



**FACULTY
OF MATHEMATICS
AND PHYSICS**
Charles University

BACHELOR THESIS

Róbert Jurčo

Simulations of quantum turbulence in thermal counterflow of superfluid helium using the vortex filament method

Department of Low Temperature Physics

Supervisor of the bachelor thesis: doc. RNDr. David Schmoranzer,
Ph.D.

Study programme: Physics

Study branch: Physics

Prague 2022

I declare that I carried out this bachelor thesis independently, and only with the cited sources, literature and other professional sources. It has not been used to obtain another or the same degree.

I understand that my work relates to the rights and obligations under the Act No. 121/2000 Sb., the Copyright Act, as amended, in particular the fact that the Charles University has the right to conclude a license agreement on the use of this work as a school work pursuant to Section 60 subsection 1 of the Copyright Act.

In date
Author's signature

I would like to dedicate my work to my parents, who have raised me to be the person I am today. You have always loved me unconditionally and your good examples have taught me to work hard for the things that I aspire to achieve. Thank you for all the guidance and support that you have given me, helping me to succeed and instilling in me the confidence that I am capable of doing anything I put in my mind. Thank you for everything.

Title: Simulations of quantum turbulence in thermal counterflow of superfluid helium using the vortex filament method

Author: Róbert Jurčo

Department: Department of Low Temperature Physics

Supervisor: doc. RNDr. David Schmoranzler, Ph.D., Department of Low Temperature Physics

Abstract: The aim of this bachelor thesis was to perform numerical simulations of quantum turbulence in superfluid helium in the so-called thermal counterflow. First, we verified an existing code developed by E. Varga with a few simple examples and with a counterflow in a rectangular channel, where we compared the simulation results with experimental data obtained from Superfluidity Laboratories and other numerical simulations. Here, we also improved the code by finishing the Barnes-Hut tree algorithm, adding a low-pass filter and partially parallelized it for use in larger data center clusters. Later, we adapted the code to less common counterflow geometries, e.g. spherically symmetric counterflow where the spatial temperature distribution can also play an important role and must therefore be considered as an integral part of the simulation.

Keywords: Superfluid helium, Quantum turbulence, Numerical simulations, Vortex Filament model

Contents

Introduction	2
1 Theory	3
1.1 Quantum turbulence in superfluid Helium	3
1.1.1 Helium hydrodynamics	3
1.1.2 Vortex filament model	5
1.1.3 Vortex dynamics	6
1.2 Implementation	7
1.2.1 Discretization	7
1.2.2 Time evolution	7
1.2.3 Boundaries and rectangular geometry	7
1.2.4 Reconnections	9
1.2.5 Barnes-Hut algorithm	9
1.3 Improvements of the model	10
1.3.1 Smoothing of vortices	11
1.3.2 Spherical geometry	12
1.3.3 Temperature gradient	13
2 Results and discussion	15
2.1 Initial tests	15
2.1.1 Single vortex ring	15
2.1.2 Vortex reconnection	15
2.1.3 Two lines	17
2.2 Rectangular geometry	18
2.2.1 Vortex density	18
2.2.2 Polarization	19
2.3 Spherical geometry	23
2.3.1 Vortex density	23
2.3.2 Polarization	24
2.3.3 Tests of periodic boundaries in spherical geometry	24
Conclusion	28
Bibliography	29
List of Figures	32
List of Tables	34
Author's Contribution	35
Appendix	36

Introduction

Turbulence in fluids is a motion that is characterized by chaotic changes in fluid pressure and flow rate. It is the opposite of laminar flow, in which the fluid flows in parallel layers, with no mixing between these layers. Turbulence commonly occurs in most of the natural and man-made flows and has consequences such as enhanced surface heating or loss of lift. We most often encounter turbulence when traveling by plane, where turbulence is a major cause of non-fatal aircraft accidents and incidents [1]. Another example where turbulence plays an important role is the surface heating of a space shuttle as it re-enters the Earth's atmosphere. If the shuttle passes through a turbulent layer, the heating will increase upto 3 to 8 times [2], therefore assuming a laminar boundary layer would be catastrophic.

The reason why the turbulence remains an unsolved problem is related to the Richardson cascade [3]. It describes the distribution of energy in turbulence. Energy enters turbulence on the largest scales in the form of kinetic energy of largest motions of turbulence. In contrast, the dissipation of energy takes place on the smallest (Kolmogorov) scales thanks to the viscosity [4]. Ideally you want to resolve all the scales separately, either numerically or experimentally. Turbulence can be fully described by complete Navier-stokes equations, which, however, we cannot yet solve analytically. Direct numerical simulation should be carried out with big enough domain sizes to represent large scale motions, while discretization must be fine enough to capture smallest scales.

In this work, we focus purely on the turbulence in the superfluid part of He-4. Helium has two stable isotopes which differs in the number of neutrons. He-4 has two neutrons while he-3 has one. From an abundance point of view he-4 is the most common one. As shown in Figure 1.1, He-4 has two liquid phases He-I and He-II. The main difference from conventional fluids is the description of the He-II using two fluid model, which is based on the idea of He-II having normal and superfluid component, with each component having its own velocity and density [5]. Quantum turbulence is then solely located in the superfluid component and quantized vortices can be understood as one-dimensional (linear) disturbances in superfluid helium. Each vortex has a fixed value of circulation, corresponding to its quantum $\kappa = 9.97 \times 10^{-8} \text{m}^2/\text{s}$.

1. Theory

1.1 Quantum turbulence in superfluid Helium

The first man who managed to liquefy helium was Kamerlingh Onnes [6] in 1908. To achieve that, he needed to reach the normal boiling point of Helium, which is 4.2 K at atmospheric pressure. As we can see in the helium phase diagram (Figure 1.1), liquid helium remains in the liquid phase under its own vapor pressure down to the absolute zero temperature. To reach a solid phase up to 25 atmospheres of pressure are required.

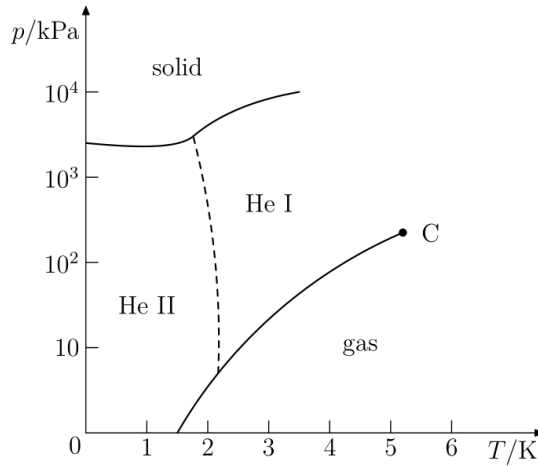


Figure 1.1: Phase diagram of He-4.

At a temperature $T_\lambda = 2.172$ K, called the lambda temperature, liquid helium undergoes a second order phase transition. The lambda transition is characterized by no specific volume change and no latent heat involved, however the heat capacity curve diverges at this temperature. In 1927 Keesom and Wolfke named the liquid phase above lambda point He-I and the liquid phase below lambda point He-II, in order to distinguish the two phases of the liquid helium. He-I can be described as a classical Newtonian fluid by Navier-Stokes equation. On the other hand various quantum phenomena take place in He-II. In 1938 Kapitza [7], Allen and Misener [8] measured zero resistance of the flow of He-II through small capillaries with diameter of the order of 10^{-4} cm. In the same year Keesom and MacWood [9] proved the existence of a viscous drag in He II using the oscillating disks. The He II property of being both viscous and non-viscous at the same time led Tisza [10] and Landau [11] to the formulation of the two fluid model.

1.1.1 Helium hydrodynamics

The two fluid model formulated by Tisza (1940) and Landau (1941) is based on the idea that He-II has two components, normal and superfluid, with each component having its own velocity and density [5]. The sum of the normal fluid density ρ_n and the superfluid density ρ_s is equal to the bulk fluid density ρ

$$\rho = \rho_n + \rho_s. \quad (1.1)$$

In Figure 1.2 we can see the temperature dependence of the normal and superfluid component densities. The superfluid component is not present at T_λ , while at the temperature of absolute zero He II consists of superfluid only. Superfluid component carries no entropy, no heat and has no viscosity.

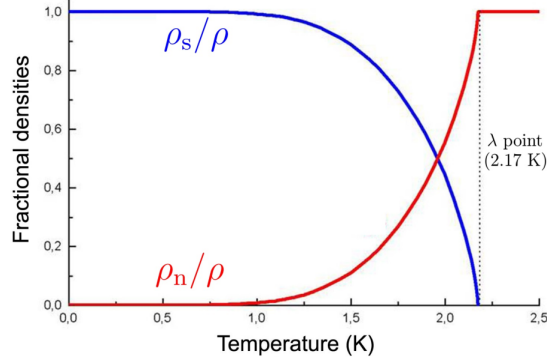


Figure 1.2: Temperature dependence of densities of the normal (red) and superfluid (blue) component densities [12].

In 1938, London came up with the idea of explaining the superfluidity of He-II using a weakly interacting Bose gas [13]. He described the superfluid component of He-II with macroscopic condensate wave function, which in stationary condition is given by

$$\Psi(\vec{r}, t) = \Psi_0 e^{i\phi(r)}, \quad (1.2)$$

where Ψ_0 is an amplitude and $\phi(r)$ a macroscopic phase. Using

$$\hat{p} \Psi = \vec{p} \Psi \quad (1.3)$$

we see that the superfluid velocity is given by the gradient of the macroscopic phase as

$$\vec{v}_s = \frac{\hbar}{m_{He4}} \nabla \phi, \quad (1.4)$$

where m_{He4} is the mass of the He-4 molecules. If we apply the rotation operator to eq. 1.4. ($\text{rot} \nabla \phi = 0$), we find that the superfluid component is irrotational, i.e.

$$\text{rot} \vec{v}_s = 0. \quad (1.5)$$

Even though the superfluid is irrotational, there exists vortex structures in it. However, this requires the macroscopic condensate wave function inside the vortex core to be zero. Therefore we consider quantum vortices to be one-dimensional (linear) defects in superfluid helium [14]. Integrating Equation 1.5 immediately leads to the quantisation of circulation Γ around a closed loop L

$$\Gamma = \oint_L \vec{v}_s \cdot d\vec{l} = \frac{\hbar}{m_{He4}} \oint_L \nabla \Phi \cdot d\vec{l} = n \frac{2\pi\hbar}{m_{He4}} = n\kappa, \quad (1.6)$$

where we use the fact that $\Psi(\vec{r}, t)$ is well defined only if the phase Φ changes by $2\pi n$ after going once around the closed-loop. This way defined $\kappa = 2\pi\hbar/m_{He4} = 9.97 \times 10^{-8} \text{m}^2/\text{s}$ is called a quantum of circulation [14]. In 1955 Feynman showed [15] that these superfluid vortex lines appear when He-II moves faster than a

certain critical velocity, what leads to formation of quantum turbulence. Quantum turbulence also occurs in He-3, however in He-4 quantum vortices acquire only the lowest quantum of circulation. This is the main reason why numerical simulations of quantum turbulence are simpler for the case of He-4.

The generalization of Landau's two-fluid model is called Hall-Vinen-Bekarevich-Khalatnikov (HVBK) model and includes the presence of vortices inside the flow. According to HVBK model [16] the incompressible equations for motion of normal component $\vec{v}_n(\vec{r}, t)$ and superfluid component $\vec{v}_s(\vec{r}, t)$ can be written as

$$\frac{\partial \vec{v}_n}{\partial t} + (\vec{v}_n \cdot \nabla) \vec{v}_n = -\frac{1}{\rho} \nabla P - \frac{\rho_s}{\rho_n} S \nabla T + \frac{1}{\rho_n} \eta \Delta \vec{v}_n + \frac{1}{\rho_n} \vec{F}_{ns} \quad (1.7)$$

$$\frac{\partial \vec{v}_s}{\partial t} + (\vec{v}_s \cdot \nabla) \vec{v}_s = -\frac{1}{\rho} \nabla P + S \nabla T - \frac{1}{\rho_s} \vec{F}_{ns} \quad (1.8)$$

where ∇P and ∇T are the pressure and temperature gradients, S is the specific entropy of the He-II and η is the viscosity of the normal component. The mutual friction term is given by [16]

$$\vec{F}_{ns} = \alpha \kappa \rho_s L (\vec{v}_n - \vec{v}_s) \quad (1.9)$$

where α is a mutual friction coefficient and

$$L = \gamma^2 (\vec{v}_n - \vec{v}_s)^2 \quad (1.10)$$

is the vortex line density. Equation 1.10 is called Vinen's equation and the empirical parameter γ is a function of temperature T that is determined from experiment.

1.1.2 Vortex filament model

At the end of the 1980s, K.W. Schwarz[17] developed the Vortex Filament Model to perform numerical simulations of quantized vortex dynamics. The motion of a superfluid component of the He II is in this model given by the Biot-Savart law [17]

$$\vec{v}_s(\vec{r}, t) = \frac{\kappa}{4\pi} \int_{\Gamma} \frac{(\vec{s} - \vec{r}) \times d\vec{s}}{|\vec{s} - \vec{r}|^3}, \quad (1.11)$$

where Γ represents integration over all the vortices located in the superfluid component of He II. The numerical calculation is implemented in the simulation by representing the vortex core (vortex) by a series of points. These points then create so-called vortex tangle. For each point of the vortex tangle we sum up the velocity induced from the segments formed by all the other points. This way we obtain self-induced velocity from the tangle onto the point of the vortex, however we may notice that integral in Equation 1.11 diverges in the vicinity of the point of interest. Therefore, the so-called LIA approximation (Local Induction Approximation) was introduced. This approximation is basically a Taylor's approximation of Equation 1.11 around the point of interest. We can then write this approximation in the form

$$\vec{v}_{LIA}(\vec{r}, t) = \frac{\kappa}{4\pi} \vec{s}' \times \vec{s}'' \ln \frac{2}{a} \sqrt{\frac{l_+ l_-}{e}}, \quad (1.12)$$

where a is a width of the vortex core, \vec{s}' , \vec{s}'' are tangent and normal vectors of the vortex, and l_+ , l_- are distances to the next and previous point of the discrete vortex tangle respectively. The self-induced velocity from all the segments except the two in the vicinity of the investigated point is then given by the sum through all the points of the vortex tangle [18]

$$\vec{v}_{BIOT}(\vec{r}, t) = \frac{\kappa}{4\pi} \sum_{j \neq i, i+1} \frac{(R_j + R_{j+1}) (\vec{R}_j \times \vec{R}_{j+1})}{R_j R_{j+1} (R_j R_{j+1} + \vec{R}_j \cdot \vec{R}_{j+1})} \quad (1.13)$$

where $\vec{R}_j = \vec{s}_j - \vec{s}_i$ and $R_j = |\vec{R}_j|$. By summing the above equations, we obtain the resulting relationship for the self-induced vortex velocity

$$\vec{v}_s(\vec{r}, t) = \vec{v}_{LIA}(\vec{r}, t) + \vec{v}_{BIOT}(\vec{r}, t). \quad (1.14)$$

This velocity also corresponds to the velocity of the fluid at point \vec{r} , i.e. in the case where the normal component has zero velocity, the vortices are carried by the superfluid component. Finally, we need to know tangent and normal vectors at the individual points. These we get by approximating the spatial derivatives \vec{s}' and \vec{s}'' by a fourth-order finite difference method. The advantage of using this method is its functionality for non-constant distances between individual points.

1.1.3 Vortex dynamics

In our model, we do not calculate turbulence in the normal component, so we must prescribe it. For this purpose, we have introduced the possibility to define a fixed so-called external velocity for both components. The mutual friction term \vec{F}_{ns} in the Equations 1.7 and 1.8 represents the friction of the two helium components. This friction results in a drag force, which describes the interaction of vortices with the normal helium component. In 1957 Vinen and Hall [19] discovered that the effect of mutual friction can be described by an equation

$$\Delta \vec{v}_s = \alpha \vec{s}' \times (\vec{v}_n - \vec{v}_s) - \alpha' \vec{s}' \times (\vec{s}' \times (\vec{v}_n - \vec{v}_s)), \quad (1.15)$$

where α and α' are the coefficients of the mutual friction. Coefficients of the mutual friction are functions of temperature and reach zero at 0 K. Therefore to allow calculations at finite temperature the Equation 1.15 needs to be incorporated into the Schwarz model. The difference $\vec{v}_n - \vec{v}_s$ is called counterflow velocity \vec{u}_{ns} and determines the mutual friction and vortex line density given by Equations 1.9 and 1.10. The first term of Equation 1.15 represents the dissipation of energy, while the second term represents the conservative force between vortices. The dissipation of energy is manifested by a reduction in the distance between vortices and by their extinction. However, energy dissipation does not occur at zero temperature and therefore we do not include the effect of internal friction described by Equation 1.15.

1.2 Implementation

1.2.1 Discretization

As already mentioned, in VFM the vortex is represented using a series of discrete points. These points then form a so-called vortex tangle. During evolution, we must be careful not to change by several orders of magnitude the distance between the individual points. We ensure this by defining a minimum and a maximum discretization distance. In our simulations we use $dl_{min} = 0.5 \times 10^{-3}$ cm and $dl_{max} = 1.5 \times 10^{-3}$ cm. If the distance between adjacent points decreases below dl_{min} and the removal of the given point does not create a segment whose length would be greater than dl_{max} , then we remove the given point. If the distance between adjacent points increases above dl_{max} , we fit a vortex line between them with a circle with the curvature given by $1/|\vec{s}''|$ and we insert a new point in the middle.

1.2.2 Time evolution

Once we have the initial state of turbulence ready, we can begin to calculate the time evolution of the individual points of the vortex tangle. For this we implemented the fourth-order Runge-Kutta (RK4) method. If we consider the Schwarz's equation $d\vec{s}_i/dt = \vec{v}_i$, the evolution of \vec{s}_i is then given by

$$\vec{s}_i(t + dt) = \vec{s}_i(t) + \frac{1}{6} \left(\vec{v}_i^{(1)} + 2\vec{v}_i^{(2)} + 2\vec{v}_i^{(3)} + \vec{v}_i^{(4)} \right) dt \quad (1.16)$$

where dt is the time step and $\vec{v}_i^{(\alpha)}$ are the velocities obtained for partial steps:

$$\begin{aligned} \vec{v}_i^{(1)} &= \vec{v}_i(\vec{s}_i, t), \\ \vec{v}_i^{(2)} &= \vec{v}_i\left(\vec{s}_i + \vec{v}_i^{(1)} dt/2, t + dt/2\right), \\ \vec{v}_i^{(3)} &= \vec{v}_i\left(\vec{s}_i + \vec{v}_i^{(2)} dt/2, t + dt/2\right), \\ \vec{v}_i^{(4)} &= \vec{v}_i\left(\vec{s}_i + \vec{v}_i^{(3)} dt, t + dt\right). \end{aligned} \quad (1.17)$$

The value of the time step we use in our simulations is $dt = 10^{-4}$ s.

1.2.3 Boundaries and rectangular geometry

Quantum turbulence is commonly calculated in the rectangular geometry. To apply this geometry, we need to implement algorithms for different types of boundary conditions. The boundary conditions of the studied volume characterize the behavior of vortices near the boundaries and the vortices passing through them. In our simulations we recognize 3 different types of boundaries: open boundary, periodic boundary and solid wall. Boundary conditions consist of two main algorithms: first algorithm handles the behavior of the vortices in contact with the boundary and the second algorithm handles so-called image tangles which complement the effect on the induced velocity from the specific boundary. Together there may be up to 26 image tangles placed around the investigated volume: 6 of them are sharing a wall with the volume, other 12 are sharing an edge and the

last 8 are touching the investigated volume in its corners. Figure 1.4 shows the placement of image tangles around the examined volume in 2 dimensions.

Open boundaries: The basic option for the boundary is the open boundary, i.e the vortices can freely move through the boundary and after reaching large enough distance (few centimeters) from the observed volume are removed. In this case, there is no need to use image tangles.

Periodic boundaries: To simulate a small element inside a much larger flow, we can use a trick by placing copies (image tangles) of the investigated volume around itself. In this case a vortex passing through the boundary returns to the volume through the opposite boundary. The scheme of placing image tangles behind the boundaries is shown in Figure 1.4. We have to always calculate the contribution to the induced velocity from these image tangles on the vortices inside the investigated volume. Note that if one of the boundaries is periodic the opposite one has to be also periodic, however the other 4 boundaries could be open or solid walls.

Solid walls: And finally the solid wall boundary is implemented by removing every point that passes the wall and then attaching the points that are in close proximity to the wall to the wall. The point that is attached to the wall must no longer leave the wall. We can see a tangle with walls in Figure 1.3. In order to calculate how the solid walls affect the velocity within the volume we need to implement image tangles, see the paragraph below.

Pinning modes: To determine if the points are able to move on the walls or have to stay on the same place, we assign a pinning mode to those points. Pinning mode can be either PINNED (the point is pinned on the wall hence will never move again) or PINNED_SLIP (the point can freely move on the wall, what is achieved by subtracting the normal component of the velocity).

Image tangle in the case of a solid wall: In the case of a solid wall, we must ensure that the component of helium velocity perpendicular to the wall is zero at the wall location. We will achieve this by introducing a mirror image tangle, i.e. a mirrored copy of our vortex tangle that is located behind the wall, however this time with all the vortices inside this image tangle having the opposite direction. This will result in zero helium flow through the wall. A diagram explaining this idea is shown in Figure 1.4. This principle is similar to the method of image charges used in electrostatics, where the so-called image charges are placed outside the examined volume in order to satisfy the boundary condition.

Typical configurations: To simulate a counterflow we used 3 different compositions of the boundaries: periodic everywhere, two parallel planes and a channel. Simulation in the everywhere periodic flow is achieved by setting all the 6 boundaries to be periodic, i.e all the 26 image tangles are precise copies of the investigated volume. In the case of the two parallel planes we need to also mirror the image tangles of the two mirror tangles to achieve periodization of the wall.

Same principle is applied in the case of the channel, however this time there are 12 image tangles double mirrored because of the 4 perpendicular solid walls and their periodic images.

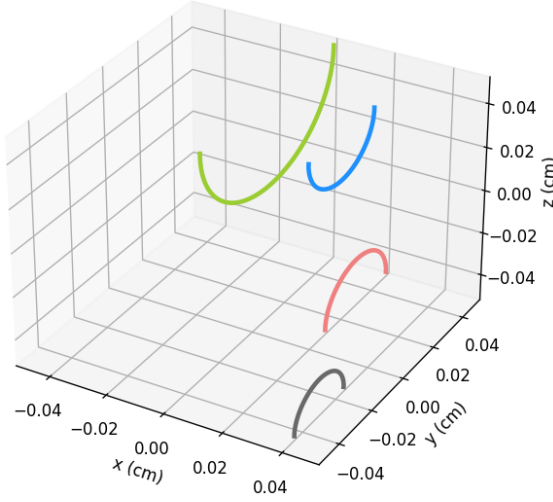


Figure 1.3: Vortex tangle simulated in counterflow between two parallel planes.

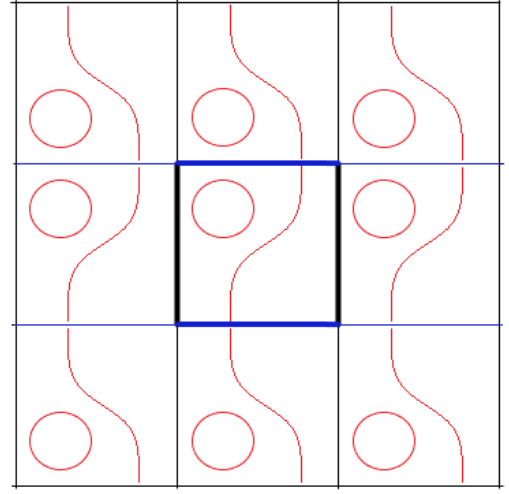


Figure 1.4: Image tangles scheme for simulation of counterflow between two parallel walls. Bold lines represents simulated volume, eight squares around the central one represents the image tangles, blue lines represents solid parallel walls and red lines represents vortex lines. We can see that solid walls acts as mirrors.

1.2.4 Reconnections

When two vortex filaments come too close to each other or even collide, they are reconnected (see figure 1.5). This phenomenon has been reported experimentally by many researchers and is also confirmed by detailed simulations of the Gross-Pitaevskii equation. However it is not possible to simulate this effect by any of above mentioned algorithms and equations. Therefore we must introduce an additional algorithm. When two vortex segments approach each other at a distance less than 4×10^{-4} cm and an angle greater than 5° we interconnect the points of both segments. It is important to ensure that the reconnection distance is smaller than the minimal discretization distance dl_{min} .

1.2.5 Barnes-Hut algorithm

The biggest disadvantage of the induced velocity calculation using Equation 1.13 is its numerical time complexity $\mathcal{O}(n^2)$. Therefore, we have implemented the Barnes-Hut tree algorithm, whose time complexity is $\mathcal{O}(n \ln n)$ [20]. This allowed us to drastically increase the number of points in the simulations. In contrast to the original number of points, usually around 1000, we are now able to run simulations with more than 100 000 points. The Barnes-Hut tree algorithm consists

of the recursive arrangement of all the points into a tree structure by dividing the volume always into 8 smaller parts called child-boxes (see Figure 1.6). For each child-box we then calculate centre of the vorticity \vec{r}_{VC} as an average over the positions of all the points contained in the given child-box and its total vorticity $\vec{\omega}_{VC}$ as a sum of all the segments contained in the given child-box (one segment is made up of two adjacent points of the vortex). Then every child-box can be approximated by one segment $\{\vec{r}_{VC}, \vec{r}_{VC} + \vec{\omega}_{VC}\}$ contained in the centre of the child-box vorticity \vec{r}_{VC} . If we want to determine the effect on a given vortex point from all the points at a distance l , the tree algorithm replaces points at a distance l with child-box which edge is d , where $d < \theta l$ where θ is so-called opening angle, which indicates the accuracy of the approximation. In our simulations we use the same opening angle $\theta = 0.4$ as Baggaley and Barenghi in [21].

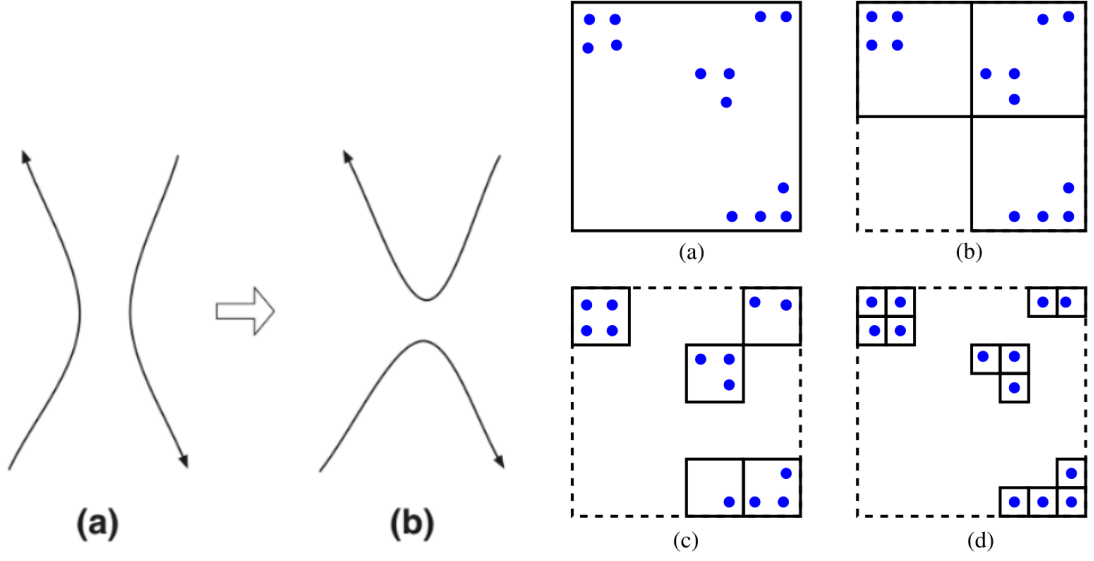


Figure 1.5: Scheme of vortex segments reconnection. (a) Two vortex lines before reconnection. (b) Vortex lines after reconnection [22].

Figure 1.6: Scheme of tree structure construction in 2D. The points (blue) are located inside the investigated volume (a), which is divided into four child-boxes of half size (b), until (d) there is only one point per child-box [23].

1.3 Improvements of the model

The code we used to simulate turbulence was developed in 2019 by Emil Varga [24]. Over the last two and a half years, we have improved the overall reliability of the code, added documentation, and completed the implementation of the Barnes-Hut algorithm. Subsequently, we implemented a smoothing algorithm that significantly improved the numerical stability and usability of the code, and we also developed a version of the code that can be applied to spherical geometry. An overview of the final version of the code can be seen in Appendix 1.

1.3.1 Smoothing of vortices

During the initial tests we encountered the problem of numerical instability of the simulation in the form of high spatial frequencies that scatter the vortex points. We have not been able to suppress this instability either with a smaller time step or with a smaller discretization of vortices. Therefore, we introduced the smoothing of vortices in every step of the simulation. We chose two methods, from which we subsequently chose the one that was more suitable for us.

Low pass filter

We made the first smoothing attempts using the Fourier transform of the spatial vortex coordinates as a function of the parameter enumerating the points. Subsequently, we removed 80% of the spatial frequencies, i.e. all the frequencies with wavelength smaller than 5 discrete points due to the calculation of tangents and normals from the position of 5 points. We obtained new positions of points by the inverse Fourier transform, while we did not change the positions of the points attached to the walls. In the simulation we used the Fast Fourier Transform (FFT) algorithm from the library FFTW 3.3.9 [25]. A clear advantage of this algorithm is that it is idempotent, i.e. the repeated application does not change the result of the filter. Therefore, we decided to use the low-pass filter in all simulations.

Savitzsky-Golay filter

The second smoothing method we tested was the Savitzsky-Golay filter. This method is based on fitting the near vicinity of the vortex point with a low-degree polynomial. Savitzky and Golay have shown [26] that a moving polynomial fit can be treated mathematically in the same way as a moving weighted average. Therefore, the Savitzsky-Golay filter was very easy to implement, in the form of a 3rd degree polynomial fit through 7 points. The relationship used in the simulation is [26]

$$Y_n = \frac{1}{21} (-2y_{n-3} + 3y_{n-2} + 6y_{n-1} + 7y_n + 6y_{n+1} + 3y_{n+2} - 2y_{n+3}). \quad (1.18)$$

In the case of an open vortex, we did not use the Savitzsky-Golay filter on the first 3 points at the wall (one fixed on the wall and his 2 neighbour points). We also implemented the Savitzsky-Golay filter fitting a polynomial of the 3rd degree just through 5 points, but it was less effective in keeping the simulation numerically stable. This method however has one downside, it is not idempotent. Therefore the repeated use of this algorithm reduces the radii of curvature of the vortices, and hence decreases the vortex density. We can see this problem directly from Equation 1.18: with repeated application, the vortex becomes more and more smoother.

1.3.2 Spherical geometry

Spherical counterflow is the flow in a spherical cell driven by a heater in the middle of the cell. According to [27], the equilibrium state of numerical simulation of the spherical counterflow can be achieved only if the temperature gradient is included in the model. Due to the high numerical complexity of the spherical simulations, it is worth to simulate only a spherical section. We chose a spherical section with a square base and angles 20° , as it is shown in Figure 1.7. To implement spherical geometry, the following parts of the simulation had to be modified: boundary conditions, i.e. periodicity, solid walls, image vortices behind the solid wall and calculation of the counterflow velocity.

Image vortices

To model a solid wall, we need to make sure that the vortex velocity has zero normal component on the wall. In rectangular geometry, this was achieved using image vortices, i.e. by mirroring the vortices over a solid wall and reversing their direction. However, in spherical geometry, this problem is more complicated. The right way to do this is to take the closed volume, calculate the velocity field at the boundary from the vortices inside, then take the opposite velocities and solve the Laplace equation to get a flow that zeroes the normal component of the velocities at the boundary [28]. The disadvantage is the requirement of a closed volume, because then we cannot use periodic boundaries. In order to obtain zeroth approximation of the velocity field which we must add, we can use the image vortices behind the spherical wall. We obtain them using the spherical inversion, known from electrostatics, given by the relation [28]

$$\vec{r}' = \frac{R^2}{|\vec{r}|^2} \vec{r} \quad (1.19)$$

where R is the radius of the spherical wall through which we mirror the vortex. The use of image vortices simplifies the calculation of the Laplace equation and it is a computationally inexpensive method working in open and periodic geometries, with limited computational errors.

Periodicity

As already mentioned, due to the high numerical complexity of the spherical simulations, we only simulated a spherical section with a square base and angles of 20° , as shown in Figure 1.7. However, there is a problem of how to periodically move vortex points that cross the periodic boundary. If we convert the Cartesian coordinates of a point to spherical, then rotate it 20° back to the investigated volume, and then we convert the coordinates back to Cartesian we create a small deviation from the position. This error is caused by rotation from one parallel of latitude to another, causing the points on the parallel to change distances between each other. The second way to rotate the points is to use rotation matrices, which however do not commute. This causes the periodic shifts of the points that have passed the corner of the square base to depend on the order of rotations. To verify that this error is insignificant, it is necessary to compare the results of the simulation in the spherical section with the simulation in the whole sphere.

Counterflow velocity

The heat in Helium II is carried only by the entropy s of normal fluid, hence in case of spherical geometry with a steady heat source \dot{Q} at any radius r following holds

$$\dot{Q} = 4\pi r^2 \rho s T u_n, \quad (1.20)$$

where ρ is the total density, T is the temperature and u_n is the velocity of the normal component of the Helium. Quantities ρ and s are functions of the temperature T , while u_n depends also on the radial distance from the heater r . Therefore we calculate velocities of the Helium components using

$$u_n(r, T) = \frac{\dot{Q}}{4\pi T \rho(T) s(T)} \frac{1}{r^2}, \quad (1.21)$$

$$u_s(r, T) = -\frac{\dot{Q}}{4\pi T \rho(T) s(T)} \frac{\rho_n(T)}{\rho_s(T)} \frac{1}{r^2}, \quad (1.22)$$

1.3.3 Temperature gradient

The quantities ρ , ρ_s , ρ_n , s , α , α' are all functions of the temperature, and we obtained the given dependencies from [29]. According to [30], the thermal gradient in 1.7 is driven by the term that contains \vec{F}_{ns} , i.e

$$\nabla T \approx \frac{1}{s\rho_s} \vec{F}_{ns}. \quad (1.23)$$

Now we substitute \vec{F}_{ns} from equations 1.9, 1.10 and the counterflow velocity $u_{ns} = u_n - u_s$ from Equations 1.21, 1.22 by what we get

$$\frac{dT}{dr} = -\frac{\alpha\kappa\gamma^2\rho^3}{s\rho_s^3} \left(\frac{\dot{Q}}{4\pi T \rho s} \right)^3 \frac{1}{r^6}, \quad (1.24)$$

where all quantities are evaluated at the reservoir temperature, called bulk temperature, $T_\infty = 1.65$ K. During integration, the bulk temperature is understood as the temperature at radial infinity. Our final result is then

$$T(r) = T_\infty + \frac{\alpha\kappa\gamma^2\dot{Q}^3}{320\pi^3 T_\infty \rho_s^3 s^4} \frac{1}{r^5}, \quad (1.25)$$

where α , κ , γ , s and ρ_s are all evaluated at T_∞ . The above mentioned γ is an empirical parameter, relating the vortex line density L with the counterflow velocity $u_{ns} = u_n - u_s$ (Equation 1.10: $L \approx \gamma^2 (u_n - u_s)^2$). The value of γ is determined experimentally, as shown in Figure 1.8. In our simulations we are using the higher fit $\gamma(T) = 155.8T - 75$.

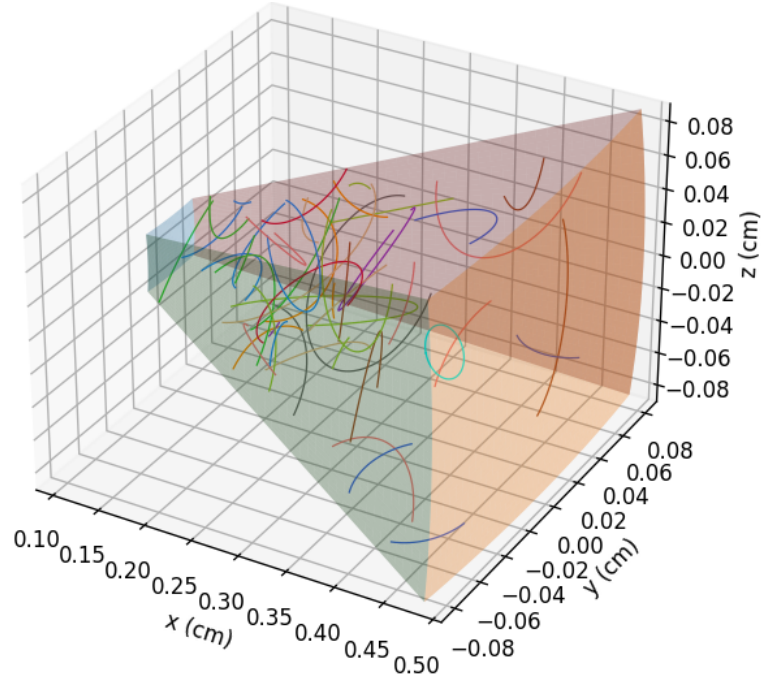


Figure 1.7: Vortex tangle simulated in spherical section with square base.

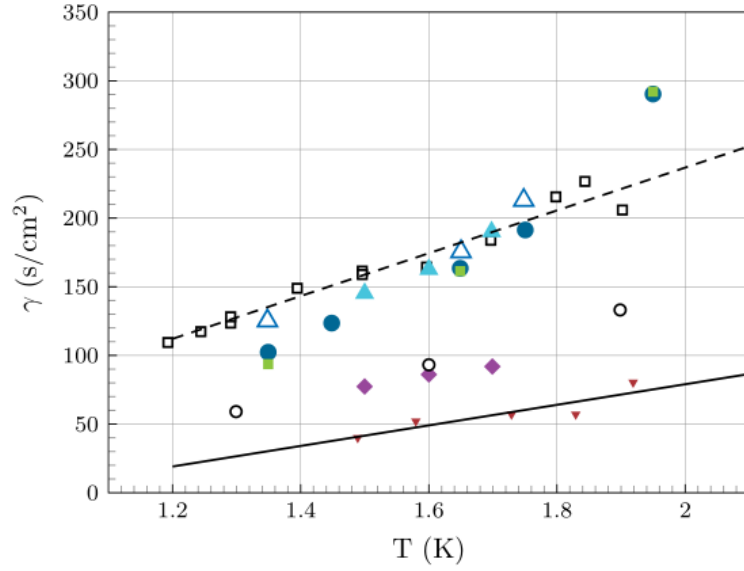


Figure 1.8: The parameter γ : The dashed line is a linear fit $\gamma(T) = 155.8T - 75$ and the solid line is a linear fit $\gamma(T) = 74.9T - 70.7$ [31]. Presented data: (i) solid blue circles, solid green squares, open blue up-triangle, Babuin et al. [31]; (ii) open squares, Ashton et al. [32]; (iii) solid red down-triangle, Chagovets and Skrbek [33]; (iv) solid blue up-triangle, solid magenta diamond, Martin and Tough [34]; (v) open circles, Childers and Tough [35].

2. Results and discussion

2.1 Initial tests

2.1.1 Single vortex ring

The first test of our simulation was the movement of single ring vortex, while evaluating its velocity and radius. Vortex ring is the easiest possible simulation, because there are no reconnections and the superfluid velocity of the ring can be analytically determined from its radius as [36]

$$v_s^{(loop)}(R) = \frac{\kappa}{4\pi R} \left[\ln \left(\frac{8R}{a_0} \right) - \frac{1}{2} \right]. \quad (2.1)$$

The simulations were done in periodic boundary conditions at a temperature of 1.3 K, with a time step $dt = 10^{-4}$ s and discretization distances of points in the interval $dl_{min} = 5 \times 10^{-4}$ cm to $dl_{max} = 2 \times 10^{-3}$ cm. Here we first encountered a problem with numerical instability of simulations. Therefore we implemented the low pass filter and in Figure 2.1 we can see a comparison of the simulation without and with the filter. Note the high scatter in the unfiltered data. In this case, distortions were observed on the vortex ring. We have come to the conclusion that we will always use a low pass filter to guarantee the numerical stability. In the case of the simulation where we used the low-pass filter, the difference between the simulation and the theory was less than 5%.

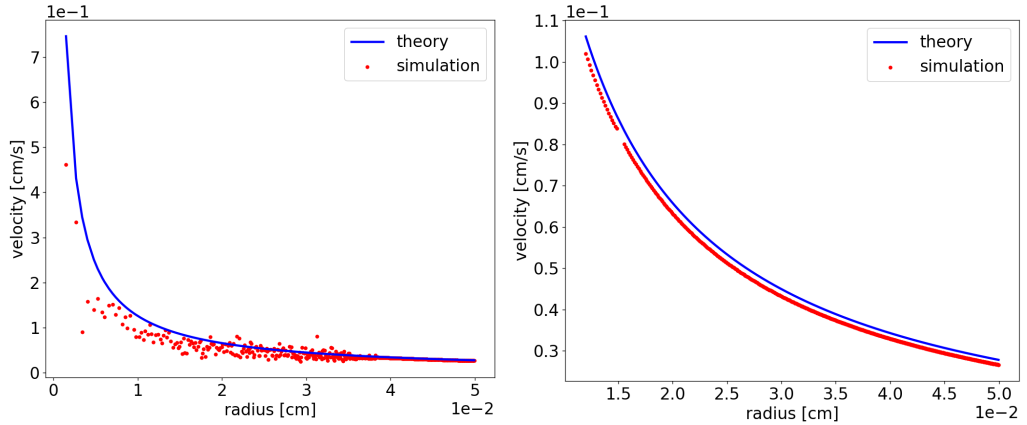


Figure 2.1: Simulation of one vortex ring: without smoothing (left), with low pass filter (right).

2.1.2 Vortex reconnection

The easiest way how to test the reconnection algorithm was a simulation of two rings colliding with each other. We can see in Figure 2.2, that the collision induces Kelvin waves on the vortices as expected. The waves gradually decay due to the mutual friction force. From the data we obtained from the simulation, we then evaluated the distance of the reconnected points as a function of time. According to [37], distance of the reconnected points is given by

$$l(t) = \beta \sqrt{\kappa (t - t_0)}, \quad (2.2)$$

where t_0 is time of the collision and the parameter β is according to [37] approximately 2.3. In Figure 2.3, we can see distance of the reconnected points obtained from the simulation together with Equation 2.2, however for $\beta \approx 3$. The difference in value of β is due to the fact that we simulated a collision at temperature 1.3 K while the simulations from [37] were performed at temperatures between 2.01 and 2.14 K. It is likely that the value of the coefficient β depends on the temperature through the coefficients of mutual friction α, α' . In Figure 2.3, we can also notice that the vortex rings approach each other linearly, with velocity $8.4 \times 10^{-2} \text{ cm.s}^{-1}$. This is a special property for vortex rings, because they move at the self-induced velocity given by Equation 2.1. According to this relationship, their approach velocity should be $8.6 \times 10^{-2} \text{ cm.s}^{-1}$, which is in accordance with the result from the simulation.

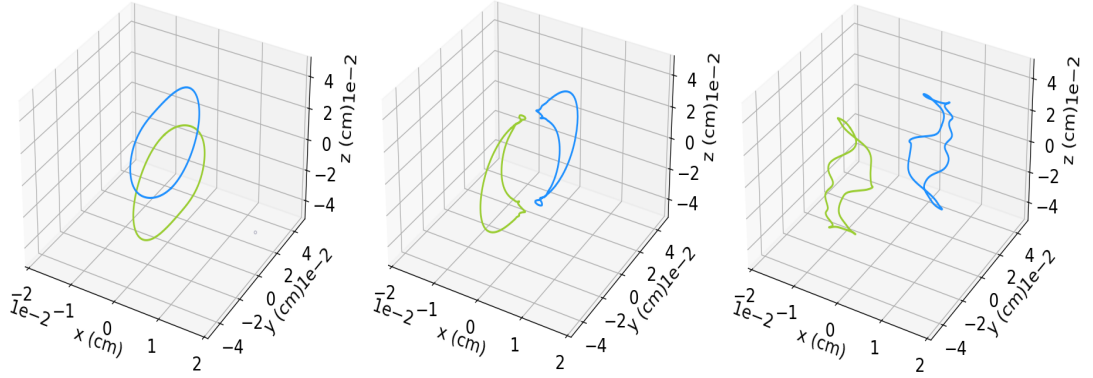


Figure 2.2: (Left) Two rings before collision. (Middle) Two rings after collision. (Right) Induced Kelvin waves.

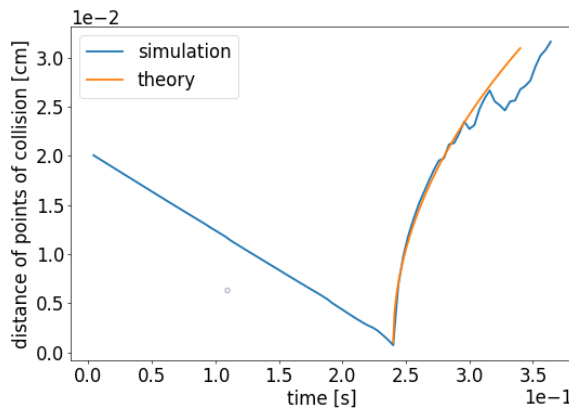


Figure 2.3: Evolution of distance of the reconnected points.

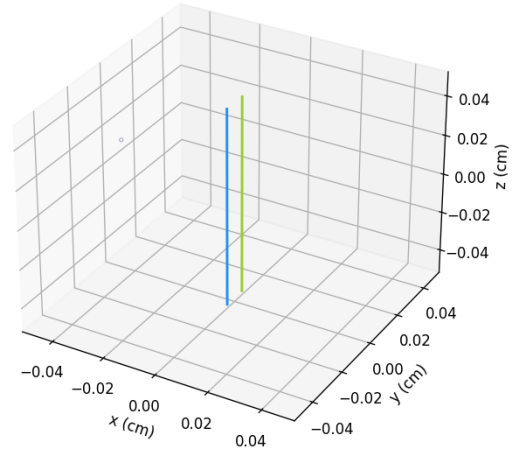


Figure 2.4: Two line vortices with opposite directions.

2.1.3 Two lines

If two vortex lines are placed close to each other, they will interact with each other through their induced velocities. Similarly in magnetostatics, where we can analytically calculate the magnetic field for two straight conductors, we can analytically calculate the velocity field of a line vortex at a distance d as [36]

$$v_s^{(line)}(d) = \frac{\kappa}{2\pi d}. \quad (2.3)$$

If we place two line vortices close to each other, two different movements will occur depending on their mutual direction. If the line vortices are oriented in the same direction, they will orbit around their common center at an angular velocity given by

$$\omega(d) = \frac{\kappa}{\pi d^2}, \quad (2.4)$$

what we obtained by dividing Equation 2.3 with their orbital radius $d/2$. If the line vortices are oriented in the opposite direction, they will move in a straight line with the speed given by the Equation 2.3. Due to the dissipation of energy, which occurs at non-zero temperature and manifests itself in internal friction, the mutual distance of line vortices will decrease. This allows us to numerically simulate both cases of motion and thus verify the functionality of our code. Both simulations are carried out in rectangular geometry between two solid walls, while the rest 4 boundaries are periodic. The main difference compared to other simulations is the necessity to use pinning mode PINNED_SLIP, so that the endpoints can move freely along the wall. The temperature was set to be 1.3 K, time step $dt = 2 \times 10^{-5}$ s and discretization distances of points was in the interval $dl_{min} = 5 \times 10^{-4}$ cm to $dl_{max} = 2 \times 10^{-3}$ cm. The initial distance of the line vortices was 0.1 mm, as we can see in Figure 2.4. Obtained dependencies for both types of motion are shown in Figure 2.5, and are in agreement with Equations 2.3 and 2.4. Deviations from theoretical dependencies are due to excited Kelvin waves, which are always excited at finite temperatures. The waves were excited in the center of the vortices, away from the walls, and amplified until the vortices reconnected. The direction of the excited waves was different for both simulations, which can be seen from the different profiles of the deviations from the theoretical dependencies.

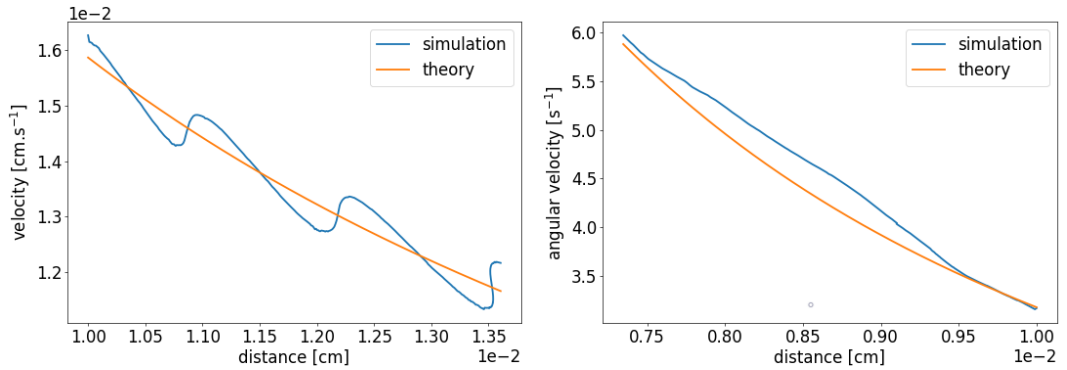


Figure 2.5: Two line vortices oriented in the opposite direction (left), in the same direction (right).

2.2 Rectangular geometry

We chose rectangular geometry to test the correct behavior of simulations of a large number of quantized vortices. We compare our results with the experiments performed by Varga, Babuin and Skrbek [38], and with the simulations performed by Adachi, Fujiyama and Tsubota [39]. All simulations in rectangular geometry had a time step $dt = 10^{-4}$ s, we saved the simulation every 50 steps, the simulations were carried out at temperature $T = 1.3$ K and the discretization distances of the points were in the interval $dl_{min} = 5 \times 10^{-4}$ cm to $dl_{max} = 1.5 \times 10^{-3}$ cm. We examined counterflows with 5 different velocities $u_{ns} = u_n - u_s$: 4 cm.s⁻¹, 5 cm.s⁻¹, 6 cm.s⁻¹, 7 cm.s⁻¹ and 8 cm.s⁻¹, in 3 different geometries: 6 periodic boundaries, 4 periodic boundaries and 2 solid walls (i.e. counterflow between 2 parallel walls), 2 periodic boundaries and 4 solid walls (i.e. counterflow inside a channel / capillary). The dimensions of the simulated volume $0.1 \times 0.1 \times 0.1$ mm³ were the same for all geometries. At the beginning of every simulation we randomly placed few randomly oriented vortex rings (always less then 30) into the investigated volume. This initial conditions were far from homogeneous, however it was important to show that different initial conditions always led to the same result. This was proven by examining density and polarization profiles after reaching counterflow equilibrium.

2.2.1 Vortex density

The turbulence is best described with its density. In Figure 2.6 we can see the time evolution of vortex density, i.e. vortex length in the unit volume. All the simulations reached equilibrium, however the simulations with high densities (10^5 cm⁻¹ and more) are numerically too complex and one time step takes several hours. Therefore, we were not able to simulate the equilibrium state over a longer time interval for these simulations. We can see that the simulations reached almost same turbulence densities regardless of geometry. Another property of vortex density is its profile in different configuration of the solid walls. As we can see in Figure 2.10, density always decreases near the walls under 70% of its average value. The vortex line densities as functions of counterflow velocity obtained from the simulations are in consistence with experimental result, $L = \gamma^2 u_{ns}^2$ [38], as we can see in Figure 2.7. The parameters γ we obtained from our simulations are 63.58 s/cm² (periodic), 62.07 s/cm² (parallel planes) and 60.49 s/cm² (channel). Adachi, Fujiyama and Tsubota [39] obtained from their simulations $\gamma = 53.5$, which is less than the values we obtained, however our results lies between the two fits indicated in Figure 1.8 and therefore agrees with experimentally measured values. The probable reason for the difference compared to Adachi, Fujiyama and Tsubota simulations is that we used Barnes-Hut tree approximation, low-pass filter, and did not simulate equilibrium over a sufficiently long time interval. As we can see in Figure 2.7 this difference decreases with increasing counterflow velocity. In the experiments of Varga, Babuin and Skrbek, significantly larger L are obtained [38], likely due to the effects of normal fluid turbulence.

2.2.2 Polarization

The second property we were interested in, was the profile of vortex polarization of the turbulence. The example of the polarization profiles for the counterflow $u_{ns} = 8 \text{ cm.s}^{-1}$ along the x-axis inside the channel is in Figure 2.11. We can see the polarization of evolving flow is not always homogeneous, because of the initial condition. However the moment polarization becomes homogeneous simulation evolves the same way irrespective of the initial condition towards the steady flow. We verified that the polarization is weaker in the flow direction, as it was noted first by Schwarz [17] and then confirmed in the experiments [40]. Therefore, the counterflow tangle is not isotropic and it looks like the polarization profile does not depend on the configuration of the walls of the rectangular geometry, neither on the counterflow velocity u_{ns} , but solely on the direction of the counterflow. In Table 2.8 we can see the final polarization for all 15 simulations and in Figure 2.9 we can see the evolution of the polarization for $u_{ns} = 8 \text{ m.s}^{-1}$ counterflow in 3 different geometries.

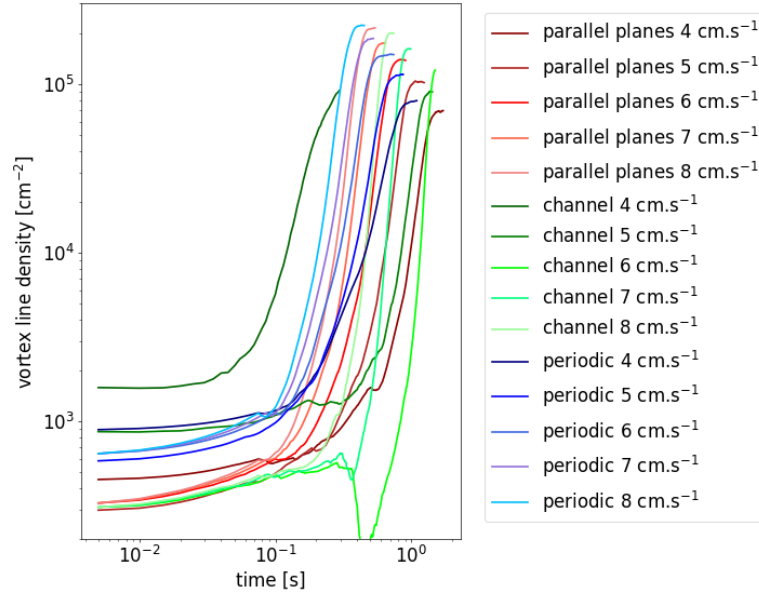


Figure 2.6: Density evolution of all the simulations in the rectangular geometry.

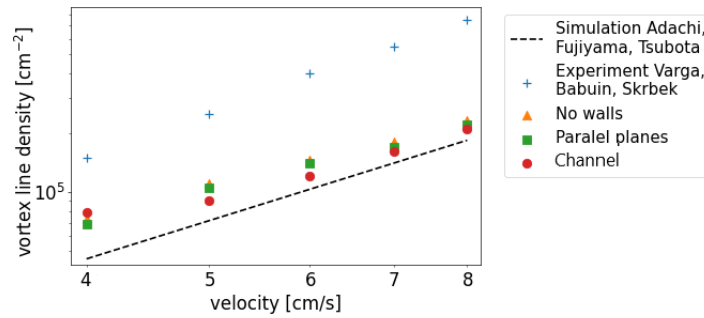


Figure 2.7: Comparison of equilibrium densities with experiment carried by Varga, Babuin and Skrbek [38], and simulation carried by Adachi, Fujiyama and Tsubota [39].

	u_{ns}	x[%]	y[%]	z[%]
periodic	4 cm.s ⁻¹	27.4	36.6	36.4
	5 cm.s ⁻¹	27.5	36.6	36.1
	6 cm.s ⁻¹	27.2	36.7	36.2
	7 cm.s ⁻¹	26.9	36.7	36.5
	8 cm.s ⁻¹	26.8	36.7	36.6
parallel walls	4 cm.s ⁻¹	27.4	36.7	36.1
	5 cm.s ⁻¹	27.4	36.6	36.4
	6 cm.s ⁻¹	27.1	36.5	36.4
	7 cm.s ⁻¹	26.7	36.5	36.7
	8 cm.s ⁻¹	26.6	36.7	36.6
channel	4 cm.s ⁻¹	28.5	36.2	35.3
	5 cm.s ⁻¹	26.6	36.6	36.8
	6 cm.s ⁻¹	26.4	36.5	37.1
	7 cm.s ⁻¹	26.6	36.6	36.8
	8 cm.s ⁻¹	26.3	36.9	36.8

Figure 2.8: Equilibrium values of polarization of all the simulations in the rectangular geometry.

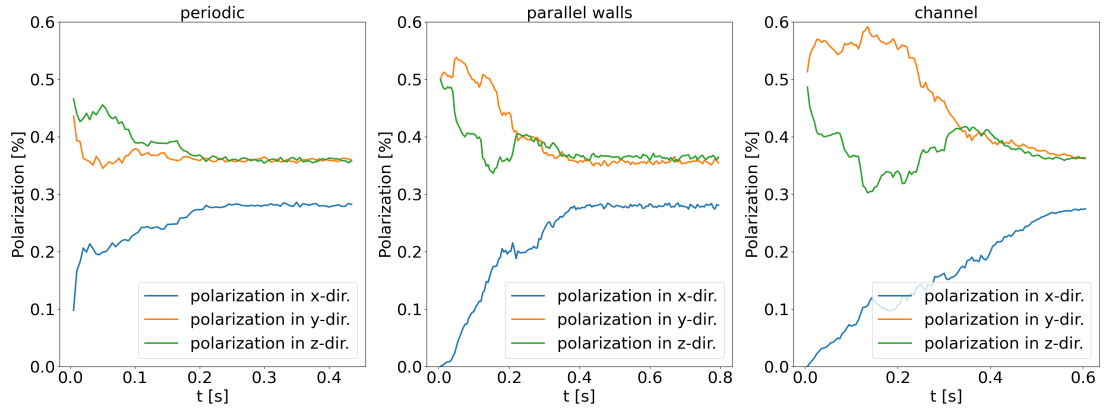


Figure 2.9: Polarization evolution in $u_{ns} = 8 \text{ m.s}^{-1}$ counterflow for 3 different setups.

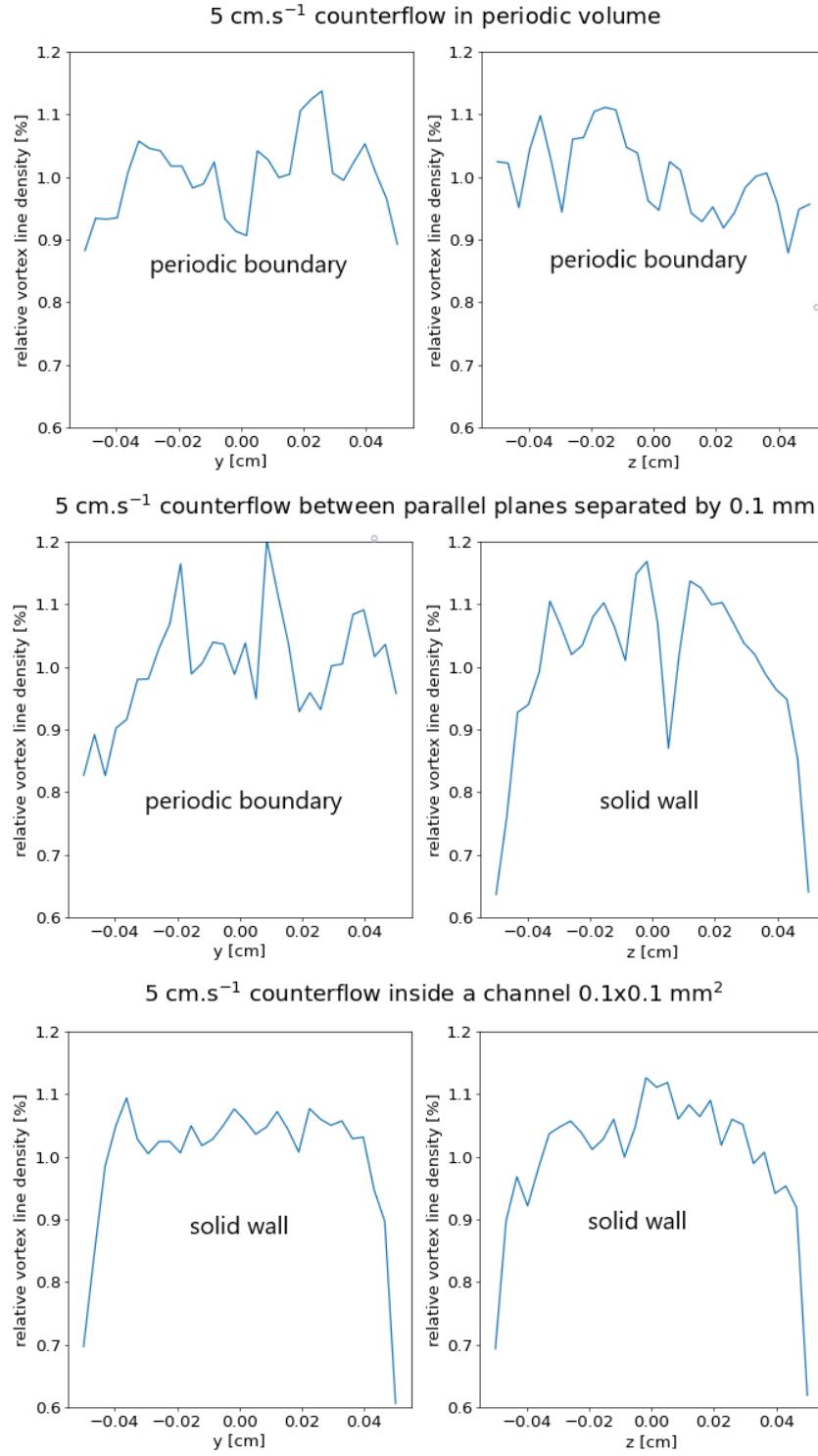


Figure 2.10: Relative density profiles in the directions perpendicular to 5 cm.s^{-1} counterflow in the volume with periodic boundaries (top), between parallel walls (middle) and inside a channel (bottom).

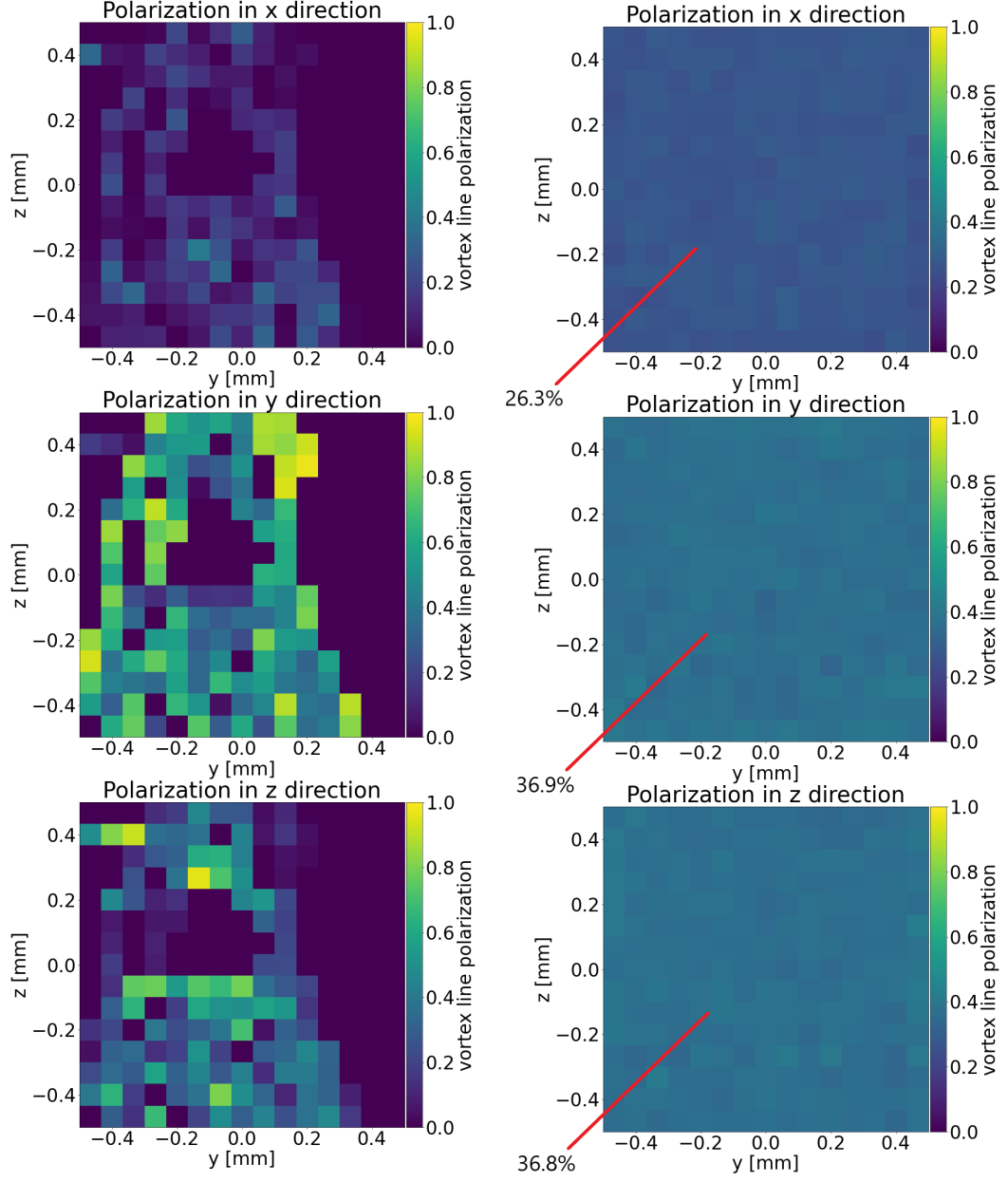


Figure 2.11: Polarization of turbulence in $u_{ns} = 8 \text{ m.s}^{-1}$ counterflow inside a channel averaged over all y-z planes of volume. (Left) Evolving turbulence. (Right) Turbulence in equilibrium.

2.3 Spherical geometry

For the simulations in spherical geometry we used the same setting as in the rectangular one, i.e.: we had a time step $dt = 10^{-4}$ s, we saved the simulation every 50 steps, the simulations were carried out at bulk temperature $T_\infty = 1.65$ K and the discretization distances of points were in the interval $dl_{min} = 5 \times 10^{-4}$ cm to $dl_{max} = 1.5 \times 10^{-3}$ cm. We examined counterflow at 4 different power outputs \dot{Q} : 30 mW, 50 mW, 100 mW and 200 mW. We simulated the turbulence in the periodic spherical section with an inner solid wall (heater) of the radius 0.1 mm and the outer boundary was chosen to be open and with the radius 0.5 mm (i.e. after the whole vortex was outside of this radius, we deleted it). The examined volume had quasi-rectangular base with both sides having 20° , see Figure 1.7. At the beginning of every simulation we randomly placed 30 randomly oriented vortex rings into the investigated volume, but regardless of the initial condition the simulation always lead to the same result. We also simulated the whole spherical volume for the power output 30 mW (at higher power the turbulence is too dense to be simulated in short enough time), with what we showed that we can use the periodic boundary condition to decrease computational time of the simulation.

2.3.1 Vortex density

The fundamental difference between the counterflow in the rectangular and spherical geometry is, that according to [27], we have to take into account the radial temperature dependence given by Equation 1.25 to reach equilibrium in the spherical counterflow. The radial dependence of velocity is given by Equation 1.22, therefore the turbulence density will have a radial profile. The radial profile of vortex density is shown in Figure 2.13. The density decreases approximately with 4th power of radius, as we would expect from Vinen's equation 1.10. The small differences compared to the decrease with the fourth power are in line with our expectations, partly due to the temperature profile, which is however not very significant, but also due to the possible influence of the solid wall. In Figure 2.12 we can see the time evolution of vortex densities in the vicinity of the heater. Due to the radial dependence of the density, we analyzed 5 layers of the turbulence. This time, not all simulations reached equilibrium, even though they have been running in the computational center MetaCentrum [41] for several weeks. For example, a 200 mW simulation ran there for over 1000 hours on 32 cores. We were able to run simulations at 30 mW and 50 mW on a 10 times larger time scale than simulations at 100 mW and 200 mW. It is clear from the time evolution as well as from the radial density profiles that the turbulence reaches equilibrium gradually, first at the lowest radii and then at the larger ones. At the outer radii of the observed volume, the turbulence never reached equilibrium, due to the low counterflow velocity. The vortex line density here is an order of magnitude smaller than in the heater area and vortex dynamics is less intensive, therefore the turbulence has not changed appreciably since the beginning of the simulation. This results in an increasing amount of noise on the radial dependence of the turbulence density and polarization with increasing distance from the heater. In contrast to the Inui and Tsubota simulations [42], we do not observe a spheri-

cal shell of quantized vortices with low vortex turbulence density directly at the heater, which is probably due to our imperfect implementation of the boundary condition using the image vortices behind the solid wall.

2.3.2 Polarization

As in rectangular geometry, we also evaluated the final polarization in spherical geometry. In Table 2.1 we can see that the polarization of the turbulence is suppressed in the radial direction, i.e. that it behaves similarly to the counterflow in rectangular geometry. The polarization was evaluated in a 0.05 mm wide layer in close proximity to the heater, because a weak counterflow at large distances does not allow us to reach an equilibrium state in the whole volume. We can notice that the difference between the polarization in the radial direction and the angular directions is initially random and during evolution of turbulence is larger than in equilibrium. This can be seen in Figure 2.15, which shows the evolution of polarization in a 0.02 mm layer in the vicinity of heater. Therefore, the polarization values in Table 2.1 for 100 mW and 200 mW cannot be understood as equilibrium values.

\dot{Q} mW	r [%]	θ [%]	ϕ [%]
30	21.68	38.73	39.59
50	22.43	38.45	39.12
100	19.60	40.26	40.14
200	15.03	42.49	42.48

Table 2.1: Reached values of polarization for different power outputs of the heater averaged over the 0.05 mm layer in the vicinity of the heater. For the case of 30 mW and 50 mW we can take the polarization values as equilibrium values, and for the cases of 100 mW and 200 mW (red) the turbulence is still evolving.

2.3.3 Tests of periodic boundaries in spherical geometry

As we mentioned in section 1.3.2, there is the problem of how to periodically move vortex points that cross the periodic boundary. Therefore, we must verify that the way we periodically move points in spherical geometry does not change the physics we simulate. First of all, we can look at the angular distribution of polarization and density in the spherical layer near the heater. In Figure 2.15, we see that periodic boundary conditions do not create inhomogeneities of polarization and density in the spherical layer. Furthermore, we simulated turbulence in the whole sphere at power output $\dot{Q} = 30$ mW. We compared the resulting values of polarization, averaged over the 0.05 mm layer in the vicinity of the heater, in Table 2.2. Comparisons of the simulation in the whole sphere and in the spherical section can be seen in Figure 2.16. There we compared time evolution and the radial profile of vortex line density. The simulation in the whole sphere did not reach equilibrium due to the overall length of the vortices being too high. However, we are working on the simulation of the equilibrium state in the whole sphere and we will make a better comparison of the simulations in the

future. From Figure 2.16 we can see that the radial profile of turbulence density is likely to remain r^{-4} also for the whole sphere equilibrium simulation.

	polarization		
	r [%]	θ [%]	ϕ [%]
section	21.68	38.73	39.59
full sphere	22.89	38.09	39.02

Table 2.2: Reached values of polarization for simulation in spherical section and full sphere for power outputs of 30 mW. Values are averaged over the 0.05 mm layer in the vicinity of the heater.

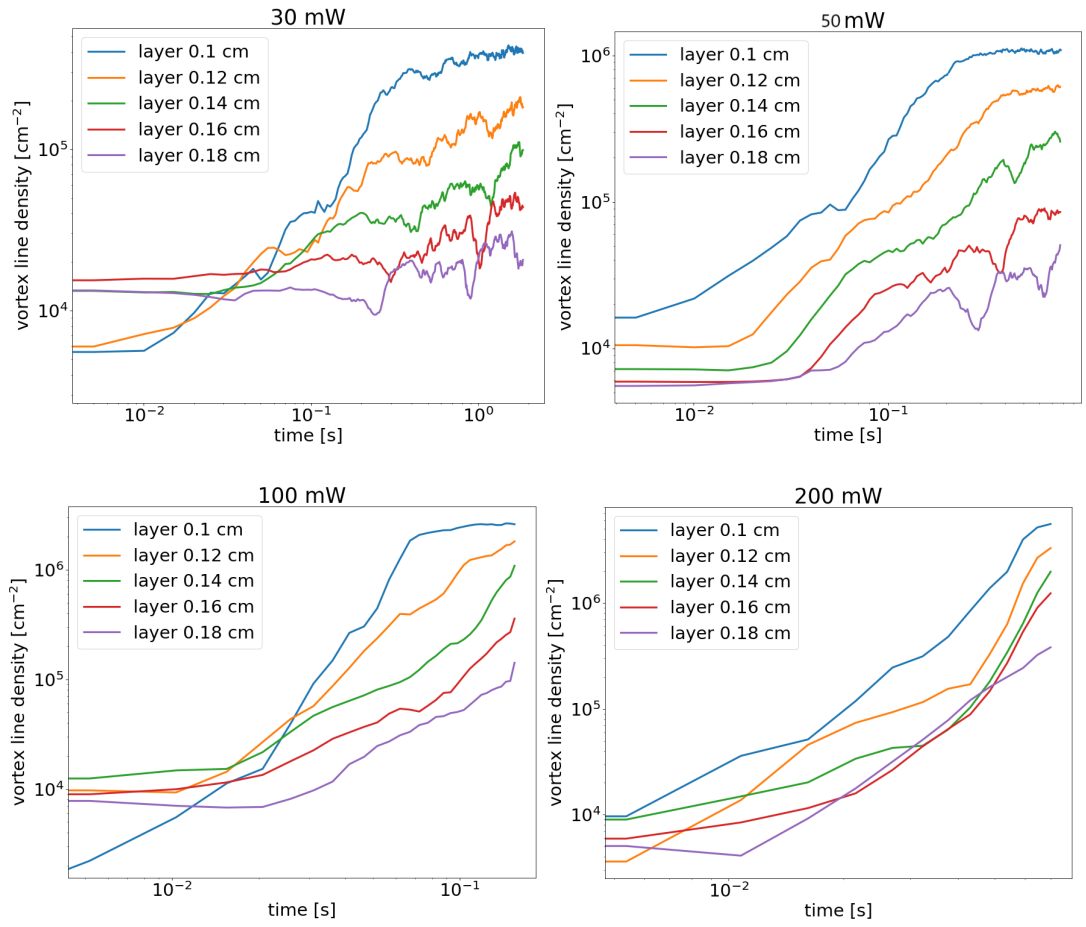


Figure 2.12: Density evolution in the 5 closest layers to the heater in a spherical counterflow.

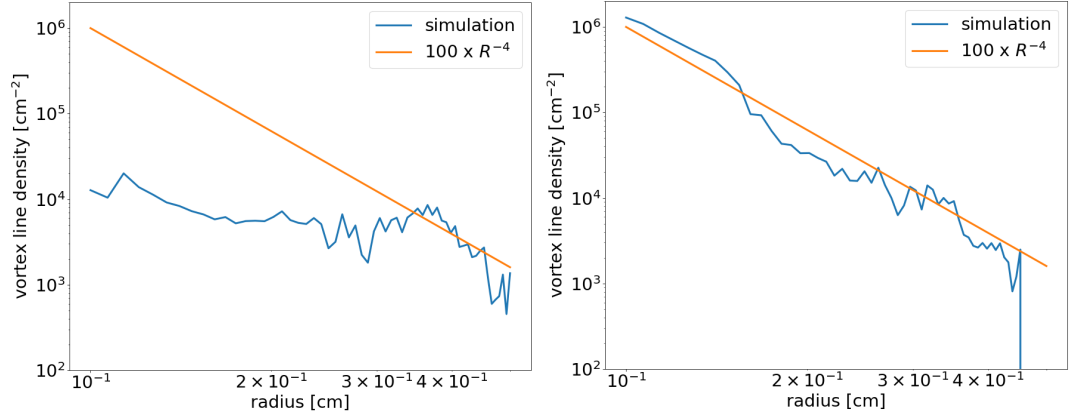


Figure 2.13: Density profile in spherical counterflow around 50 mW heater. Left: Evolving turbulence. Right: Turbulence in equilibrium.

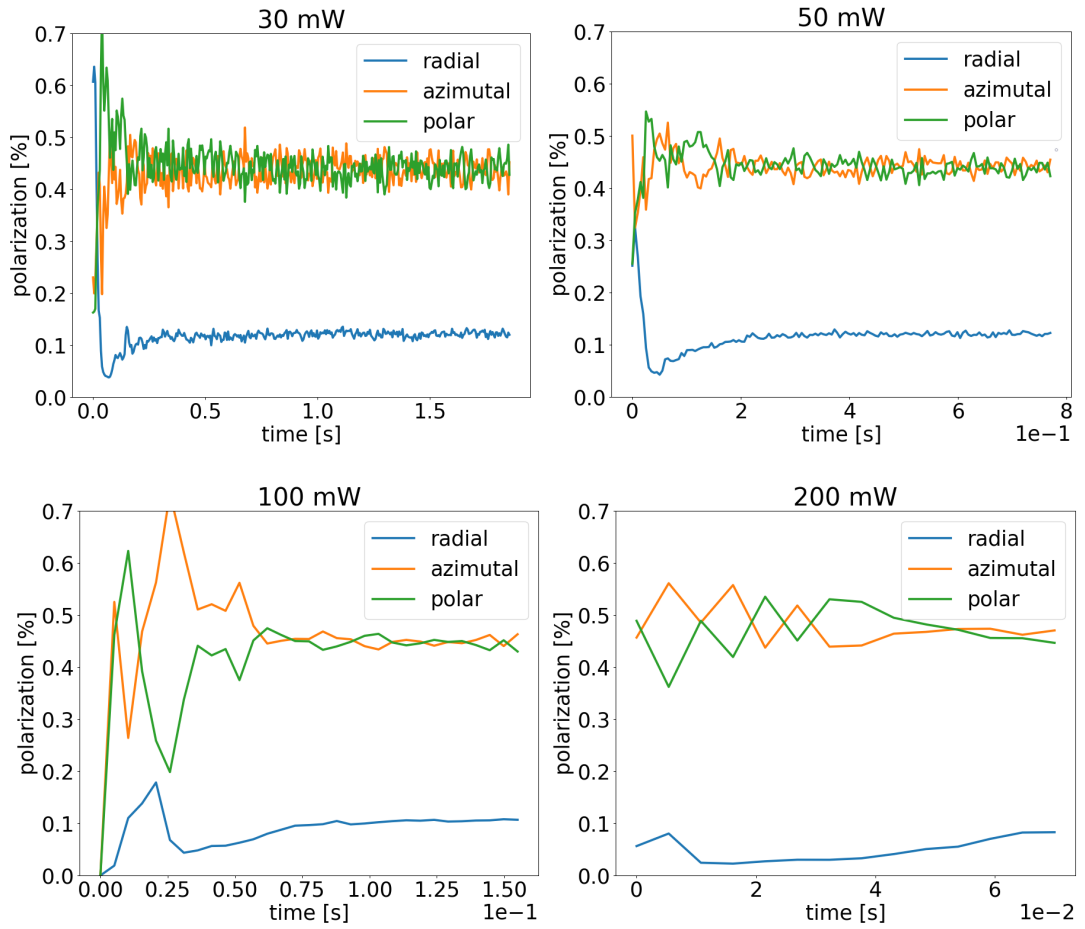


Figure 2.14: Polarization evolution in the vicinity of the heater (layer of width 0.02 mm) in a spherical counterflow. We can notice that for lower power output we are able to simulate turbulence at orders of magnitude higher times.

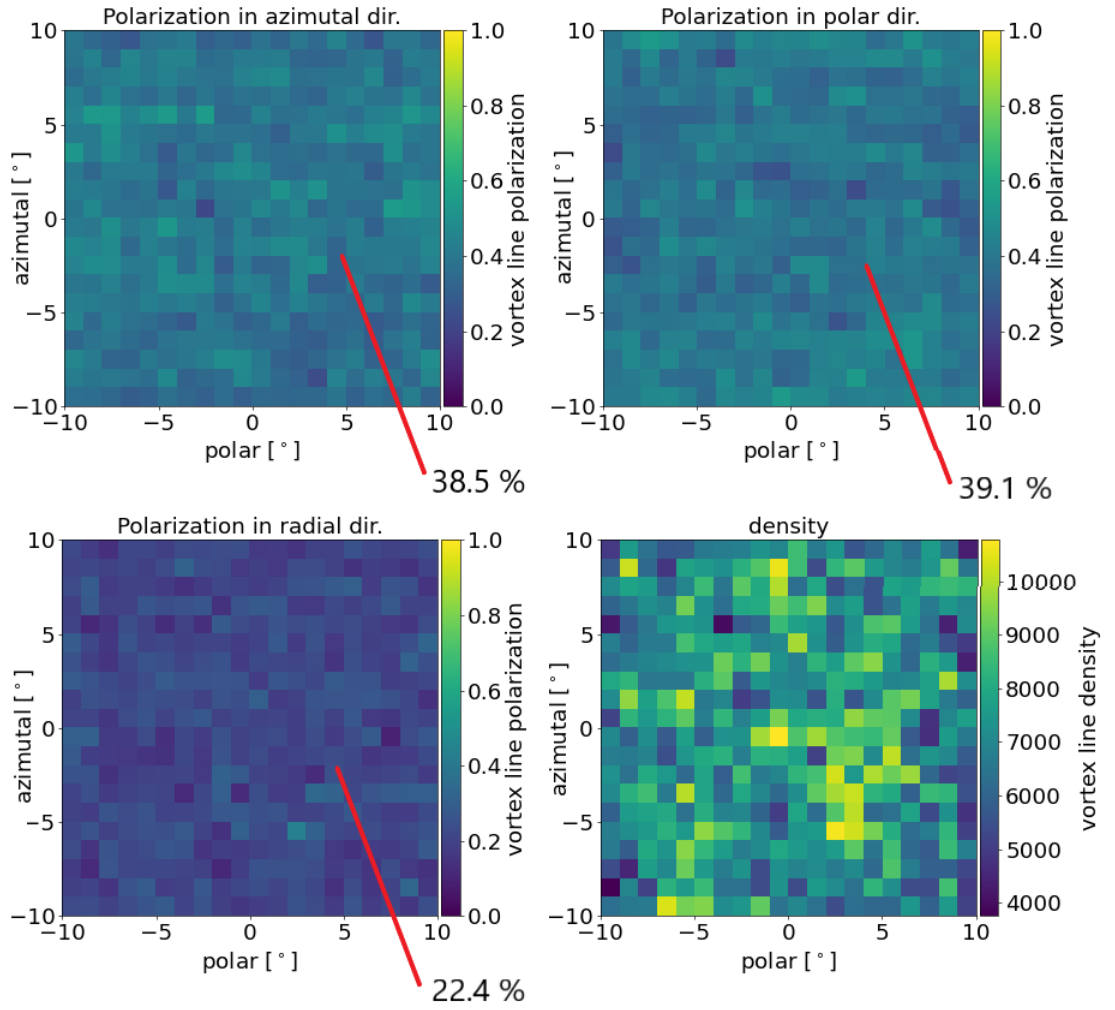


Figure 2.15: Polarization and density of turbulence in the spherical layer near the heater with power output of 50 mW.

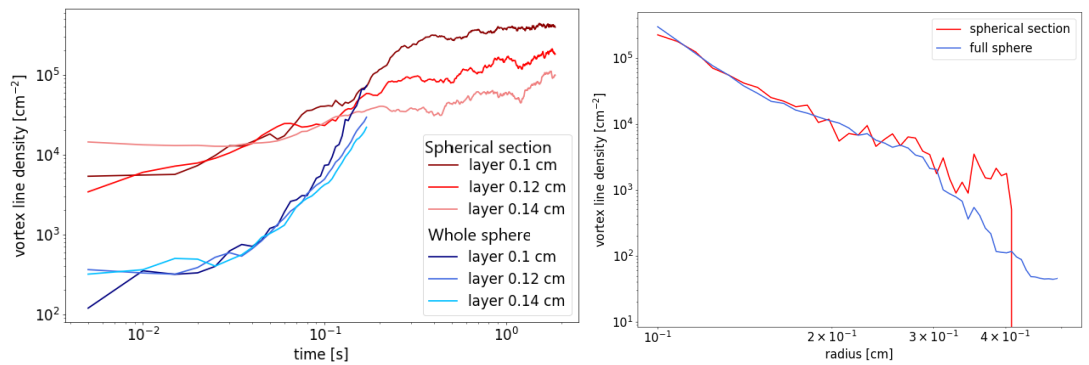


Figure 2.16: Comparison of time evolution (left) and the radial profile in time 10⁻¹ s (right) of turbulence density for spherical section and full sphere simulations inside the layer near the heater at heater's power output 30 mW.

Conclusion

In this thesis, we have summarized our work on numerical simulations of thermal counterflow turbulence using the vortex filament model with the use of Barnes-Hut algorithm. First, we verified an existing code developed by E. Varga and improved the code by finishing the Barnes-Hut tree algorithm, adding a low-pass filter and partially parallelized it for use in larger data center clusters. Later we investigated the properties of statistical equilibrium state of counterflow turbulence in two different geometries.

Our numerical results in rectangular geometry revealed the characteristic relation $L = \gamma u_{ns}^2$, which has been observed in many experiments, and the parameter γ approximately agrees with experimentally measured values by Varga, Babuin and Skrbek [38], and with older simulations carried by Adachi, Fujiyama and Tsubota [39]. The behavior of this type of flow corresponded to known data. The differences compared to the real counterflow are mainly due to the neglect of the turbulence in the normal component of liquid Helium.

Later, we adapted the code to spherically symmetric counterflow where the spatial temperature distribution also plays an important role and must therefore be considered as an integral part of the simulation. We observed a gradual development of turbulence, first in layers near the heater. Here, we again observed the characteristic relation $L = \gamma u_{ns}^2$. Finally, we implemented spherical geometry and observed that the polarization and radial profile of turbulence density remained the same for spherical section and whole sphere simulations during the evolution of the turbulence. However, we were unable to verify this for steady state.

Bibliography

- [1] European Aviation Safety Agency. Safety analysis research department: Annual safety review 2014. pages 42–47, 2015.
- [2] Robert I. Bowles. Transition to turbulent flow in aerodynamics. *Philosophical Transactions: Mathematical, Physical and Engineering Sciences*, 358(1765):245–260, 2000.
- [3] L.F. Richardson. Weather prediction by numerical processes. *Cambridge University Press*, page 66, 1922.
- [4] M.T. Landahl and E. Mollo-Christensen. *Turbulence and Random Processes in Fluid Mechanics*. Cambridge University Press, 2 edition, 1992.
- [5] L. Skrbek and col. *Fyzika nízkých teplot, I. část*. MATFYZPRESS, Prague, 2011.
- [6] H. Kamerlingh Onnes. The liquefaction of helium. *Communications from the Physical Laboratory at the University of Leiden*, 9(108):1–23, 1908.
- [7] P. Kapitza. Viscosity of liquid helium below the λ -point. *Nature*, 141:74, 1938.
- [8] J.F. Allen; A.D. Misener. Flow of liquid helium ii. *Nature*, 141:75, 1938.
- [9] W.H. Keesom; G.E. Macwood. The viscosity of liquid hydrogen. *Physica*, 5:745–748, 1938.
- [10] L. Tisza. Transport phenomena in helium ii. *Nature*, 141:913, 1938.
- [11] L.D. Landau. The theory of superfluidity of helium ii. *Physical Review*, 60:356–358, 1941.
- [12] J. Bahyl. Measurement of quantum turbulence in superfluid he-4. *Student conference, FMPH UK, Bratislava*, 2016.
- [13] F. London. *Nature*, 141:643, 1938.
- [14] O. Penrose and L. Onsager. Bose-einstein condensation and liquid helium. *Phys. Rev.*, 104:576, 1956.
- [15] R. Feynman. Application of quantum mechanics to liquid helium. *Prog. in Low Temp. Phys.*, 1:17–53, 1955.
- [16] F. Barenghi. Superfluid vortices and turbulence. *Quantized Vortex Dynamics and Superfluid Turbulence*, page Chap.1, 2001.
- [17] K.W. Schwarz. Three-dimensional vortex dynamics in superfluid he 4: Homogeneous superfluid turbulence. *Phys Rev B*, 38:2398, 1988.
- [18] D.C. Samuels. Superfluid vortices and turbulence. *Quantized Vortex Dynamics and Superfluid Turbulence*, page Chap.9, 2001.

- [19] W.F. Vinen; H.E. Hall. The theory of mutual friction in uniformly rotating helium ii. *Proc. Royal Soc. London*, 238:204, 1957.
- [20] J. Barnes; P. Hut. A hierarchical $O(n \log n)$ force-calculation algorithm. *Nature*, 324:446–449, 1986.
- [21] A.W. Baggaley and C.F. Barenghi. 166:3, 08 2012.
- [22] J. Bahyl. Drag force scaling and quantum turbulence in superfluid helium. *Diploma Thesis, FMPH UK, Bratislava*, page 54, 2018.
- [23] A.W. Baggaley; C.F. Barenghi. Tree method for quantum vortex dynamics. *J. Low Temp. Phys.*, 166:3–20, 2012.
- [24] Emil Varga. Peculiarities of spherically symmetric counterflow. *Journal of Low Temperature Physics*, 196, 07 2019.
- [25] Matteo Frigo; Steven G. Johnson; MIT. Fftw library. <http://fftw.org/>.
- [26] A. Savitzky; Golay. Smoothing and differentiation of data by simplified least squares procedures. *Analytical Chemistry*, 36:1627–1639, 1964.
- [27] E. Rickinson, C. F. Barenghi, Y. A. Sergeev, and A. W. Baggaley. Superfluid turbulence driven by cylindrically symmetric thermal counterflow. *Phys. Rev. B*, 101:134519, Apr 2020.
- [28] R. Hänninen, A. Mitani, and M. Tsubota. Superfluid $^3\text{He-B}$ vortex simulations inside a rotating cylinder. *Journal of Low Temperature Physics - J LOW TEMP PHYS*, 138:589–594, 02 2005.
- [29] Russell J. Donnelly and Carlo F. Barenghi. The observed properties of liquid helium at the saturated vapor pressure. *Journal of Physical and Chemical Reference Data*, 27(6):1217–1274, 1998.
- [30] Z. Xie, Y. Huang, and F. et al. Novotný. Spherical thermal counterflow of He-II . *Journal of Low Temperature Physics*, 42, 2022.
- [31] S. Babuin, M. Stammeier, E. Varga, M. Rotter, and L. Skrbek. Quantum turbulence of bellows-driven ^4He superflow: Steady state. *Phys. Rev. B*, 86:134515, Oct 2012.
- [32] J. T. Tough, R. A. Ashton, and L. B. Opatowsky. Superfluid turbulence in counterflow and pure superflow. *Physica B-condensed Matter*, 108:1127–1128, 1981.
- [33] T. V. Chagovets and Ladislav Skrbek. On flow of He II in channels with ends blocked by superleaks. *Journal of Low Temperature Physics*, 153:162–188, 2008.
- [34] K. P. Martin and J. T. Tough. Evolution of superfluid turbulence in thermal counterflow. *Phys. Rev. B*, 27:2788–2799, Mar 1983.

- [35] R. K. Childers and J. T. Tough. Helium ii thermal counterflow: Temperature- and pressure-difference data and analysis in terms of the vinen theory. *Phys. Rev. B*, 13:1040–1055, Feb 1976.
- [36] R. J. Donnelly. *Quantized Vortices in Helium II*. 1991.
- [37] Gregory P. Bewley, Matthew S. Paoletti, Katepalli R. Sreenivasan, and Daniel P. Lathrop. Characterization of reconnecting vortices in superfluid helium. *Proceedings of the National Academy of Sciences*, 105(37):13707–13710, sep 2008.
- [38] Emil Varga, Simone Babuin, and L. Skrbek. Second-sound studies of coflow and counterflow of superfluid ^4He in channels. *Physics of Fluids*, 27(6):065101, 2015.
- [39] Hiroyuki Adachi, Shoji Fujiyama, and Makoto Tsubota. Steady-state counterflow quantum turbulence: Simulation of vortex filaments using the full biot-savart law. *Phys. Rev. B*, 81:104511, Mar 2010.
- [40] R. T. Wang, C. E. Swanson, and R. J. Donnelly. Anisotropy and drift of a vortex tangle in helium ii. *Phys. Rev. B*, 36:5240–5244, Oct 1987.
- [41] z.s.p.o. CESNET. Metacentrum (metavo) – virtual organization. <https://metavo.metacentrum.cz/en/>.
- [42] Sosuke Inui and Makoto Tsubota. Spherically symmetric formation of localized vortex tangle around a heat source in superfluid ^4He . *Phys. Rev. B*, 101:214511, Jun 2020.

List of Figures

1.1	Phase diagram of He-4.	3
1.2	Temperature dependence of densities of the normal (red) and superfluid (blue) component densities [12].	4
1.3	Vortex tangle simulated in counterflow between two parallel planes.	9
1.4	Image tangles scheme for simulation of counterflow between two parallel walls. Bold lines represents simulated volume, eight squares around the central one represents the image tangles, blue lines represents solid parallel walls and red lines represents vortex lines. We can see that solid walls acts as mirrors.	9
1.5	Scheme of vortex segments reconnection. (a) Two vortex lines before reconnection. (b) Vortex lines after reconnection [22].	10
1.6	Scheme of tree structure construction in 2D. The points (blue) are located inside the investigated volume (a), which is divided into four child-boxes of half size (b), until (d) there is only one point per child-box [23].	10
1.7	Vortex tangle simulated in spherical section with square base.	14
1.8	The parameter γ : The dashed line is a linear fit $\gamma(T) = 155.8T - 75$ and the solid line is a linear fit $\gamma(T) = 74.9T - 70.7$ [31]. Presented data: (i) solid blue circles, solid green squares, open blue up-triangle, Babuin et al. [31]; (ii) open squares, Ashton et al. [32]; (iii) solid red down-triangle, Chagovets and Skrbek [33]; (iv) solid blue up-triangle, solid magenta diamond, Martin and Tough [34]; (v) open circles, Childers and Tough [35].	14
2.1	Simulation of one vortex ring: without smoothing (left), with low pass filter (right).	15
2.2	(Left) Two rings before collision. (Middle) Two rings after collision. (Right) Induced Kelvin waves.	16
2.3	Evolution of distance of the reconnected points.	16
2.4	Two line vortices with opposite directions.	16
2.5	Two line vortices oriented in the opposite direction (left), in the same direction (right).	17
2.6	Density evolution of all the simulations in the rectangular geometry.	19
2.7	Comparison of equilibrium densities with experiment carried by Varga, Babuin and Skrbek [38], and simulation carried by Adachi, Fujiyama and Tsubota [39].	19
2.8	Equilibrium values of polarization of all the simulations in the rectangular geometry.	20
2.9	Polarization evolution in $u_{ns} = 8 \text{ m.s}^{-1}$ counterflow for 3 different setups.	20
2.10	Relative density profiles in the directions perpendicular to 5 cm.s^{-1} counterflow in the volume with periodic boundaries (top), between parallel walls (middle) and inside a channel (bottom).	21

2.11	Polarization of turbulence in $u_{ns} = 8 \text{ m.s}^{-1}$ counterflow inside a channel averaged over all y-z planes of volume. (Left) Evolving turbulence. (Right) Turbulence in equilibrium.	22
2.12	Density evolution in the 5 closest layers to the heater in a spherical counterflow.	25
2.13	Density profile in spherical counterflow around 50 mW heater. Left: Evolving turbulence. Right: Turbulence in equilibrium. . . .	26
2.14	Polarization evolution in the vicinity of the heater (layer of width 0.02 mm) in a spherical counterflow. We can notice that for lower power output we are able to simulate turbulence at orders of magnitude higher times.	26
2.15	Polarization and density of turbulence in the spherical layer near the heater with power output of 50 mW.	27
2.16	Comparison of time evolution (left) and the radial profile in time 10^{-1} s (right) of turbulence density for spherical section and full sphere simulations inside the layer near the heater at heater's power output 30 mW.	27

List of Tables

2.1	Reached values of polarization for different power outputs of the heater averaged over the 0.05 mm layer in the vicinity of the heater. For the case of 30 mW and 50 mW we can take the polarization values as equilibrium values, and for the cases of 100 mW and 200 mW (red) the turbulence is still evolving.	24
2.2	Reached values of polarization for simulation in spherical section and full sphere for power outputs of 30 mW. Values are averaged over the 0.05 mm layer in the vicinity of the heater.	25

Author's Contribution

The original version of the code we used to simulate turbulence was developed in 2019 by Emil Varga [24]. The code was in a state where it was able to simulate turbulence in rectangular geometry with open, periodic boundary conditions or solid walls using Biot-Savart law 1.13. At the same time, we were able to select different types of counterflow in the simulations. The contribution of this bachelor thesis includes:

1. Addition of code documentation.
2. Complete overhaul of simulation configurations settings.
3. Minor fixes to spatial discretization of the points.
4. Minor fixes to boundary conditions and periodic point shifts.
5. Completion of the implementation of the Barnes-Hut algorithm.
6. Implementation of the Low-pass filter.
7. Implementation of spherical geometry.
8. Implementation of temperature gradient in spherical geometry.
9. Partiall parallelization for use in larger data center clusters.
10. Code testing on selected simple problems such as vortex ring and straight line vortices.
11. Testing the newly implemented on selected simple problems such as vortex ring and straight line vortices.
12. Testing of numerically complex simulations in rectangular and spherical geometries on a larger data center cluster Metacentrum.

Appendix

The code is written in C and the main loop consists of:

1. **Save frame:** every 50th step we save the frame.
2. **Enforce wall boundaries:** here we apply solid walls, see Subsection 1.2.3. This goes first, because the vortex could have been initialized outside of the domain or the points could have moved behind the solid walls.
3. **Eliminate small loops:** enforcing wall boundaries can create small loops on the walls, which can interfere with the reconnections, low-pass filter and calculation of tangents and normals. Therefore, we must remove them.
4. **Reconnect:** we reconnect the vortices as described in Subsection 1.2.4.
5. **Eliminate small loops:** reconnect can create small loops, which can interfere with the low-pass filter and calculation of tangents and normals.
6. **Smooth:** here we apply the low-pass filter as described in Subsection 1.3.1
7. **Enforce periodic boundaries:** vortices could have escaped from the investigated volume across periodic boundaries and the low-pass filter algorithm destroys periodicity. Therefore, we have to apply periodic boundary conditions, see Subsection 1.2.3.
8. **Remesh:** remesh have to be applied before the calculation of tangents and normals to remove points close to each other.
9. **Eliminate small loops:** remesh can create small loops, which can interfere with the calculation of tangents and normals (see Subsection 1.2.1).
10. **Update tangents and normals:** updates tangents and normals, which are needed to calculate velocities.
11. **Update velocities:** first we calculate the contribution from the LIA velocity, then using the Barnes-Hut algorithm we calculate the induced velocity (see Subsection 1.2.5 and Equation 1.14) and finally we calculate the effect of mutual friction (see Equation 1.15). We repeat this also for image vortices (see Subsection 1.2.3).
12. **RK4 step:** finally, using the RK4 algorithm, we calculate the new positions of vortex points (see Subsection 1.2.2) and then we increment the simulation time.

Subsequent analysis was performed using python. The final version of the code is available at <https://github.com/robertjurco/OpenVort>.