form

$$Lu = f \tag{2.1}$$

where L is a linear differential operator given by

$$L = \sum_{|\alpha|\leq m} A_\alpha D^\alpha \tag{2.2}$$

where $\alpha = (\alpha_1, \ldots, \alpha_N)$ with $\alpha_n \geq 0$ and $|\alpha| = \sum_n \alpha_n$, A_α are functions in $\mathbb{R}^N$ and

$$D^\alpha = \left(\frac{\partial}{\partial x_1}\right)^{\alpha_1} \ldots \left(\frac{\partial}{\partial x_N}\right)^{\alpha_N} = \frac{\partial^{|\alpha|}}{\partial x_1^{\alpha_1} \ldots \partial x_N^{\alpha_N}}, \tag{2.3}$$

u is the function being solved for and f is an arbitrary given function. When having to solve an equation like this one several times, with the same L and boundary conditions but with different homogeneous parts, it would be very convenient to form the inverse of L, writing

$$u = L^{-1} f \tag{2.4}$$

obeying the correct boundary conditions. To form this inverse one must first consider equations on the form

$$LG = \delta \tag{2.5}$$

where δ is Dirac's delta function. Acting with L on the convolution of G and f now yields

$$\begin{aligned} L(G * f) &= \sum_{|\alpha|\leq m} A_\alpha D^\alpha (G * f) = \sum_{|\alpha|\leq m} A_\alpha (D^\alpha G * f) = \\ &= \left(\sum_{|\alpha|\leq m} A_\alpha D^\alpha G\right) * f = (LG) * f = \\ &= \delta * f = f, \end{aligned} \tag{2.6}$$

which immediately gives that if (2.5) holds then $u = G * f$ and $L^{-1} = G*$ in some sense. The function G is called the Green's function, or the fundamental solution, corresponding to the differential operator L and the given boundary conditions. This is a very profound result and its importance cannot be stressed enough. Given L and the boundary conditions one can obtain the corresponding Green's function and thus solve every equation on the form $Lu = f$, with the same boundary conditions, by calculating the integral $u = G * f$. [5]

2.1 Propagators and Green's Functions

This derivation mainly follows [6] and [7]. Consider the time-dependent Schrödinger equation in natural units

$$\left(H - i\frac{\partial}{\partial t}\right)\psi \quad = \quad 0, \tag{2.7}$$

where H is the time-independent Hamiltonian. Suppose that H generates a complete orthonormal set of eigenfunctions $H\phi_n = E_n\phi_n$ so that the arbitrary wavefunction ψ can be expanded as, $\psi = \sum_n c_n\phi_n$. Inserting this into (2.7) yields

$$\sum_n \left(E_n c_n - i\frac{\partial c_n}{\partial t}\right)\phi_n \quad = \quad 0, \tag{2.8}$$

because H is linear. Due to orthogonality the expression within the parentheses must vanish,

$$E_n c_n - i\frac{\partial c_n}{\partial t} \quad = \quad 0, \forall n, \tag{2.9}$$

which gives

$$\psi(\boldsymbol{r}, t) \quad = \quad \sum_n c_n(0)\phi_n(\boldsymbol{r})e^{-iE_n t}. \tag{2.10}$$

Now, assuming that $\psi(\boldsymbol{r}, 0)$ is known $c_n(0)$ can be determined by once again utilizing that ϕ_n form a complete orthogonal set. Simply act on (2.10) with ϕ_m^* and integrate over all of space, giving

$$c_n(0) \quad = \quad \int \mathrm{d}^3\boldsymbol{r}\phi_n^*(\boldsymbol{r})\psi(\boldsymbol{r}, 0). \tag{2.11}$$

Noticing that the choice of initial time $t = 0$ is arbitrary, and that the wavefunction only depends on the time difference, one can safely set the initial time as $t_0 = t'$ and do the substitution $t \to t - t'$. Combining (2.11) and (2.10) and now yields

$$\psi(\boldsymbol{r}, t) \quad = \quad \sum_n e^{-iE_n(t-t')}\phi_n(\boldsymbol{r}) \int d^3\boldsymbol{r}'\phi_n^*(\boldsymbol{r}')\psi(\boldsymbol{r}', t') \tag{2.12}$$

or

$$\psi(\boldsymbol{r}, t) \quad = \quad \int d^3\boldsymbol{r}\tilde{G}(\boldsymbol{r}, \boldsymbol{r}', t, t')\psi(\boldsymbol{r}', t') \tag{2.13}$$

defining the quantity

$$\tilde{G}(\boldsymbol{r},\boldsymbol{r}',t,t') \quad = \quad \sum_n e^{iE_n(t-t')}\phi_n(\boldsymbol{r})\phi_n^*(\boldsymbol{r}'), \tag{2.14}$$

which is known as a propagator as it propagates a state from $\boldsymbol{r}'$ at time t' to $\boldsymbol{r}$ at time t. This is even more clear using Dirac's bra-ket notation. Now, $\tilde{G}$ is considered an operator

$$\tilde{G}(t,t') \quad = \quad \sum_n e^{iE_n(t-t')}|\phi_n\rangle\langle\phi_n| = e^{iH(t-t')}, \tag{2.15}$$

which is nothing but the familiar time evolution operator. Furthermore, consider the evolution of a point source $\psi(\boldsymbol{r}',0) = \delta(\boldsymbol{r}')$. This gives that $\tilde{G}$ is itself a solution to the Schrödinger equation, $\psi(\boldsymbol{r},t) = \tilde{G}(\boldsymbol{r},\boldsymbol{r},t,0)$ and implies that it is not, strictly speaking, a Green's function. To be a proper Green's function remember that (2.5) must be fulfilled, in this case

$$\left(H - i\frac{\partial}{\partial t}\right)G \quad = \quad \delta^{(3)}(\boldsymbol{r}-\boldsymbol{r}')\delta(t-t'). \tag{2.16}$$

$\tilde{G}$ only gives the spatial delta function. This is easily seen from equation (2.13) letting $t \to t'$; if no time passes the state will not change, thus $\tilde{G}(\boldsymbol{r},\boldsymbol{r}',t,t' = t) = \delta^{(3)}(\boldsymbol{r}-\boldsymbol{r}')$, i.e. the completeness relation restated. Recall that a Green's function is the inverse of a linear operator obeying certain boundary conditions. Taking the Fourier transform of (2.16) with respect to the time difference yields

$$(H-\omega)\,G(\omega) \quad = \quad 1 \tag{2.17}$$

in Dirac's bra-ket notation. Remember that $G(\boldsymbol{r},\boldsymbol{r}';\omega) = \langle\boldsymbol{r}|G(\omega)|\boldsymbol{r}'\rangle$ so there should not be any spatial dependence in (2.17). Furthermore, inserting a complete set of eigenfunctions corresponding to H in (2.17) and using that for any well behaved function F it holds that $F(H)|\phi_n\rangle = F(E_n)|\phi_n\rangle$, one gets

$$G(\omega) \quad = \quad \sum_n \frac{|\phi_n\rangle\langle\phi_n|}{E_n-\omega}. \tag{2.18}$$

Here care must be taken. When transforming back to the time domain one must do so introducing a small complex parameter, $i\varepsilon$, in the denominator, which later will be permitted to tend to zero, to avoid the pole. The inverse Fourier transform is made with a contour integral enclosing the real axis. This can be done in two ways, either from above or from below. The sign of the small complex parameter therefore determines which half-plane the semicircle part of the contour must lie in. The inverse Fourier transform of (2.18) is

$$G(t,t') \quad = \quad \int_{-\infty}^{\infty}\frac{\mathrm{d}\omega}{2\pi}e^{i\omega(t-t')}\sum_n\frac{|\phi_n\rangle\langle\phi_n|}{E_n-\omega\pm i\varepsilon}. \tag{2.19}$$

Closing the contour and assuming that the semicircle part goes to zero as its radius goes to infinity one is left with

$$G(t,t') \quad = \quad \lim_{\omega\to E_n\pm i\varepsilon} 2\pi i\sum_n\frac{e^{i\omega(t-t')}}{2\pi}|\phi_n\rangle\langle\phi_n|. \tag{2.20}$$

But this is not the whole story. When writing the semicircle integral one writes $\omega = Re^{i\varphi}$ where $\varphi \in [0, \pm\pi]$ and $R \to \infty$ eventually. This gives that the integrand in (2.19) will be proportional to $\exp\left[i(t-t')Re^{i\varphi}\right]$ and it is therefore of grave importance that the time difference is positive when enclosing the upper pole, and negative when enclosing the lower pole. Otherwise the integral diverges instead of vanishing. One must therefore introduce Heaviside's step function to ensure that this criteria is met. Thus (2.20) has two forms,

$$G^R(t,t') = i\theta(t-t')\sum_n e^{iE_n(t-t')}|\phi_n\rangle\langle\phi_n| \tag{2.21}$$

and

$$G^A(t,t') = -i\theta(t'-t)\sum_n e^{iE_n(t-t')}|\phi_n\rangle\langle\phi_n|, \tag{2.22}$$

where the extra minus sign in G^A is due to taking the contour clockwise. These functions are called retarded and advanced Green's function respectively. Although the advanced one is used in some calculations, the physical one is G^R because a present state does not depend on future states, but past ones; hence the name *retarded* (in the meaning delayed). G^R is causal and G^A is anti-causal.

Looking back on this section one sees that G^R and G^A are indeed proper Green's functions and are related to $\tilde{G}$ by

$$\tilde{G} = -i\left(G^R - G^A\right), \tag{2.23}$$

$$G^R = i\theta(t-t')\tilde{G}, \tag{2.24}$$

$$G^A = -i\theta(t'-t)\tilde{G}. \tag{2.25}$$

It is therefore no surprise that $\tilde{G}$ did not turn out to be a proper Green's function as it, in reality, is the difference between two Green's functions.

2.2 Second Quantization

To be able to understand the formulation of the BCS theory one needs to first study some elementary quantum field theory (QFT), or more precisely, the second quantization. The term *second* quantization is somewhat misleading as the theory is quantized only *once*. The confusion stems from the fact that early quantum mechanics were the subject of the quantization of particles, whereas second quantization were the subject of quantization of fields. Quantizing a relativistic particle one obtains the Klein-Gordon equation, which the wavefunction must satisfy. However, this equation is the classical equation of motion for a relativistic field, thus quantizing a field superficially looks like quantizing a classical particle twice. [8]

When quantizing a particle in a harmonic oscillator, creation and annihilation operators are introduced ($a^\dagger$ and a respectively) with the only non-vanishing commutation relation $\left[a, a^\dagger\right] = 1$ [9]. More or less the same thing is done in QFT. When quantizing the Klein-Gordon field, every Fourier mode is treated as an independent harmonic oscillator, having separate $a^\dagger$ and a. However, the Klein-Gordon field is a bosonic quantum field and we are interested in electrons, thus we need to analyze the Dirac field. Analogously to the Klein-Gordon field,

the Dirac field is expanded in Fourier modes with separate creation and annihilation operators, but as it turns out, it is not a scalar field, not even a 2-spinor field as one might first expect, but a 4-spinor field. Thus, two separate sets of creation (annihilation) operators are needed for each mode, $a^\dagger_{\boldsymbol{p}\sigma}$ ($a_{\boldsymbol{p}\sigma}$) creating (annihilating) an electron with momentum $\boldsymbol{p}$ and spin σ and $b^\dagger_{\boldsymbol{p}\sigma}$ ($b_{\boldsymbol{p}\sigma}$) doing the same thing but for a *positron* (i.e. the electron's corresponding antiparticle). As hinted previously, the Dirac field is not a bosonic quantum field but a fermionic one (as it describes the electron), thus the operators must obey the canonical equal times anticommutation relations

$$\left\{a_{\boldsymbol{p}\rho}, a^\dagger_{\boldsymbol{q}\sigma}\right\} = \left\{b_{\boldsymbol{p}\rho}, b^\dagger_{\boldsymbol{q}\sigma}\right\} = (2\pi)^3\, \delta^{(3)}\left(\boldsymbol{p}-\boldsymbol{q}\right) \delta^{\rho\sigma}, \tag{2.26}$$

with all other anticommutators vanishing. This is simply the Pauli exclusion principle restated, implying

$$a^\dagger_{\boldsymbol{p}\sigma} a^\dagger_{\boldsymbol{p}\sigma} \left|0\right\rangle = b^\dagger_{\boldsymbol{p}\sigma} b^\dagger_{\boldsymbol{p}\sigma} \left|0\right\rangle = 0. \tag{2.27}$$

For the interested reader, see [10] and [11].

A field is expanded and the Fourier modes are promoted to operators obeying the appropriate commutation or anticommutation relations. In terms of field operators one would thus write (2.10) as

$$\psi^\dagger(\boldsymbol{r},t) = \sum_n c^\dagger_n \phi^*_n(\boldsymbol{r}) e^{iE_n t}, \tag{2.28}$$

$$\psi(\boldsymbol{r},t) = \sum_n c_n \phi_n(\boldsymbol{r}) e^{-iE_n t}. \tag{2.29}$$

where $\left\{c_m, c^\dagger_n\right\} = \delta_{mn}$ for fermions. Skipping ahead slightly, the Green's functions of interest are defined as

$$\tilde{G}(\boldsymbol{r},\boldsymbol{r}',t,t') = \left\langle 0\right| \left\{\psi(\boldsymbol{r},t), \psi^\dagger(\boldsymbol{r}',t')\right\} \left|0\right\rangle \tag{2.30}$$

$$G^R(\boldsymbol{r},\boldsymbol{r}',t,t') = \theta(t-t') \left\langle 0\right| \left\{\psi(\boldsymbol{r},t), \psi^\dagger(\boldsymbol{r}',t')\right\} \left|0\right\rangle \tag{2.31}$$

$$G^A(\boldsymbol{r},\boldsymbol{r}',t,t') = -\theta(t'-t) \left\langle 0\right| \left\{\psi(\boldsymbol{r},t), \psi^\dagger(\boldsymbol{r}',t')\right\} \left|0\right\rangle \tag{2.32}$$

$$G(\boldsymbol{r},\boldsymbol{r}',t,t') = \left\langle 0\right| T\psi(\boldsymbol{r},t)\psi^\dagger(\boldsymbol{r}',t') \left|0\right\rangle \tag{2.33}$$

where

$$T\psi(\boldsymbol{r},t)\psi^\dagger(\boldsymbol{r}',t') = \begin{cases} \psi(\boldsymbol{r},t)\psi^\dagger(\boldsymbol{r}',t') & , \quad t > t' \\ -\psi^\dagger(\boldsymbol{r}',t')\psi(\boldsymbol{r},t) & , \quad t < t' \end{cases} \tag{2.34}$$

is the time ordering operator which orders operators chronologically from left to right. To show that (2.30) - (2.32) are indeed the second quantization form of (2.23) - (2.25) it is sufficient to show that (2.30) equals (2.23). This is trivially done using the anticommutation relation for the field

$$\begin{aligned} \left\langle 0\right| \left\{\psi(\boldsymbol{r},t), \psi^\dagger(\boldsymbol{r}',t')\right\} \left|0\right\rangle 5 &= \left\langle 0\right| \sum_n e^{-iE_n(t-t')} \phi_n(\boldsymbol{r}) \phi^*_n(\boldsymbol{r}') \left|0\right\rangle = \\ &= \sum_n e^{-iE_n(t-t')} \phi_n(\boldsymbol{r}) \phi^*_n(\boldsymbol{r}'). \end{aligned} \tag{2.35}$$

2.3 Finite Temperatures

When doing the transition from single-particle Green's function at zero temperature, where the expectation value is taken with respect to the vacuum, to many-particle Green's function at finite temperatures one uses the notion of the grand canonical ensemble. The density matrix was implicitly taken to be pure in the previous section. Now probabilities are assigned to each possible state according to

$$p_i = \frac{e^{-\beta E_i}}{\mathrm{Tr}\left(e^{-\beta H}\right)}, \tag{2.36}$$

where $\beta^{-1} = k_B T$ (although, henceforth $k_B = 1$) and $\mathrm{Tr}(A) = \sum_i \langle i|A|i\rangle$. The energy is measured from the chemical potential, thus

$$\rho = \frac{\sum_i e^{-\beta E_i}|\phi_i\rangle\langle\phi_i|}{\mathrm{Tr}\left(e^{-\beta H}\right)} = e^{\beta\Omega}e^{-iHt}, \tag{2.37}$$

where $e^{-\beta\Omega} = \mathrm{Tr}\left(e^{-\beta H}\right)$. This means that the Green's functions now involve both e^{iHt} and $e^{-\beta H}$as factors. It is therefore very attractive to be able to treat time and temperature on equal footing. What Matsubara did was to treat time as a complex temperature. This is done by setting $t \to -i\tau$. It turns out that the Green's function is periodic in τ and one can thus treat it as lying solely within this period, i.e. $\tau \in [-\beta, \beta]$. Because τ now only is defined in a certain range it is tempting to expand the Green's functions in their Fourier series.

$$G(\tau) = T\sum_n G(\varepsilon_n)e^{-in\pi\tau/T}, \tag{2.38}$$

$$G(\varepsilon_n) = \frac{1}{2}\int_{-\beta}^{\beta} G(\tau)e^{in\tau/T}\mathrm{d}\tau. \tag{2.39}$$

where I have once again skipped ahead a little; the definition of ε_n will be given soon. Now, these are fermionic Green's functions which implies that $G(\tau)$ is an odd function, $G(\tau) = -G(\tau+\beta)$ when $\tau \in (-\beta, 0)$. One can therefore separate the interval in (2.39),

$$\begin{aligned} G(\varepsilon_n) &= \frac{1}{2}\left[\int_0^{\beta} G(\tau)e^{in\tau/T}\mathrm{d}\tau + \int_{-\beta}^{0} G(\tau)e^{in\tau/T}\mathrm{d}\tau\right] = \\ &= \frac{1}{2}\left(1 - e^{in\pi}\right)\int_0^{\beta} G(\tau)e^{in\tau/T}\mathrm{d}\tau. \end{aligned} \tag{2.40}$$

Clearly $G(\varepsilon_n)$ is only nonzero when n is odd. Defining $\varepsilon_n = \pi T(2n+1)$, which are called the Matsubara frequencies, one can thus write (2.38) and (2.39) as

$$G(\tau) = T\sum_n G(\varepsilon_n)e^{-i\tau\varepsilon_n}, \tag{2.41}$$

$$G(\varepsilon_n) = \int_0^{\beta} G(\tau)e^{i\tau\varepsilon_n}\mathrm{d}\tau. \tag{2.42}$$

To retrieve the Green's function of interest one only needs to do the replacement $i\varepsilon_n \to \varepsilon + i\delta$, where δ is an infinitesimal, called analytic continuation. The

advantage of the Matsubara procedure is that operators can be (complex) time ordered. This could also have been done by treating temperature as a complex time instead of the other way around as done above, but it is easier to use Matsubara Green's functions as soon as one has digested the conceptual difficulties of using time as a complex temperature. [12]

Chapter 3

The BCS Theory of Superconductivity

The microscopic theory of superconductivity was first formulated by Bardeen, Cooper and Schrieffer in 1957, hence its name, the BCS theory. They proposed an attractive electron-electron interaction mediated through phonons at low temperatures. This causes the electrons to pair up, creating so called Cooper pairs. These pairs have lower energy than twice the Fermi energy, $E < 2E_F$, and therefore the Fermi sea is unstable to the creation of at least one Cooper pair, no matter how weak the pairing potential is. An intuitive picture of Cooper pair creation is that when $T < T_c$ an electron creates a bulge in the lattice because of the attractive electron-ion interaction. This bulge can in turn attract another electron. This effect is the strongest when the two electrons are time-reversed version of each other, i.e. they have opposite spin and momenta, which allows them to utilize each others bulge in the lattice. [11]

3.1 The Energy Gap

The BCS Hamiltonian expressed in terms of second quantization is

$$H = \sum_{\boldsymbol{k},\sigma} \xi_{\boldsymbol{k}} c^\dagger_{\boldsymbol{k}\sigma} c_{\boldsymbol{k}\sigma} + \sum_{\boldsymbol{k},\boldsymbol{k}'} \lambda_{\boldsymbol{k}\boldsymbol{k}'} c^\dagger_{\boldsymbol{k}\uparrow} c^\dagger_{-\boldsymbol{k}\downarrow} c_{-\boldsymbol{k}'\downarrow} c_{\boldsymbol{k}'\uparrow}, \tag{3.1}$$

where $\lambda_{\boldsymbol{k}\boldsymbol{k}'}$ is the attraction between the electrons and $\xi_{\boldsymbol{k}}$ is the energy measured relative the chemical potential. The interaction term only pairs two electrons of opposite spin and momenta and there are, of course, many more terms one could include, but they are not necessary for a qualitative understanding of superconductivity. [1]

In this thesis only s-wave superconductors are considered. This means that the pairing is constant in momentum space, i.e. $\lambda_{\boldsymbol{k}\boldsymbol{k}'} = \lambda < 0$. Defining the so called order parameter, $\Delta = -\lambda \sum_{\boldsymbol{k}} \langle c_{-\boldsymbol{k}\downarrow} c_{\boldsymbol{k}\uparrow} \rangle$ (which is assumed to be nonzero due to the existence of the Cooper pair condensate) and will be shown to be the energy gap of the superconductor, the Hamiltonian can be written as

$$H = \sum_{\boldsymbol{k},\sigma} \xi_{\boldsymbol{k}} c^\dagger_{\boldsymbol{k}\sigma} c_{\boldsymbol{k}\sigma} - \sum_{\boldsymbol{k}} \left(\Delta^*_{\boldsymbol{k}} c^\dagger_{\boldsymbol{k}\uparrow} c^\dagger_{-\boldsymbol{k}\downarrow} + \Delta_{\boldsymbol{k}} c_{-\boldsymbol{k}\downarrow} c_{\boldsymbol{k}\uparrow} \right), \tag{3.2}$$

where a constant term is omitted as it effectively only shifts the chemical potential. This Hamiltonian is not diagonal, but it is diagonalizable by doing a Bogoliubov transformation

$$\begin{pmatrix} \gamma_{\boldsymbol{k}\uparrow} \\ \gamma^{\dagger}_{-\boldsymbol{k}\downarrow} \end{pmatrix} = \begin{pmatrix} u^{*}_{\boldsymbol{k}} & v_{\boldsymbol{k}} \\ -v^{*}_{\boldsymbol{k}} & u_{\boldsymbol{k}} \end{pmatrix} \begin{pmatrix} c_{\boldsymbol{k}\uparrow} \\ c^{\dagger}_{-\boldsymbol{k}\downarrow} \end{pmatrix} \tag{3.3}$$

where the coefficients $u_{\boldsymbol{k}}$ and $v_{\boldsymbol{k}}$ are related to the expansion coefficients in the BCS ground state

$$|\psi_{BCS}\rangle \quad = \quad \prod \boldsymbol{k} \left(u_{\boldsymbol{k}} + v_{\boldsymbol{k}} c^{\dagger}_{\boldsymbol{k}\uparrow} c^{\dagger}_{-\boldsymbol{k}\downarrow} \right) |0\rangle \ , \tag{3.4}$$

at $T = 0$ where they are equal. It holds that $|u_{\boldsymbol{k}}|^2 + |v_{\boldsymbol{k}}|^2 = 1$ and they correspond to the probability of a Cooper pair formed by electrons with momentum $|\boldsymbol{k}|$ being occupied, or not occupied, respectively. Explicitly they are

$$|u_{\boldsymbol{k}}|^2 = \frac{1}{2}\left(1 + \frac{\xi_{\boldsymbol{k}}}{E_{\boldsymbol{k}}}\right) \quad \text{and} \quad |v_{\boldsymbol{k}}|^2 = \frac{1}{2}\left(1 - \frac{\xi_{\boldsymbol{k}}}{E_{\boldsymbol{k}}}\right) \ , \tag{3.5}$$

where $E_{\boldsymbol{k}} = \left(\xi^2_{\boldsymbol{k}} + |\Delta_{\boldsymbol{k}}|^2\right)^{\frac{1}{2}}$. Moving on, the Bogliubov transformation diagonalizes the Hamiltonian giving

$$H \quad = \quad \sum_{\boldsymbol{k}} E_{\boldsymbol{k}} \left(\gamma^{\dagger}_{\boldsymbol{k}\uparrow} \gamma_{\boldsymbol{k}\uparrow} + \gamma^{\dagger}_{\boldsymbol{k}\downarrow} \gamma_{\boldsymbol{k}\downarrow} \right) \tag{3.6}$$

which shows that Δ is indeed the energy gap of the superconductor as there are no quasiparticle excitations with energy in the interval $[-\Delta, \Delta]$. [11]

3.1.1 Thermodynamic Properties of Bulk *s*-wave Superconductors

The pairing potential in a homogeneous s-wave superconductor for $H \leq H_{\mathrm{c1}}$ at a given temperature has no spin, momentum or spatial dependence; Δ is simply a scalar. A remarkable prediction of the BCS theory is that all[1] superconductors have the same ratio of its zero temperature gap to critical temperature, i.e.

$$\frac{\Delta(0)}{T_c} \quad = \quad 1.76 \ . \tag{3.7}$$

In figure 3.1 one can see the predicted BCS temperature dependence of the gap function.

Another very important property of superconductors is the local density of states (LDOS). Within the energy gap there are, naturally, no allowed states. Because the number of states does not change when passing to the superconducting regime, what will happen is that the states now forbidden effectively move to edges of the gap. See the inset in figure the 3.1.

[1] All s-wave superconductors having phonon-mediated electron-electron interactions.

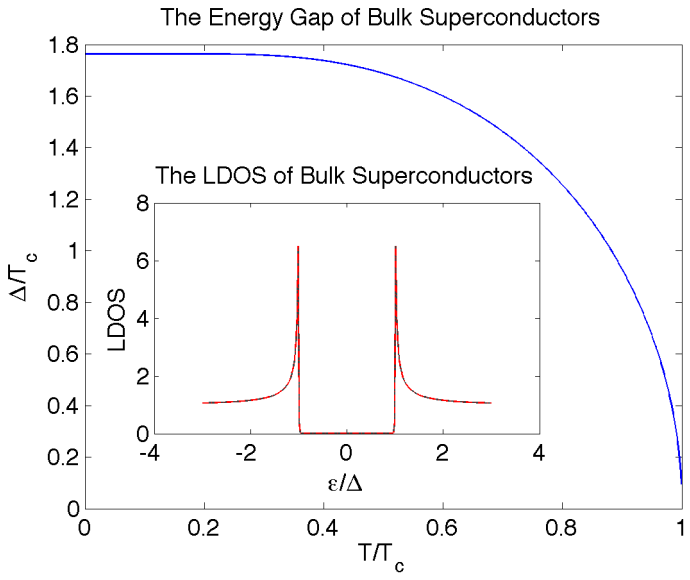


Figure 3.1: A plot showing the temperature dependence of the gap function, Δ in the BCS theory, as well as an inset of the LDOS of bulk s-wave superconductors.

3.2 The Gor'kov Equation

Returning to the BCS Hamiltonian, equation (3.1), following [13] one can rewrite it in terms of field operators

$$H = \int \mathrm{d}^3 r \left(-\psi_\alpha^\dagger \frac{\nabla^2}{2m} \psi_\alpha + \frac{\lambda}{2} \psi_\beta^\dagger \psi_\alpha^\dagger \psi_\alpha \psi_\beta \right) , \tag{3.8}$$

where an s-wave pairing is assumed. The (imaginary) time evolution of the field operators is

$$\frac{\partial \psi_\alpha}{\partial \tau} = \frac{\nabla^2}{2m} \psi_\alpha - \lambda \psi_\gamma^\dagger \psi_\gamma \psi_\alpha , \tag{3.9}$$

$$\frac{\partial \psi_\alpha^\dagger}{\partial \tau} = -\frac{\nabla^2}{2m} \psi_\alpha^\dagger + \lambda \psi_\alpha^\dagger \psi_\gamma^\dagger \psi_\gamma , \tag{3.10}$$

where the spatial and imaginary time dependence is omitted on all operators. Form the Green's functions

$$G_{\alpha\beta}(\tau, \boldsymbol{r}, \boldsymbol{r}') = \left\langle T_\tau \psi_\alpha(\tau, \boldsymbol{r}) \psi_\beta^\dagger(0, \boldsymbol{r}') \right\rangle , \tag{3.11}$$

$$G_{\alpha\beta}^\dagger(\tau, \boldsymbol{r}, \boldsymbol{r}') = \left\langle T_\tau \psi_\alpha^\dagger(\tau, \boldsymbol{r}) \psi_\beta(0, \boldsymbol{r}') \right\rangle . \tag{3.12}$$

To tidy up the notation, unprimed operators have a $(\tau, \boldsymbol{r})$-dependence and primed operators have a $(0, \boldsymbol{r}')$-dependence. The Green's functions are understood to depend on both unless otherwise noted. Evaluating their imaginary

time evolution gives

$$\frac{\partial G_{\alpha\beta}}{\partial \tau} = \delta_{\alpha\beta}\delta^{(3)}(\boldsymbol{r}-\boldsymbol{r}')\delta(\tau) + \left\langle T_\tau \frac{\partial \psi_\alpha}{\partial \tau}\psi_\beta'^\dagger)\right\rangle = \tag{3.13}$$

$$= \delta_{\alpha\beta}\delta^{(3)}(\boldsymbol{r}-\boldsymbol{r}')\delta(\tau) + \frac{\nabla^2}{2m}G_{\alpha\beta} - \lambda\left\langle \psi_\gamma^\dagger\psi_\gamma\psi_\alpha\psi_\beta'^\dagger\right\rangle \ , \tag{3.14}$$

where the spatial delta function comes from the equal time commutation relation of the field operators. The last term can be simplified using Wick's theorem which reduces it to products of pairs. Ignoring the terms on the form $\langle\psi\psi^\dagger\rangle\langle\psi\psi^\dagger\rangle$, as they only lead to a shift of the chemical potential, one is left with the term

$$\langle T_\tau\psi_\alpha\psi_\gamma\rangle\left\langle T_\tau\psi_\gamma^\dagger\psi_\beta'^\dagger\right\rangle \ . \tag{3.15}$$

As stated earlier, these factors are assumed to be nonzero in the superconducting state. Thus, defining the so called anomalous Green's functions

$$F_{\alpha\beta} = \left\langle T_\tau\psi_\alpha\psi_\beta'\right\rangle \ , \tag{3.16}$$

$$F_{\alpha\beta}^\dagger = \left\langle T_\tau\psi_\alpha^\dagger\psi_\beta'^\dagger\right\rangle \ , \tag{3.17}$$

and the order parameter

$$\Delta_{\alpha\beta}(\boldsymbol{r}) = -\lambda F_{\alpha\beta}(\boldsymbol{r}'=\boldsymbol{r},\tau=0) \ , \tag{3.18}$$

where the minus sign comes from the fact that $\lambda<0$, the equation of motion of $G_{\alpha\beta}$ can be written as

$$\left(\frac{\partial}{\partial\tau}-\frac{\nabla^2}{2m}\right)G_{\alpha\beta} + \Delta_{\alpha\gamma}F_{\gamma\beta}^\dagger = \delta_{\alpha\beta}\delta^{(3)}(\boldsymbol{r}-\boldsymbol{r}')\delta(\tau) \ . \tag{3.19}$$

Similarly, the equations of motion for the other Green's functions, both normal and anomalous, are

$$-\left(\frac{\partial}{\partial\tau}+\frac{\nabla^2}{2m}\right)G_{\alpha\beta}^\dagger + \Delta_{\alpha\gamma}^*F_{\gamma\beta} = \delta_{\alpha\beta}\delta^{(3)}(\boldsymbol{r}-\boldsymbol{r}')\delta(\tau) \ , \tag{3.20}$$

$$\left(-\frac{\partial}{\partial\tau}+\frac{\nabla^2}{2m}\right)F_{\alpha\beta} + \Delta_{\alpha\gamma}G_{\gamma\beta}^\dagger = 0 \ , \tag{3.21}$$

$$\left(\frac{\partial}{\partial\tau}+\frac{\nabla^2}{2m}\right)F_{\alpha\beta}^\dagger + \Delta_{\alpha\gamma}^*G_{\gamma\beta} = 0 \ . \tag{3.22}$$

As is evident in the equations above, (3.18) is a self-consistency equation for the order parameter because the equation of motion of $F_{\alpha\beta}$ contains $\Delta_{\alpha\beta}$. Equation (3.19) - (3.22) can be expressed in a much more condensed form in Nambu space, i.e. particle-hole space, defining the matrix Green's function

$$\hat{G} = \begin{pmatrix} G_{\alpha\beta} & F_{\alpha\beta} \\ -F_{\alpha\beta}^\dagger & G_{\alpha\beta}^\dagger \end{pmatrix} \tag{3.23}$$

where the entries are 2×2 spin matrices, and

$$\hat{\Delta} = \begin{pmatrix} 0 & \Delta_{\alpha\beta} \\ -\Delta_{\alpha\beta}^* & 0 \end{pmatrix} \ , \tag{3.24}$$

and finally

$$\hat{\tau}_3 = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, \tag{3.25}$$

which is simply the Nambu space equivalent of σ_3 in spin space, and, $\hat{1}$, the identity in Nambu space gives the very compact matrix Gor'kov equation

$$\left(\hat{\tau}_3 \frac{\partial}{\partial \tau} + \hat{1}\frac{\nabla^2}{2m} - \hat{\Delta}\right) \otimes \hat{G} = \hat{1}\,, \tag{3.26}$$

where $\otimes$ implies matrix multiplication as well as derivatives [14]

$$(A \otimes B)(X,p) = \exp\left[i\frac{\partial^A}{\partial X}\frac{\partial^B}{\partial p} - i\frac{\partial^A}{\partial p}\frac{\partial^B}{\partial X}\right] A(X,p)B(X,p)\,. \tag{3.27}$$

Furthermore, going to frequency and momentum space in the center of mass frame it has the form

$$\left(\varepsilon\hat{\tau}_3 - \xi_p - \hat{\Delta}\right) \otimes \hat{G}(\boldsymbol{p},\boldsymbol{R}) = \hat{1}\,. \tag{3.28}$$

Likewise, for the primed coordinates one has

$$\hat{G} \otimes \left(\varepsilon\hat{\tau}_3 - \xi_p - \hat{\Delta}\right)(\boldsymbol{p},\boldsymbol{R}) = \hat{1}\,. \tag{3.29}$$

3.3 The Quasi-classical Method

The full Green's function used above contains more information than one often needs. For example, it contains high energy parts that can be discarded assuming that they vary very little on the scale $\xi_p = v_F(|\boldsymbol{p}| - p_F) \in [-E_c, E_c]$ where $k_B T_c \ll E_c \ll E_F$. Separating the Green's function into low and high energy parts, $\hat{G} = \hat{G}_{\rm low} + \hat{G}_{\rm high} = \delta(\xi_p)\hat{g} + G_{\rm high}$, and integrating defines the quasi-classical matrix propagator

$$\hat{g}(\boldsymbol{p}_F,\boldsymbol{R};\varepsilon_n) = \frac{1}{a}\int_{-E_c}^{E_c} \mathrm{d}\xi_p \hat{\tau}_3 \hat{G}_{\rm low}(\boldsymbol{p}_F,\boldsymbol{R};\varepsilon_n) = \begin{pmatrix} g & f \\ \bar{f} & \bar{g} \end{pmatrix}, \tag{3.30}$$

where a is the spectral weight (which will eventually disappear) and $\hat{\tau}_3$, which is understood to be in particle-hole space, is introduced for future convenience. This procedure is then applied to the left and right Gor'kov equations (3.29) and (3.28). However, the term $\xi_p \otimes \hat{G}(\boldsymbol{p}_F,\boldsymbol{R})$ is in need of more massage because one does not want to have to know the momentum deviation from the Fermi surface included in the variable ξ_p. Gradient expanding this term to first order gives

$$\begin{aligned} \xi_p \otimes \hat{G}(\boldsymbol{p},\boldsymbol{R}) &= \exp\left[i\frac{\partial^{\xi_p}}{\partial \boldsymbol{R}}\frac{\partial^{\hat{G}}}{\partial \boldsymbol{p}} - i\frac{\partial^{\xi_p}}{\partial \boldsymbol{p}}\frac{\partial^{\hat{G}}}{\partial \boldsymbol{R}}\right]\xi_p\hat{G} \simeq \\ &\simeq \xi_p\hat{G} + \frac{i}{2}\left[\frac{\partial \xi_p}{\partial \boldsymbol{R}}\frac{\partial \hat{G}}{\partial \boldsymbol{p}} - \frac{\partial \xi_p}{\partial \boldsymbol{p}}\frac{\partial \hat{G}}{\partial \boldsymbol{R}}\right] = \\ &= \left(\xi_p - \frac{i}{2}\boldsymbol{v}_F \cdot \vec{\nabla}_{\boldsymbol{R}}\right)\hat{G}, \end{aligned} \tag{3.31}$$

and similarly

$$\hat{G} \otimes \xi_p(\boldsymbol{p}, \boldsymbol{R}) \simeq \left(\xi_p + \frac{i}{2}\boldsymbol{v}_F \cdot \vec{\nabla}_{\boldsymbol{R}}\right)\hat{G}. \tag{3.32}$$

Now, one subtracts (3.29) from (3.28) after having acted with $\hat{\tau}_3$ on the Green's functions in the equations from the left, and then integrates, exactly as when defining the quasi-classical propagator. This is the so called left-right trick, which gives

$$\begin{aligned}
\frac{1}{a}\int_{-E_c}^{E_c} \mathrm{d}\xi_p \left[\left(\varepsilon\hat{\tau}_3 - \xi_p - \hat{\Delta}\right) \otimes \hat{\tau}_3\hat{G} - \hat{\tau}_3\hat{G} \otimes \left(\varepsilon\hat{\tau}_3 - \xi_p - \hat{\Delta}\right)\right] &= 0 \\
\Longrightarrow \frac{1}{a}\int_{-E_c}^{E_c} \mathrm{d}\xi_p \left[\left(\varepsilon\hat{\tau}_3 - \hat{\Delta} - \xi_p\right)\hat{\tau}_3\hat{G} + \frac{i}{2}\boldsymbol{v}_F \cdot \vec{\nabla}_{\boldsymbol{R}}\hat{\tau}_3\hat{G} - \right. \\
\left. - \hat{\tau}_3\hat{G}\left(\varepsilon\hat{\tau}_3 - \hat{\Delta} - \xi_p\right) + \frac{i}{2}\boldsymbol{v}_F \cdot \vec{\nabla}_{\boldsymbol{R}}\hat{\tau}_3\hat{G}\right] = \\
= \frac{1}{a}\int_{-E_c}^{E_c} \mathrm{d}\xi_p \left[\left(\varepsilon\hat{\tau}_3 - \hat{\Delta}\right)\hat{\tau}_3\hat{G} - \hat{\tau}_3\hat{G}\left(\varepsilon\hat{\tau}_3 - \hat{\Delta}\right) + i\boldsymbol{v}_F \cdot \vec{\nabla}_{\boldsymbol{R}}\hat{\tau}_3\hat{G}\right] = \\
= \frac{1}{a}\int_{-E_c}^{E_c} \mathrm{d}\xi_p \left(\left[\varepsilon\hat{\tau}_3 - \hat{\Delta}, \hat{\tau}_3\hat{G}\right] + i\boldsymbol{v}_F \cdot \vec{\nabla}_{\boldsymbol{R}}\hat{\tau}_3\hat{G}\right) = 0 \\
\Longrightarrow \left[\varepsilon\hat{\tau}_3 - \hat{\Delta}, \hat{g}\right] + i\boldsymbol{v}_F \cdot \vec{\nabla}_{\boldsymbol{R}}\hat{g} = 0,
\end{aligned} \tag{3.33}$$

which is the Eilenberger transport equation for clean superconductors. However, the left-right trick removes the inhomogeneity of the Gor'kov equation. This information is re-inserted by demanding that the quasi-classical matrix propagator, $\hat{g}$, is subject to the normalization condition $\hat{g}^2 = -\pi^2\hat{1}$.

A neat way to parameterize $\hat{g}$ that automatically satisfies the normalization constraint is by introducing the two coherence functions γ and $\bar{\gamma}$ [15, 16]

$$\hat{g} = -i\pi \begin{pmatrix} (1-\gamma\bar{\gamma})^{-1} & 0 \\ 0 & (1-\bar{\gamma}\gamma)^{-1} \end{pmatrix} \begin{pmatrix} 1+\gamma\bar{\gamma} & 2\gamma \\ -2\bar{\gamma} & -1-\bar{\gamma}\gamma \end{pmatrix}. \tag{3.34}$$

Inserting this into the transport equation gives two coupled Riccati type equations for the coherence functions

$$\begin{cases} i\boldsymbol{v}_F \cdot \vec{\nabla}_{\boldsymbol{R}}\gamma + 2\varepsilon\gamma - \gamma\bar{\Delta}\gamma + \Delta = 0 \\ -i\boldsymbol{v}_F \cdot \vec{\nabla}_{\boldsymbol{R}}\bar{\gamma} - 2\varepsilon\bar{\gamma} - \bar{\gamma}\Delta\bar{\gamma} + \bar{\Delta} = 0 \end{cases}. \tag{3.35}$$

The Riccati equations are solved by an initial guess of the order parameter with the constraint that it satisfies the topology, that is it gives the correct phase winding far away from the origin. After obtaining γ and $\bar{\gamma}$, f is constructed by inserting the coherence functions in the expression for the quasi-classical propagator. The self-consistency equation for the order parameter (in an s-wave superconductor) is

$$\Delta(\boldsymbol{r}) = -\lambda f(\boldsymbol{r}' = \boldsymbol{r}, \tau = 0) \tag{3.36}$$

or in terms of Matsubara Green's functions

$$\Delta(\boldsymbol{r}) \quad = \quad -\lambda T \sum_{\varepsilon_n}^{\varepsilon_n \leq E_c} f(\boldsymbol{p}_F, \boldsymbol{R}; i\varepsilon_n) \; . \tag{3.37}$$

As previously stated, this only holds for an s-wave superconductor. In general the pairing can be momentum and spin dependent, in which case the self-consistency equation becomes [17]

$$\Delta_{\alpha\beta}(\boldsymbol{p}_F, \boldsymbol{R}) \quad = \quad T \int \mathrm{d}^2 \boldsymbol{p}'_F \lambda_{\alpha\beta\gamma\delta}(\boldsymbol{p}'_F, \boldsymbol{p}_F) \sum_{\varepsilon_n}^{\varepsilon_n \leq E_c} f_{\gamma\delta}(\boldsymbol{p}'_F, \boldsymbol{R}; \varepsilon_n) \; , \tag{3.38}$$

which is the equation being parallelized in this thesis. Thus the code can easily be generalized to For example, a d-wave superconductor has a momentum dependent pairing [13]

$$\lambda(\boldsymbol{p}'_F, \boldsymbol{p}_F) \quad = \quad -2\lambda_d \left(p_x^2 - p_y^2\right) \left(p_x'^2 - p_y'^2\right) \; . \tag{3.39}$$

3.4 Δ as a Step Function

When solving the Riccati equations for a trajectory parameterized by x one does not have to solve them in their entirety, but one can utilize the fact that the order parameter is only known at certain discrete points. Thus solving the equations analytically for $\Delta(x) = \Delta_0 + (\Delta_1 - \Delta_0)\,\theta(x)$ and evaluating the coherence functions at $x = h$, where h is the grid spacing of the position matrix, one gets the next matrix element along the trajectory.

The equations need to be solved separately in the two domains, $x < 0$ and $x > 0$, and then patched together at the origin. It is important to note that the coherence functions are sensitive to the direction of the Fermi velocity. γ needs to be solved in the same direction as $\boldsymbol{v}_F$ in the sense $\hat{\boldsymbol{n}} \cdot \boldsymbol{v}_F > 0$, and for $\tilde{\gamma}$ it holds that $\hat{\boldsymbol{n}} \cdot \boldsymbol{v}_F < 0$ keeping in mind that the $\boldsymbol{v}_F$ has to be the same as when calculating γ. Parameterizing the trajectory with x this means that one has to use $v_F = \hat{\boldsymbol{n}} \cdot \boldsymbol{v}_F$ and $-v_F = \hat{\boldsymbol{n}} \cdot \boldsymbol{v}_F$. Furthermore, expressing all energies in terms of $k_B T_c$ and all distances in $\xi_0 = \hbar v_F / (k_B T_c)$ a tidy notation is obtained as well as being the natural scale of the problem, giving high numerical accuracy. For a singlet state the equations are now simply

$$i\partial_x \gamma + 2\epsilon\gamma + \widetilde{\Delta}\gamma^2 + \Delta \quad = \quad 0, \tag{3.40}$$

$$-i\partial_x \widetilde{\gamma} - 2\epsilon\widetilde{\gamma} + \Delta\widetilde{\gamma}^2 + \widetilde{\Delta} \quad = \quad 0. \tag{3.41}$$

In the interval $x \in (-\infty, 0]$ one has $\gamma = \gamma_0$, i.e. the bulk value at the very beginning of the trajectory. This is the particular solution to (3.40),

$$\gamma_0 = -\frac{\Delta_0}{\epsilon + i\Omega_0} \quad , \quad \Omega_0 = \sqrt{\Delta_0 \widetilde{\Delta}_0 - \epsilon^2}. \tag{3.42}$$

A homogeneous solution is not necessary in this interval as it only gives another term which will turn out to be zero when matching the solutions at the origin. In $x \in [0, \infty)$ on the other hand, the homogeneous solution is not zero. As before one has the particular solution,

$$\gamma_1 = -\frac{\Delta_1}{\epsilon + i\Omega_1} \quad , \quad \Omega_1 = \sqrt{\Delta_1 \widetilde{\Delta}_1 - \epsilon^2}. \tag{3.43}$$

Now, the ansatz

$$\gamma \;=\; \gamma_1 + \frac{1}{y} \tag{3.44}$$

yields the following differential equation

$$y' - 2\Omega_1 y \;=\; -i\widetilde{\Delta}_1 \,, \tag{3.45}$$

which is separable and thus straightforward to solve. The solution is

$$y \;=\; \frac{\widetilde{\Delta}_1}{2i\Omega_1} + \frac{1}{2i\Omega_1 C}\exp\left[2\Omega_1 x\right], \tag{3.46}$$

where C is a constant of integration and the extra $2i\Omega$ in the second denominator is extracted from C because of later convenience. Inserting (3.46) into (3.44) and determining C by matching the two solutions at the origin finally gives γ over the entire range,

$$\gamma\left(x\right) \;=\; \begin{cases} \gamma_0 & , \quad x < 0 \\ \gamma_1 + \frac{2i\Omega_1 C \exp[-2\Omega_1 x]}{1-\widetilde{\Delta}_1 C \exp[-2\Omega_1 x]} & , \quad x > 0 \end{cases} , \tag{3.47}$$

with

$$C \;=\; \frac{\gamma_0 - \gamma_1}{2i\Omega_1 + \widetilde{\Delta}_1\left(\gamma_0 - \gamma_1\right)}. \tag{3.48}$$

Completely analogously for $\tilde{\gamma}$ one obtains

$$\widetilde{\gamma}\left(x\right) \;=\; \begin{cases} \widetilde{\gamma}_0 + \frac{2i\Omega_0 \widetilde{C} \exp[2\Omega_0 x]}{1+\Delta_0 \widetilde{C} \exp[2\Omega_0 x]} & , \quad x < 0 \\ \widetilde{\gamma}_1 & , \quad x > 0 \end{cases} , \tag{3.49}$$

with

$$\widetilde{C} \;=\; \frac{\widetilde{\gamma}_0 - \widetilde{\gamma}_1}{2i\Omega_0 - \Delta_0\left(\widetilde{\gamma}_0 - \widetilde{\gamma}_1\right)}. \tag{3.50}$$

Having solved the Riccati equations for a step function order parameter it is now easy to see how to implement this scheme as a means of calculating the coherence functions across an entire trajectory. First calculate γ_0 which is given by the boundary conditions, then calculate γ_1 from the next element in the array comprising Δ. Insert these values into (3.47), set $x = h$ and get $\gamma\left(h\right)$. Store this value as the second element of the γ array. Now repeat this process with $\gamma_0 \longleftarrow \gamma\left(h\right)$.

3.5 Results

Here the single vortex properties of s-wave superconductors are presented. All calculations were carried out on the graphics card using CUDA C. The details of these calculations are explained in the next chapter.

3.5.1 The Vortex Core Size

The result of the self-consistency equation for the order parameter, with winding number $n = 1$, is seen in figure 3.2. Because it is cylindrically symmetrical only trajectories through the vortex core centers are shown. One sees that the bulk energy gap deacreases with increasing tempereature, which was already shown in figure 3.1. Furthermore, because there is a sign change by necessity when passing through the center the energy gap is zero in the middle of the vortex core, i.e. the specimen is normal there. As seen in the figure the slope of the order parameter in the core increases with decreasing temperature. Thus when $T \to T_c$ the slope of the core goes to zero and the gap vanishes, as one would expect.

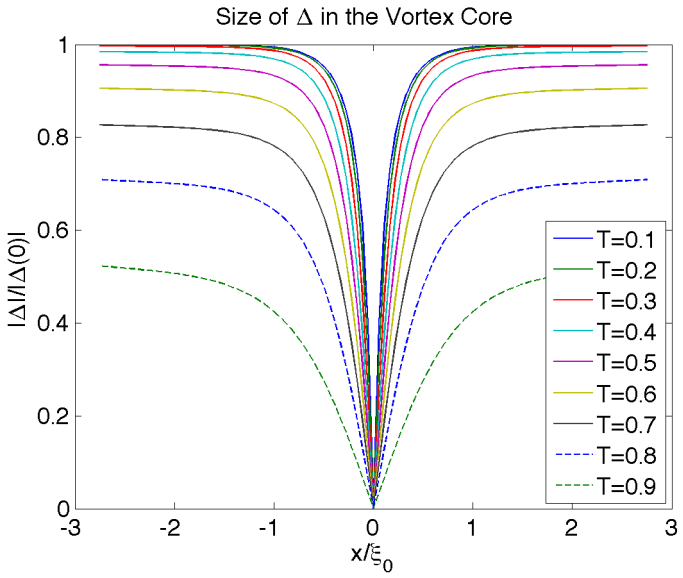


Figure 3.2: A plot showing the absolute value of the order parameter of a trajectory through the vortex core center. It is clearly visible that the core broadens at high temperatures. When $T \to T_c$ the slope of the order parameter in the core goes to zero and the gap vanishes. $\Delta(T)$ were solved0 self-consistently to an accuracy of 10^{-9} in norm between iterations. The order parameter is a 101×101 matrix with a physical spacing of $h = 0.0545\xi_0$ between the points. The number of Matsubara frequencies used were $n_c = 10/T$.

3.5.2 The Density of States

Unlike the LDOS in a bulk superconductor, seen in the inset of figure 3.1, the order parameter is not homogeneous when $H_{c1} \leq H \leq H_{c2}$, i.e. when the magnetic flux lines start penetrating the material. This gives that the LDOS will vary in space. Far away from the vortex the LDOS looks like the bulk one, but at the center, where the energy gap vanishes and the material is in the normal non-superconducting state, the situation is non-trivial. For example, in

the very center there is a small normal region[2] surrounded by an energy gap, thus one expects that there should exist bound low energy states there.

In figure 3.3 - 3.6 the LDOS of a trajectory passing through the vortex core center at different temperatures is seen. As expected there exists bound quasiparticle states in the center. But the form of the graphs needs to be explained. The broadening of the spectrum is given by the imaginary part in the analytic continuation $i\varepsilon_n \to \varepsilon + i\delta$, i.e. when switching from the Matsubara to the Keldysh representation. By adding $i\delta$ one attributes a phenomenological lifetime to the state and this is motivated by there not existing eternal states in nature.[3] Because δ is arbitrary one needs to match it to the spatial resolution of the order parameter. When δ is small the states are very sharp and the spectrum becomes extremely pointy. This is so because the angular and spatial resolution is finite and so the trajectories are missing states lying between them. Some pointiness can still be seen at $\varepsilon = 0$ in figure 3.4 and 3.6.

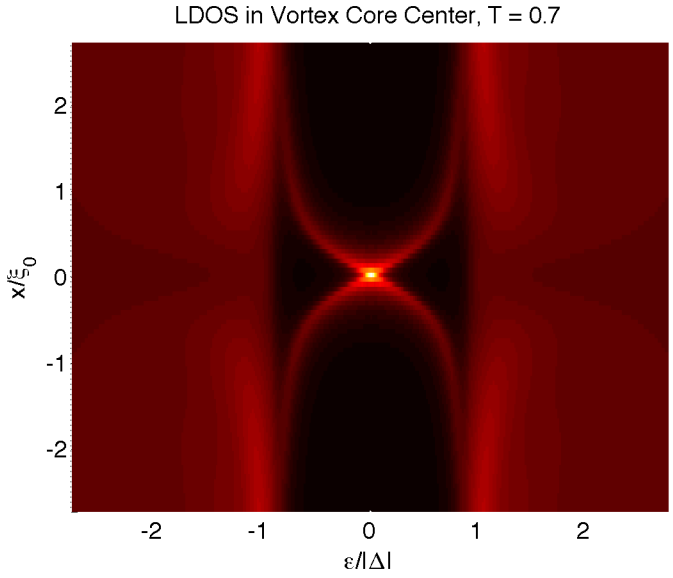


Figure 3.3: The local density of states (LDOS) for a trajectory passing through the vortex core center. In calculating the LDOS 333 energies in the interval $[-3\Delta, 3\Delta]$ and 1332 angles were used. The imaginary part of the energy, δ in $\epsilon \to \epsilon + i\delta$, was set to $\delta = 0.1$. The temperature was $T = 0.7T_c$.

3.5.3 The Kramer-Pesch Effect

As mentioned in the introduction, type II superconductors are identified by the ratio of two characteristic length scales, the penetration depth λ and the coherence length ξ together forming the dimensionless Ginzburg-Landau parameter $\kappa = \lambda/\xi_0$. As it turns out these length scales corresponds roughly to the shape of the supercurrent density, j_s around the vortex core. The magnitude of j_s rises linearly from the vortex center to $\xi_1 \sim \xi_0 \frac{T}{T_c}$ (ξ_0 being the BCS coherence

[2] In the idealized, perfectly clean superconductor only a single infinitesimal point is normal.
[3] Except ground states of course.

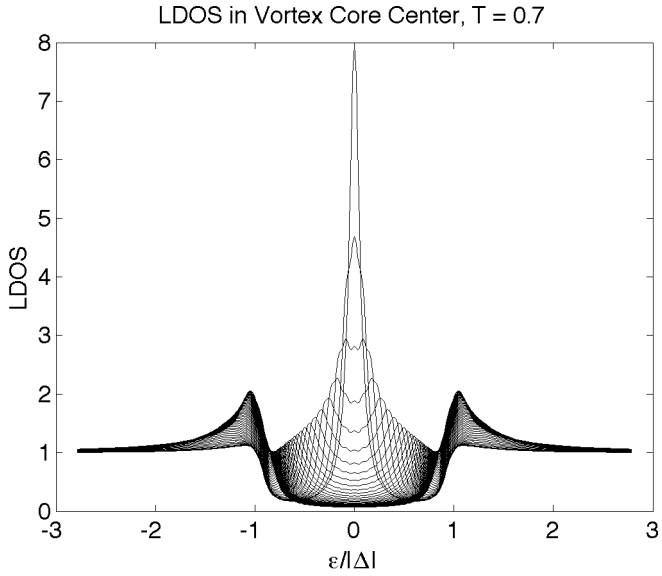


Figure 3.4: The local density of states (LDOS) for a trajectory passing through the vortex core center. This is the same LDOS as in 3.3 only viewed from the side.

length) where it dies off, exponentially decaying with the characteristic length λ. This maximum of the supercurrent density is temperature dependent and approaches the vortex core center when $T \longrightarrow 0$. The effect is known as the Kramer-Pesch effect, first described in 1974. [18]

The length related to the maximum of the supercurrent density is defined by [19]

$$\frac{1}{\xi_1} \quad = \quad \frac{1}{\Delta_0} \lim_{r \leftarrow 0} \frac{\Delta(r)}{r} \ , \tag{3.51}$$

where Δ_0 is the bulk energy gap. Thus ξ_1 is directly related to the reciprocal radial derivative of the order parameter at the core center. To reliably evaluate the derivative one needs a high resolution at the vortex core center, i.e. a large order parameter matrix in terms of the number of elements. It will be shown in chapter 4 that the order parameter dimensions directly translate to shared memory usage, which is the bottleneck of the graphics card used in this thesis. So one needs to be careful in defining the physical size, in coherence lengths not elements, of the order parameter. As the temperature decreases the vortex core shrinks thus allowing one to decrease the physical order parameter size as well.[4] A plot of $\xi_1(T)$ can be seen in figure 3.7.

[4]Or rather forcing one to decrease the physical order parameter size. Higher resolution is needed at lower temperatures and lower temperatures means that more Matsubara frequencies are needed. Thus keeping the same physical size increases the memory usage a lot and it is the memory that is the bottleneck of this problem.

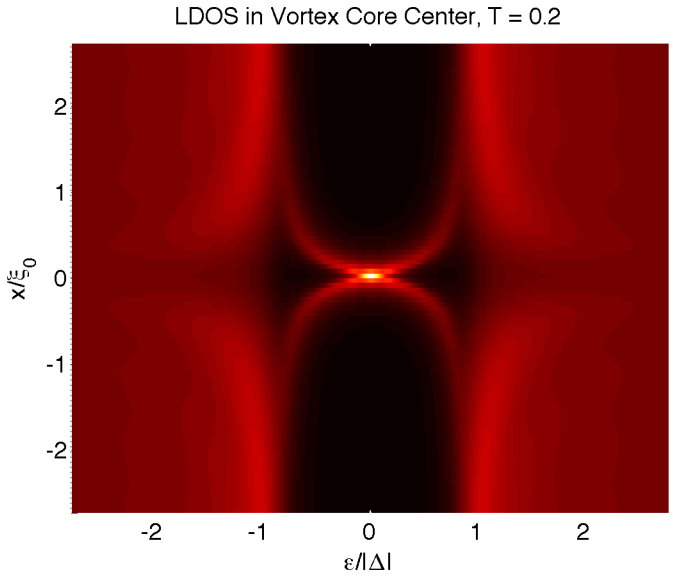


Figure 3.5: The local density of states (LDOS) for a trajectory passing through the vortex core center. In calculating the LDOS 333 energies in the interval $[-3\Delta, 3\Delta]$ and 1332 angles were used. The imaginary part of the energy, δ in $\epsilon \to \epsilon + i\delta$, was set to $\delta = 0.2$. The temperature was $T = 0.2T_c$.

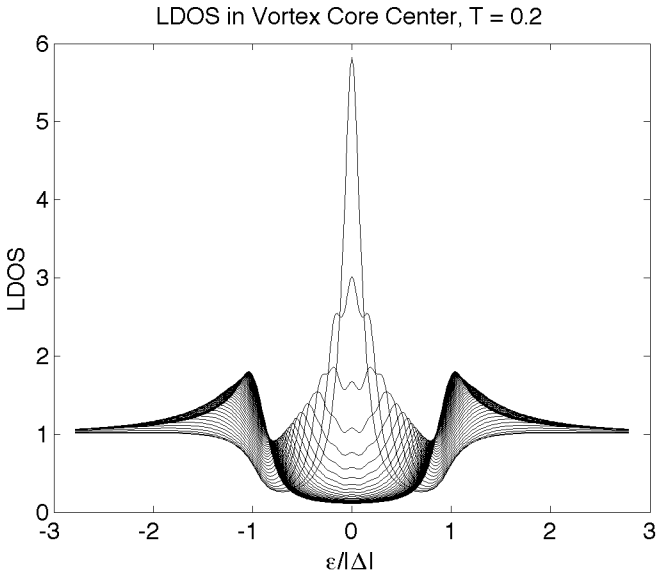


Figure 3.6: The local density of states (LDOS) for a trajectory passing through the vortex core center. This is the same LDOS as in 3.5 only viewed from the side.

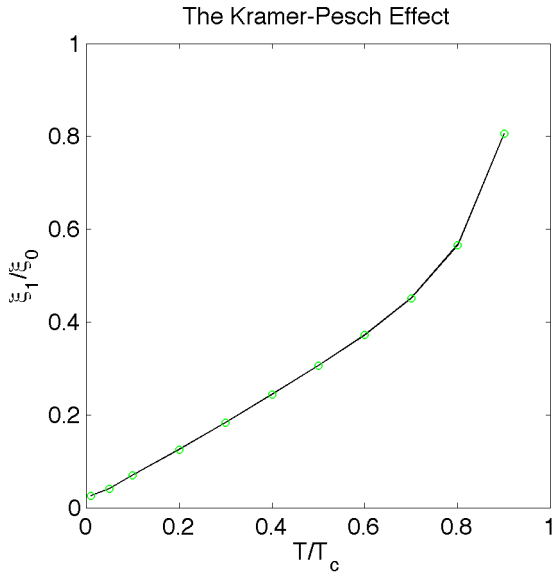


Figure 3.7: A plot showing the the Kramer-Pesch length scale ξ_1. The green circles indicate the temperatures where $\xi_1(T)$ has been calculated, the black line is only there to guide the eye.

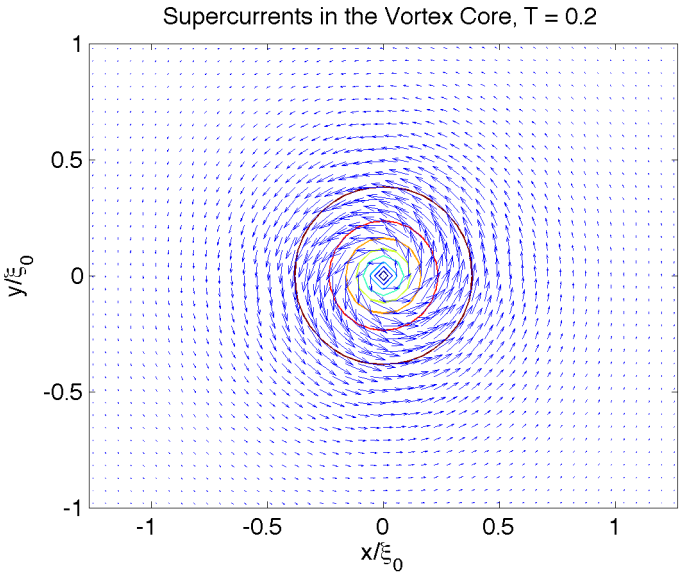


Figure 3.8: A quiver plot of the induced supercurrents superimposed on top of a contour plot of the absolute value of the order parameter. Here $T = 0.2T_c$. As indicated by 3.2 and 3.7 both the order parameter radius and the supercurrent radius is small.

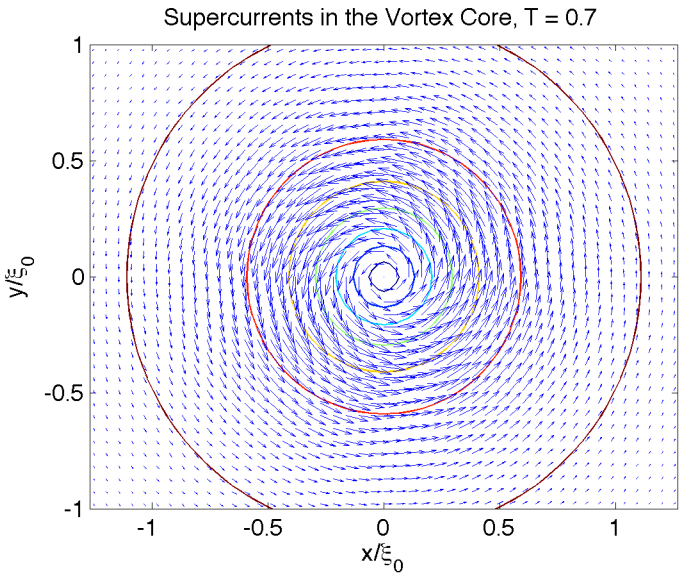


Figure 3.9: A quiver plot of the induced supercurrents superimposed on top of a contour plot of the absolute value of the order parameter. Here $T = 0.7T_c$. As indicated by 3.2 and 3.7 both the order parameter radius and the supercurrent radius is large.

Chapter 4

CUDA and Parallel Computing

Only ten years ago one needed a cluster of computers to be able to do parallel computing. Nowadays, even a cheap laptop has two CPU (central processing unit) cores and is therefore capable of carrying out true parallel computations. But it is the development of the GPUs (graphics processing unit) that has really brought parallelism to the masses. A modern GPU has between 8 and 400 cores and is capable of massively parallel computing. One difficult aspect with parallel programming is that one needs to change the way one usually thinks about algorithms. Some algorithms are very well suited to be parallelized while others are not. If a program has many threads which do not need to communicate (often) with each other, parallelization is easy and straightforward. But to really decrease computing time one needs to minimize the amount of copying of data from the host (CPU) to the device (GPU) and vice versa. This copying of data together with the GPUs small cache memory is the bottleneck of parallel programming on GPUs. Fortunately the self-consistency equation (3.38) is straightforward to parallelize and does not require vast amounts of memory. [20]

4.1 CUDA C

In 2006, NVIDIA released their CUDA architecture. CUDA is an acronym for Compute Unified Device Architecture and make NVIDIAs CUDA graphics cards accessible to programmers. The GPUs are thus not only designed to handle computer graphics, but also general purpose programming; hence the often used term GPGPU. Although there exists other competing GPU programming languages (OpenCL and CUDA Fortran for example) CUDA C was chosen as the appropriate language in this thesis because of the vast amount of material, tutorials and code samples available.

4.1.1 Terminology

- Host and device The host is the CPU on which the main program is run. A device is any GPU callable from the host.

- KERNEL The code being executed on the device. It is only callable from the host.
- THREAD The smallest unit of execution on the device.
- WARP A group of 32 threads, which is the smallest amount of data being processed,
- BLOCK A 3D array of threads. Threads within a block can communicate with each other.
- GRID The 2D array of blocks. Blocks cannot communicate with each other.
- BANK A unit of the memory which cannot be accessed in parallel. If two address calls occur in the same bank, their accesses are serialized.

4.2 The GPU Architecture

The parallel modern graphics cards grew out of the gaming industry's need to render high resolution real-time video, which is intrinsically a highly parallel task. The CPU has a completely different architecture than the GPU, seen in figure 4.1, which makes it unsuitable to render graphics. Unlike the CPU, a GPU has much more transistors dedicated to data processing than to flow control and data caching. Thus, if one has a problem involving the same computationally demanding code being executed multiple times on different data, for example many nested for-loops with independent iterations, i.e. one iteration does not depend on the previous ones, one can achieve a tremendous speed-up by moving these calculations from the host to the device. Not only will the GPU execute the code faster, it will also free the host while the device executes the code letting the host work on other tasks, in parallel with the device.

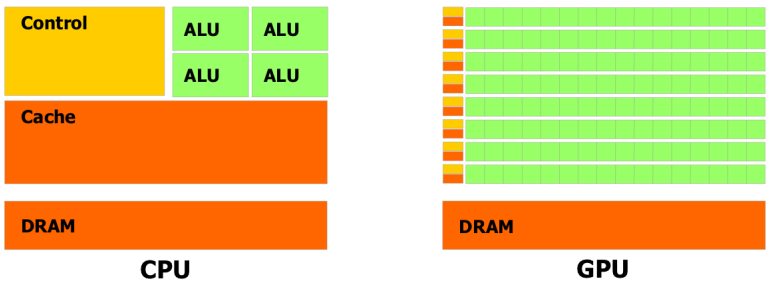


Figure 4.1: A schematic picture showing the difference in architecture of a general CPU an GPU. The CPU is designed to do a single task very fast, controlling the data flow and caching large amounts of data for fastest possible access. The GPU on the other hand is designed to carry out many parallel calculations with high arithmetic intensity, thus having more arithmetic logic units (ALU) and smaller cache memory. The size of the dynamic random access memory (DRAM) are typically of approximately the same size. Picture taken from [4].

4.2.1 Invoking the Kernel

The code being executed on the device is called the *kernel* and must include the identifier `__global__`, which tells the compiler that it should not be compiled using `gcc` or `g++`, but using `nvcc` which is NVIDIAS CUDA C compiler. The kernel is launched by the host and the specified data is copied from the host memory to the device memory. See figure 4.2. This copying introduces an overhead which makes parallel data processing unnecessary if the data being processed is small and/or the amount of arithmetic performed on the data is small. It is for large amounts of data and arithmetically intense code that parallelism really makes a difference. [21]

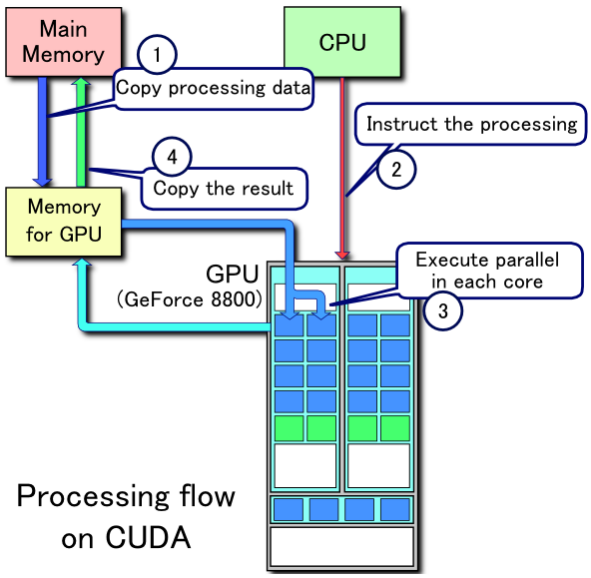


Figure 4.2: A schematic picture showing the processing flow of launching a CUDA kernel. First the data is copied from the host to the device. Then the host instructs the device what to do with the data. The kernel is executed in parallel on the device and the result is lastly copied from the device to the host. Picture taken from [22].

The kernel launch creates a grid of *blocks*, each running a number of *threads*. See figure 4.3.[1] The maximum number of threads in a block is 512 and the maximum number of blocks is 65535×65535. Each thread has access to different kinds of memories of different sizes. The registers and the shared memory are on-chip and thus extremely fast. The shared memory allows parallel access to the so called memory banks, which the global memory do not.

[1] For the Quadro FX 4800 card used in this thesis figure 4.3 is correct. However, for newer cards with compute capability 2.x the threads are organized as 3D and not 2D arrays.

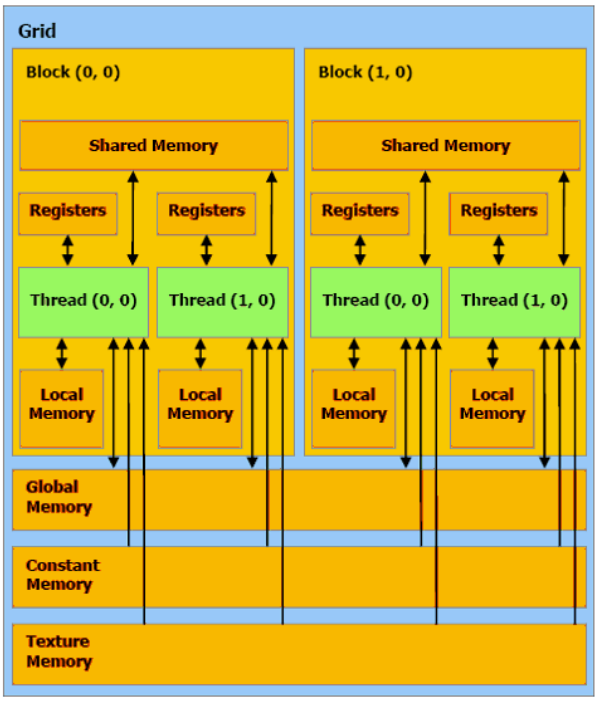


Figure 4.3: A picture from [4] showing how the kernel is divided into blocks and threads, and which memories the threads can access. The registers are fast, small and only accessible by one thread each. The shared memory is accessible by all threads within a block. The global memory is large and accessible by all threads in all blocks. When the host launches the kernel the data is copied to the global and constant memories.

4.3 Optimizing Performance

After writing the kernel, it is time to optimize it. At this point, it is important to analyze it carefully to see where the bottleneck is.[2] For example, if the bottleneck is the latency reading the global memory it is not important to start optimizing the instruction throughput. There are three ways one can tackle the problem of optimizing the kernel; parallelizing the problem as much as possible, optimizing memory usage and optimizing instruction throughput [4]. An approximation of the total speedup, SU, gained from parallelizing the code can be calculated using Amdahl's law,

$$\mathrm{SU} = \frac{1}{(1-p) + \frac{p}{S}}, \tag{4.1}$$

where p is the part (measured in time) of the serial program being parallelized and S is the speedup of that particular part, usually proportional to the number of parallel threads being executed. See figure 4.4. Amdahl's law does not take into account any overhead introduced by the parallelization, for example the

[2] *"Premature optimization is the root of all evil."* - Donald Knuth (or Tony Hoare)

copying of data between host and device, which means that the speedup gained in many circumstances is lower than indicated. On the other hand, it does not take memory caches into consideration either. If it is not only the available CPU clock cycles that are limited but also the memory, parallelization may allow more data to be cached and thus giving a speedup greater than that indicated by Amdahl's law. Anyhow, it gives a fairly good approximate upper limit in most cases. [23]

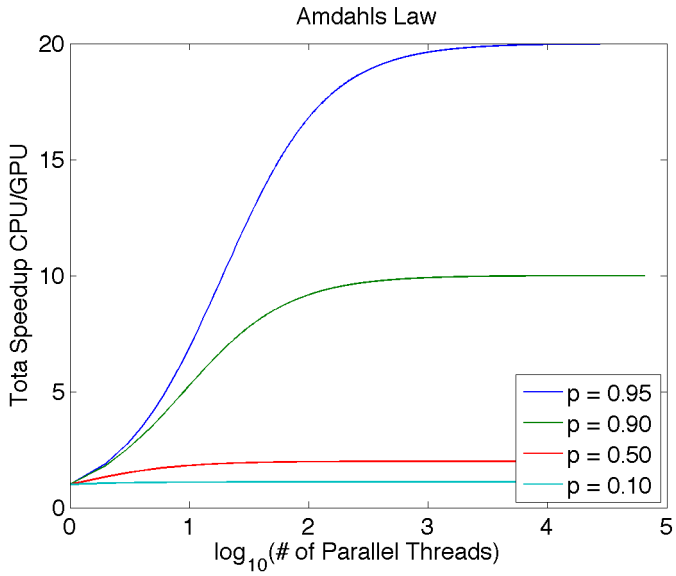


Figure 4.4: A semi-log plot of Amdahl's law, equation (4.1), for different values of the part of the code (measured in time) being parallelized. The x-axis is the 10-logarithm of the total number of parallel threads, and the y-axis is the total gained speedup. S is set to be equal the number of parallel threads. Amdahl's law gives the maximum theoretical speedup.

4.3.1 Parallelization

To fully use the power of the graphics card one needs to keep it busy as much as possible. One key concept is occupancy. First of all, one should have more blocks than cores. The number of blocks is determined by the grid size (which can be 2D)[3] There should be several blocks per core to assure that they always are busy, even when some blocks are waiting on thread synchronization. But there is a trade off of having many blocks residing on the same multiprocessor, and that is that the registers available per thread decreases. Thus higher occupancy does not necessarily lead to better performance as some of the variables may be placed in the local memory when there are too few registers available.

This leads to the next consideration, the block size. There exists a maximum number of threads per block. If the block size is greater than half this number only one block can reside on the multiprocessor. Unless this is necessary

[3]Having 2D grids and thread arrays does not in any way affect the performance.

for register reasons (see next subsection) this should be avoided, allowing for higher occupancy. The reason for wanting many threads is that despite their sequential execution of instructions, one can hide memory latencies by executing instructions on a second warp while the first one is paused or waiting for data. A heuristic for optimizing parallelization is the following.

- High priority. Have at least two blocks per multiprocessor.
- Medium priority. Occupancy, have enough threads to hide latency.
- Medium priority. The block size should be multiples of 32 to avoid almost empty warps.

4.3.2 Memory Usage

Optimizing memory usage is the most important factor in achieving high performance. As stated above one should try to minimize the calls made to the global memory. This is because it is off-chip and not cached. It takes four clock cycles to read a float from the registers of a thread (and the same for the shared memory) but it takes between 400 - 600 clock cycles to read it from the global memory [20]. Thus one is able to gain a speed-up of a factor of 100-150 by reusing the memory in the multiprocessors as much as possible. Furthermore, one should make sure that there are no bank conflicts reading the shared memory. If several requests are made to the same memory bank, the requests are serialized thus destroying the parallelization during a period of time. Lastly, if a kernel is using much shared memory it can be beneficial to store some of the data in the constant memory. A heuristic for memory usage optimization can be summarized as follows [24]

- High priority. Minimize data transfer between host and device and vice versa.
- High priority. Coalescing accesses to the global memory.
- Medium priority. Avoid bank conflicts accessing the shared memory.
- Low priority. Use constant memory if much shared memory is being used.

4.3.3 Instruction Throughput

One should always optimize ones code with regard to parallelization and memory usage first, before looking into optimizing the instruction throughput. Furthermore, instruction throughput optimization is something one is usually already trained to think about when writing code, for example avoiding divisions by square roots and instead use the reciprocal square root function. Because of the assumed familiarity of these ideas to the reader, it will be skipped here. For more information see [24].

- Medium priority. Use faster specialized functions instead of slower general ones.
- Medium priority. If precision is secondary to speed, use fast math libraries.

4.4 Results

The results of the parallel computing part of this thesis is presented with the reasoning behind this particular way of parallelizing the code. They could have been presented by simply plotting the speedup and attaching the code. This, however, would not have been very instructive.

4.4.1 Exposing the Data Parallelism

The equation being solved is the self-consistency equation for the order parameter, given here again for convenience

$$\Delta_{\alpha\beta}(\boldsymbol{p}_F, \boldsymbol{R}) = T \int \mathrm{d}^2\boldsymbol{p}'_F \lambda_{\alpha\beta\gamma\delta}(\boldsymbol{p}'_F, \boldsymbol{p}_F) \sum_{\varepsilon_n}^{\varepsilon_n \leq E_c} f_{\gamma\delta}(\boldsymbol{p}'_F, \boldsymbol{R}; \varepsilon_n). \quad (4.2)$$

In the general case, the object needed at each iteration is the anomalous Green's function summed over all angles and Matsubara frequencies. All of which are independent. Furthermore, f is calculated by solving the Ricatti equations along certain trajectories, which are all independent as well. A schematic picture of the layers of exploitable parallelism is shown in figure 4.5. At each iteration one starts with an order parameter Δ, then the entire matrix is rotated an angle set beforehand and interpolated. For each rotation and Matsubara frequency the Ricatti equations are solved along trajectories along the interpolated grid and then rotated back and interpolated once again to get the anomalous Green's function values at the points of the unrotated grid. As indicated by 4.5 all calculations corresponding to different, trajectories, frequencies and angles of rotation can be done in parallel, at least with an ideal computer.

4.4.2 Exploiting the Data Parallelism

Having identified the parallelism inherent in the problem of calculating the order parameter, it is necessary to analyze the part of the sequential code which one wants to parallelize to see if it is worthwhile and if so, how large the speedup will be.

When solving for a trajectory it has to be done in serial because the values are not independent. Thus if one parallelize all trajectories it will still at least take the time calculating a single trajectory. But this single trajectory must be calculated at some Matsubara frequency. So because all frequencies are independent, ideally the number of frequencies up to a certain point used in the parallel program does not affect the running time. At low temperatures, the number of Matsubara frequencies soar which means that the gains of parallelization at low temperatures can be very high indeed. By running the sequential program with the trajectory and Matsubara loops removed, and replacing them with a single trajectory calculation one can calculate the number p used in Amdahl's law. For a moderate system with 200 Matsubara frequencies and and a order parameter matrix size 215×215 one gets that $p \approx 0.99$. Letting $S \to \infty$ gives SU ≈ 100, i.e. an immense speedup!

Here it is appropriate to stop and think about the optimization of memory usage. Although parallelization has HIGH PRIORITY, so does minimizing data transfer between host and device as well as calls to the global memory. Each

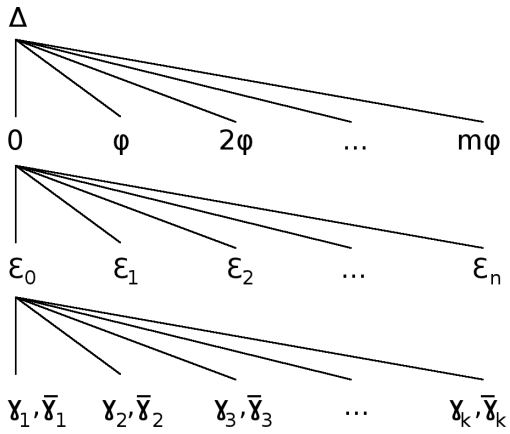


Figure 4.5: A schematic picture showing the different layers of exploitable parallelism. One starts each iteration of the self-consistency equation (4.2) for the order parameter with the last Δ and then rotating it several times depending on the angular resolution needed. For each rotation the Riccati equations (3.35) are solved along several trajectories giving γ and $\bar{\gamma}$. This is done for all Matsubara frequencies. Lastly the anomalous Green's function corresponding to the different rotations and Matsubara frequencies are rotated back, interpolated and summed, thus giving the order parameter used as input in the next iteration. In this thesis however are the rotations not executed in parallel.

block needs to read a certain part, I have chosen a single trajectory, of the order parameter from the global memory to have something to work with. By storing it on the shared memory it only needs to be read once. In each block the order parameter trajectory is used to calculate the corresponding elements of the anomalous Green's function for certain number of Matsubara frequencies. All of these are stored on the shared memory and then copied to the global memory, thus coalescing the global memory access. However, if the length of the trajectory is large and/or the number of Matsubara frequencies are vast, i.e. the temperature is low, the shared memory is too small and the calculations involving the same trajectory needs to be split to two or more blocks. These extra blocks need to read the same part of the global memory, because they are working with the same order parameter trajectory. This, at first glance, seems to pose a problem of global memory bank conflicts, serializing the code for some time, but this effect is fortunately negligible if the memory accesses are coalesced which they always should be. This can be seen in figure 4.6 where the anomalous Green's function is calculated for a very small order parameter, 7×7, for different numbers of Matsubara frequencies with increasing number of threads per block. The speedup quickly reaches the approximate maximum with only a few threads per block. This initial speedup is due to higher occupancy and latency hiding. However, just picking a number for the block size is not encouraged because there are rather great differences in performance for small changes in block size. A good heuristic though is to pick the largest number

possible, i.e. for this particular problem

$$\text{Block Size} \quad = \quad \frac{\text{Max Shared Memory}}{16\sqrt{2}k} - 1 \,, \tag{4.3}$$

where k is the side length of the order parameter. For maximum performance though, one should analyze how the speedup depends on the block size. But for very large order parameters this is not an option and simply choosing the largest block size possible is recommended.

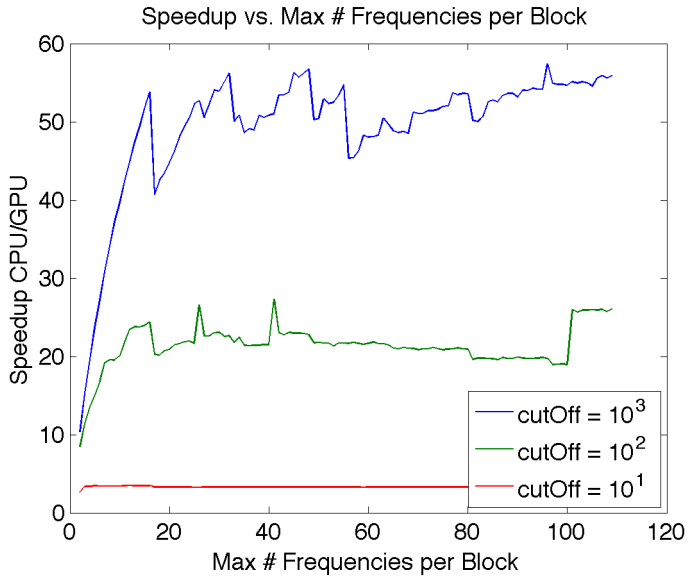


Figure 4.6: The plot shows how the choice of block size affects the speedup for different values of the cut off energy. The plot was made using $k_0 = 5$ and $T = 0.5$ giving the cut off frequency $n_c = \text{cutOff}/T$. The lines level out at certain block sizes because more calculations cannot be executed in parallel.

4.4.3 Speedup

The speedup gained running the parallelized code depends strongly upon the number of frequencies and the dimension of the order parameter. As seen in figure 4.7, the speedup is greater for smaller order parameters and for lower temperatures. This is a direct consequence of threads occupying more shared memory the larger the matrix, and that more calculations are done in parallel the lower the temperature. However, the speedup does not depend on the number of Matsubara frequencies when having a large order parameter matrix, in other words speedup is zero at the margin.

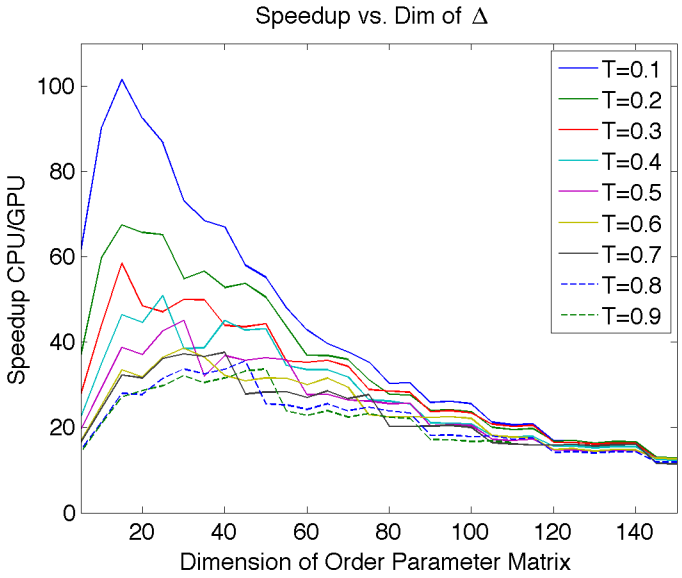


Figure 4.7: A measure of the speedup gained from using the parallelized code compared to the sequential one when solving the self-consistency equation for the order parameter. The speedup for the LDOS and current a little bit less but have the same essential behaviour. Ten iterations of the order parameter while-loop were carried out for each matrix size and temperature. The cut off frequency was set to $n_c = 10/T$.

Chapter 5

Conclusions and outlook

In this thesis I have replicated known results regarding vortices in s-wave type II superconductors using both serial and parallel code, executed on the CPU and GPU respectively. In doing this I have shown the power of using the GPU in numerical calculations of condensed matter physics systems. Although the speedup is large, there is definitely room for improvements. First of all, the calculations have been done by invoking the kernel from Matlab, which is not optimal. The obvious next step is to write the code entirely in C and push the optimization to its limits. In this thesis I chose to keep the calculations of the rotations of the order parameter on the CPU and therefore make more kernel calls. A large improvement of the program would be to move the rotation to the GPU as well. Assuming that the rotations are implemented efficiently on the GPU one can anticipate truly impressive speedups because there would only be one instance of copying from host to device.

Having done this transition to using C exclusively and moved the rotations to the GPU one can carry out very high resolution calculations of vortices and/or at very low temperatures, this would be especially interesting for unconventional pairing important in high T_c superconductors. Because the size of the shared memory of the particular graphics card being used effectively imposes an upper bound on the spatial resolution of the order parameter it is advised to drive the research towards the investigation of low temperature effects on the vortex cores. However, the maximum number of Matsubara frequencies and the spatial resolution are naturally not independent. Better resolution means longer trajectories (in terms of number of elements) being stored on the shared memory, this in turn means that fewer threads per block can be executed and thus decreasing the Matsubara frequency cutoff. One easy way to circumvent these problems is simply to buy more graphics cards. This, however, does not scale equally well as it would have to increase the shared memory size. This is so because when one has a very high resolution order parameter there exists only enough shared memory for one warp of threads per block. This means that there will be much copying from global to shared memory which would have been avoided if increasing the amount of shared memory was an option. Nevertheless, even if twice the amount of graphics cards generally does not mean twice the speedup, it does so when having saturated GPUs. In other words, when having saturated ones GPUs one can do the calculations at half the temperature in the same amount of time and with the same precision, simply by doubling the number of

available GPUs.

To conclude, parallel computing of vortex core structures in superconductors is indeed worthwhile, capable of achieving truly immense speedups. This speedup allows investigation of very low temperature behaviour not reachable before unless having access to a computer cluster. Hopefully, this will lead to further insights in unconventional pairings important in high T_c superconductors, and eventually room-temperature superconductors.

5.1 Acknowledgements

I would like to thank all the people at the Applied Quantum Physics Laboratory for letting me be a part of their research group and for having the best coffee in all of MC2. Furthermore, I would like to thank Mikael Fogelström and Tomas Löfwander for teaching me the quasi-classical method of superconductivity. Special thanks to Gustav Hjelmare for proofreading the thesis and to Ian Kellet for keeping me company at the office, watching South Park and eating pizza. Lastly, and most importantly, I would like to thank Daniel Midtvedt for putting up with me babbling on and on about parallelization, superconductors and other cool stuff, and for giving me a place to sleep when I really needed it.

Bibliography

[1] Michael Tinkham. *Introduction to Superconductivity.* Dover, 2nd edition, 2004.

[2] Pierre-Gilles de Gennes. *Superconductivity of Metals and Alloys.* Addison-Wesley, 2nd edition, 1989.

[3] Andrei Mourachkine. *Room-Temperature Superconductivity.* Cambridge International Science Publishing, 1st edition, 2004.

[4] NVIDIA. *NVIDIA CUDA C Programming Guide*, 2010. Version 3.2.

[5] Lokenath Debnath and Piotr Mikusinski. *Introduction to Hilbert Spaces with Applications.* Elsevier Academic Press, 3rd edition, 2005.

[6] Frederick W. Byron, Jr. and Robert W. Fuller. *Mathematics of Classical and Quantum Physics.* 1992.

[7] Eleftherios N. Economou. *Green's Functions in Quantum Physics.* Springer, 3rd edition, 2006.

[8] Stephen Weinberg. *The Quantum Theory of Fields.* Cambridge University Press, 2005.

[9] Jun J. Sakurai. *Modern Quantum Mechanics.* Prentice Hall, revised edition, 2005.

[10] Michael E. Peskin and Daniel V. Schroeder. *An Introduction to Quantum Field Theory.* Westview Press, 1995.

[11] Henrik Bruus and Karsten Flensberg. *Many-Body Quantum Theory in Condensed Matter Physics.* Oxford University Press, 1st edition, 2004.

[12] Gerald D. Mahan. *Many-Particle Physics.* Kluwer Academic/Plenum Publishers, 3rd edition, 2000.

[13] Nikolai Kopnin. *Theory of Nonequilibrium Superconductivity.* Oxford University Press, 1st edition, 2001.

[14] J. Rammer and H. Smith. Quantum field-theoretical methods in transport theory of metals. 2009.

[15] Matthias Eschrig. Distribution functions in nonequilibrium theory of superconductivity and andreev spectroscopy in unconventional superconductors. *Physical Review B*, 61(13):9061–9076, 2000.

[16] Matthias Eschrig, J. A. Sauls, and D. Rainer. Electromagnetic response of a vortex in layered superconductors. *Physical Review B*, 60(14):10447–10454, 1999.

[17] J. A. Sauls and M. Eschrig. Vortices in chiral, spin-triplet superconductors and superfluids. 2009.

[18] Lorenz Kramer and Werner Pesch. Core structure and low-energy spectrum of isolated vortex lines in clean superconductors at $t \ll t_c$. 1974.

[19] N Hayashi, Y. Kato, and M. Sigrist. Impurity effect on kramer-pesch core shrinkage in s-wave vortex and p-wave chiral vortex. 2004.

[20] Rob Farber. Cuda, supercomputing for the masses.

[21] Jason Sanders and Edward Kandrot. *CUDA by Example: An Introduction to General-Purpose GPU Programming*. Addison-Wesley, 1st edition, 2010.

[22] Tosaka. Cuda processing flow, 11 2008.

[23] Tim Mattson, Beverly Sanders, and Berna Massingill. *Patterns for Parallel Programming*. 1st edition, 2004.

[24] NVIDIA. *CUDA C Best Practices Guide*, 2010. Version 3.2.