# Vortex filament dynamics in quantum system

Jakub Bahyl[1]*

Supervisor: Emil Varga[2]†

[1] FMFI UK, Mlynská Dolina 842 48 Bratislava
[2] Katedra fyziky nízkych teplot, MFF UK, V Holešovičkách 747/2 Praha 8

**Abstract:** We present a numerical method to compute the evolution of vortex filaments in superfluid Helium II. The method is based on a Biot-Savart law and Local Induction Approximation. We test the method and its properties for a ring-shaped vortex configuration.

*Keywords:* Superfluidity · Vortex filament · Quantum vortex ring

## 1. Introduction to superfluidity

The liquid state of $^4$He exists in two phases:
He-I ($T > 2.17\,\mathrm{K}$) and He-II ($T < 2.17\,\mathrm{K}$). These two phases are connected with the *lambda transition*, which occurs at the critical temperature $T_\lambda = 2.17\,\mathrm{K}$. He-I is a classical fluid described by ordinary Navier-Stokes equation, whereas He-II indicates the Bose-Einstein condensate (superfluid) in much more way and therefore is described by complicated motion equations.

### Hydrodynamics of He-II

A simple phenomenological model of the He-II motion was proposed by Tisza and Landau [1][2] - the *two-fluid model*. This model depicts He-II as a mixture of two interpenetrating fluid components:

- normal component - density $\rho_n$, velocity field $\mathbf{v}_n$, ordinary viscous N-S equation of motion, carrying entropy and thermal excitations
- superfluid component - density $\rho_s$, velocity field $\mathbf{v}_s$, modified Euler equation (without viscosity) of motion by quantum terms, no entropy, represented by macroscopic wave function

The total density of He-II sums up to $\rho = \rho_n + \rho_s$ and the relative proportion of normal/superfluid component is determined by the temperature. Near $T \to 0$ Helium II becomes entirely superfluid $\rho_s/\rho \to 1$.

*bahylkubo@gmail.com
†varga.emil@gmail.com

According to experimental data [3] and Landau's microsocopic theory, the dependence of fractional densities on temperature is sketched in **Figure 1**.

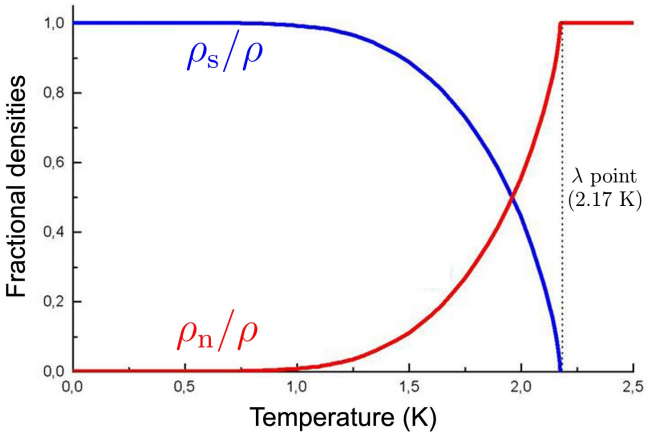


**Figure 1:** Temperature dependence of fractional densities of the normal (red) and superfluid (blue) components. Source: [4]

It arises from the quantum nature of superfluid, that superfluid component is *irrotational*: $\nabla \times \mathbf{v}_s = \mathbf{0}$. However, the circulation is *quantised* with the magnitude $\kappa = 9.97 \times 10^{-4}\,\mathrm{cm}^2 \cdot \mathrm{s}^{-1}$ and so-called *quantised vortices* can be created (**Figure 2**) when Helium II rotates or moves faster than a critical velocity.

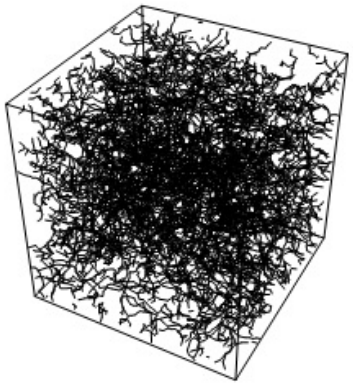

**Figure 2:** Cube of numerically simulated tangle of randomly distributed quantised vortices. Source: [4]

In classical fluid dynamics, one of the useful numeric tools is the *vortex filament model*. With the rapid development of computational power, large simulations have become the methods of choice for calculating the motion of fluids. Particularly in superfluids like He-II, due to the quantised circulation, vorticity can only exist within vortex filament with a certain core size, which makes the model a way simpler and applicable than in fields of classical fluids.

### Quantum vortex

The key properties of Feynman's [5] quantised vortex are the quantized circulation $\varkappa$, superfluid rotational velocity field $\mathbf{v}_s(\mathbf{r})$ and the *vortex core parameter* $a_0 \approx 10^{-10}\,\mathrm{m} = 1\,\text{Å}$, which represents the characteristic distance over which superfluid density $\rho_s$ drops from bulk value to zero. (a vortex radius)

A special case of vortex line configuration is a freely moving *vortex ring* (**Figure 3**). Such rings are usually created as a result of multi-vortex interconnection and have limited life expectancy. The exact expressions derived by classical hydrodynamics [7] for the energy $E_{\text{ring}}$ and center velocity $v_{\text{ring}}$, moving in a He-II of density $\rho$ and having a radius $R$ much greater than its core radius $R >> a_0$, are

$$E_{\text{ring}} = \frac{1}{2}\varkappa^2 \rho R\Big(\ln(8R/a_0) - 2 + c\Big) \qquad (1)$$

$$v_{\text{ring}} = \frac{\varkappa}{4\pi R}\Big(\ln(8R/a_0) - 1 + c\Big), \qquad (2)$$

where $c$ is a constant based on inner structure of the vortex. Since we assume a hollow core, $c = 0$. Equations (1), (2) apply only for temperature $T = 0$. In a real case $T > 0$, there are dissipation processes caused by the *mutual friction* with a fluid around and velocity (2) scales with $(1 - \alpha')$, where $\alpha'$ is a temperature dependent friction constant.

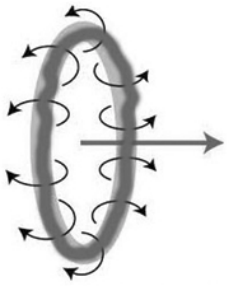

**Figure 3:** Depiction of quantized vortex ring motion and induced velocity field. Source: internet

## 2. Vortex filament model

The vortex line can be represented as a curve with positional vector $\mathbf{s} = \mathbf{s}(\xi, t)$ in a three-dimensional space. Here, $\xi$ is an arclength along the vortex line. Next we label $\mathbf{s}' \equiv \mathrm{d}\mathbf{s}/\mathrm{d}\xi$ and $\mathbf{s}'' \equiv \mathrm{d}\mathbf{s}'/\mathrm{d}\xi$. In our context, $\mathbf{s}'$ is a tangent vector and $|\mathbf{s}''|$ is a local curvature $R^{-1}$ at a given point. The triad of vectors $\mathbf{s}'$, $\mathbf{s}''$, $\mathbf{s}' \times \mathbf{s}''$ are perpendicular to each other (**Figure 4**) and work as tangent, normal and binormal, respectively:

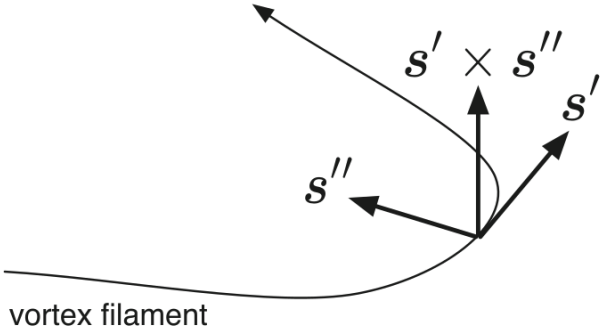


**Figure 4:** Schematic of the vortex filament and the triad vectors $\mathbf{s}'$, $\mathbf{s}''$, $\mathbf{s}' \times \mathbf{s}''$. Source: [6]

We suppose that the superfluid component is incompressible $\nabla \cdot \mathbf{v}_s = 0$ and the vorticity $\omega_s$ is localised and quantised $\omega_s(\mathbf{s}, t) = \nabla \times \mathbf{v}_s = \varkappa$. Combining these two properties and using a Fourier transformation, one obtains [8] the Biot-Savart law for the induced superfluid velocity in space:

$$\mathbf{v}_s(\mathbf{r}) = \frac{\varkappa}{4\pi}\int_{\mathscr{L}} \frac{(\mathbf{r}' - \mathbf{r}) \times \mathrm{d}\mathbf{r}'}{|\mathbf{r}' - \mathbf{r}|^3}, \qquad (3)$$

where the integral path $\mathscr{L}$ goes along the entire vortex filament. Now we define the *self-induced* velocity $\mathbf{v}_{\text{ind}}$, describing the motion which a vortex line induces onto itself ($\mathbf{r} = \mathbf{s}$) due to its own curvature.

However, this integral diverges as $\mathbf{r}' \to \mathbf{s}$ because the core structure of the quantized vortex was neglected. We avoid this divergence by splitting the integral (3) into two parts - direct neighbourhood of the point $\mathbf{s}$ (local part) and the rest $\mathscr{L}'$ (nonlocal part) $\mathbf{v}_{\text{ind}} = \mathbf{v}_{\text{ind,local}} + \mathbf{v}_{\text{ind,nonlocal}}$.

The Taylor expansion of the local part leads to a finite result:

$$\mathbf{v}_{\text{ind}} \approx \beta \mathbf{s}' \times \mathbf{s}'' + \frac{\varkappa}{4\pi}\int_{\mathscr{L}'} \frac{(\mathbf{r}' - \mathbf{s}) \times \mathrm{d}\mathbf{r}'}{|\mathbf{r}' - \mathbf{s}|^3}, \qquad (4)$$

where $\beta = (\varkappa/4\pi)\ln(R/a_0)$ and $\mathscr{L}'$ is the original vortex line without a close area of the studied vortex point. $R$ is a *local curvature* and often is calculated as $1/|\mathbf{s}''|$.

The local part of induced velocity is called as *Local Induction Approximation* (LIA).

### Vortex dynamics

To determine the equation of motion of $\mathbf{s}(t)$ we recognize the forces acting upon the line - the magnus force $\mathbf{f}_M$ and the drag force $\mathbf{f}_D$ (both are per unit length). Also we note that we expect no external sources of normal nor superfluid flow and all fllowing equations are derived assuming this premise.

The magnus force arises when a vortex line with circulation quantum $\varkappa$ rotates and moves in a flow:

$$\mathbf{f}_M = \rho_s \varkappa \mathbf{s}' \times (\dot{\mathbf{s}} - \mathbf{v}_{\text{ind}}), \tag{5}$$

where $\dot{\mathbf{s}} \equiv \mathrm{d}\mathbf{s}/\mathrm{d}t$ is the velocity of a particular point on a vortex line.

The drag force $\mathbf{f}_D$ arises from the *mutual friction*, the interaction between the normal component $\mathbf{v}_n$ and rotational superfluid component $\mathbf{v}_s$. This force was experimentally derived by Vinen and Hall [9]:

$$\mathbf{f}_D = \alpha(T)\rho_s \varkappa \mathbf{s}' \times [\mathbf{s}' \times \mathbf{v}_{\text{ind}}] \tag{6}$$

$$+\alpha'(T)\rho_s \varkappa \mathbf{s}' \times \mathbf{v}_{\text{ind}} \tag{7}$$

The temperature dependent dimensionless parameters $\alpha(T)$ and $\alpha'(T)$ are known from another experiments of Samuels and Donnelly [10].

Since the mass of vortex core is neglected, the two forces $\mathbf{f}_M$ and $\mathbf{f}_D$ sum up into zero: $\mathbf{f}_M + \mathbf{f}_D = \mathbf{0}$. Hence, solving for $\mathrm{d}\mathbf{s}/\mathrm{d}t$, we obtain the Schwarz's equation:

$$\dot{\mathbf{s}} = \mathbf{v}_{\text{ind}} - \alpha \mathbf{s}' \times \mathbf{v}_{\text{ind}} + \alpha' \mathbf{s}' \times [\mathbf{s}' \times \mathbf{v}_{\text{ind}}], \tag{8}$$

On the basis of Schwarz's equation (8), algorithms to numerically simulate vortex time evolution of an arbitrary configuration can be developed. Also, the vortex parametrisation $\mathbf{s}(\xi, t)$ and dynamics description provide the baseline of what we call as Vortex Filament model (VF).

## 3. Simulation

This part of work serves as a brief documentation for the `PyVort` codebase, a new platform to simulate quantum vortex rings. The code is written in well commented Python 3 and arranged in a modular structure. The aim is to highlight which module(s) are involved and how they work.

All pieces of the Python code can be found in `src/` and they are executable with `main.py`. The entire project is open-source and can be found as a public GitHub repository.

At present, we use infinite boundary conditions with no external flow sources. Therefore, only closed-loop self-induced vortices can be realized using simulation. However, the codebase is flexible and supports the future implementation of unclosed loops.

### Implementation of VF model

Superfluid vortex filament is represented by a series of mesh points (segments) distributed along the centerline of the filament (**Figure 5**). The motion of the whole VF is summed up by the motion of each mesh point.

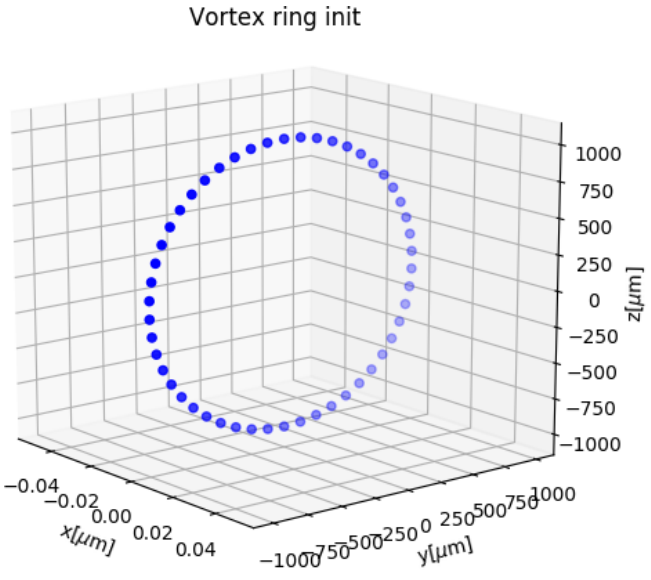


**Figure 5:** Visualisation of vortex ring segments right after their initialisation.

Each segment is defined by its coordinates $\mathbf{s}_i$ and its direct neighbour indices (previous $(i-1)$ and next $(i+1)$). This resolves in a directed digraph, which is used as an initial data structure.

Next we define the tangent vector $\mathbf{s}'$, normal vector $\mathbf{s}''$, and binormal vector $\mathbf{s}' \times \mathbf{s}''$ by taking numerical derivatives, achieved using Finite Differences method, a common approximative numerical method.

## Finite differences

Choosing a particular segment with position $\mathbf{s}_i$ , we define the distance to the particle in-front $\mathbf{s}_{i+1}$ as $l_i = |\mathbf{s}_i - \mathbf{s}_{i+1}|$ and the distance to the particle behind $\mathbf{s}_{i-1}$ as $l_{i-1} = |\mathbf{s}_{i-1} - \mathbf{s}_i|$ (**Figure 6**). Similarly, we can define the $l_{i+1}$ and $l_{i-2}$ for the farther segment lines.

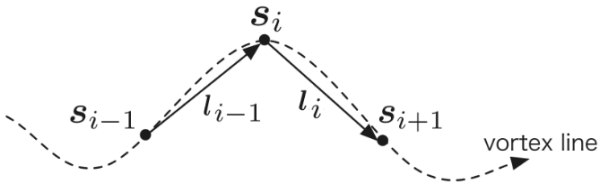


**Figure 6:** Depiction of $i$-th segment vector and its corresponding lengths. Source: [6]

We then approximate all the spatial derivatives $\mathbf{s}_i'$, $\mathbf{s}_i''$ by a fourth-order finite difference method (FD4), which can also account the varying distances along the VF. With this, the first and second derivatives can be obtained using Taylor's expansion based on coordinates of 2 closest neigbours (on each side):

$$\frac{\mathrm{d}^n \mathbf{s}_i}{\mathrm{d}\xi^n} \approx A_i \mathbf{s}_{i-2} + B_i \mathbf{s}_{i-1} + C_i \mathbf{s}_i + D_i \mathbf{s}_{i+1} + E_i \mathbf{s}_{i+2} \quad (9)$$

Calculation of coefficients $A, B, C, D, E$ in code is done directly using the analytical solution of *Vandermonde matrix* invertion. This invertion can be done also numerically, which is a more scalable way of implementation, however, this method often meets with problems when the Vandermonde matrix becomes singular.

## Biot-Savart discretisation

As showed earlier, the equation of motion for given segment is given directly by Schwarz's equation(8):

$$\frac{\mathrm{d}\mathbf{s}_i}{\mathrm{d}t} = \mathbf{v}_{\mathrm{ind}}^{(i)} + \mathbf{v}_{\mathrm{drive}}^{(i)} \quad (10)$$

The first difficulty in the VF model comes from the calculation of term $\mathbf{v}_{\mathrm{ind}}$. We suggested previously (4), this advection term can be split into the LIA part and a reduced Biot-Savart integral:

$$\mathbf{v}_{\mathrm{ind}}^{(i)} = \mathbf{v}_{\mathrm{LIA}}^{(i)} + \mathbf{v}_{\mathrm{BIOT}}^{(i)} \quad (11)$$

Using a geometrical approach for estimating the local curvature $R$, one can derive [11]:

$$\mathbf{v}_{\mathrm{ind}}^{(i)} = \frac{\varkappa}{4\pi}(\mathbf{s}_i' \times \mathbf{s}_i'') \ln\left(\frac{2\sqrt{l_{i-1} l_i}}{a}\right) \quad (12)$$

$$+ \frac{\varkappa}{4\pi} \int_{\mathscr{L}'} \frac{(\mathbf{r}' - \mathbf{s}_i) \times \mathrm{d}\mathbf{r}'}{|\mathbf{r}' - \mathbf{s}_i|^3}, \quad (13)$$

where $l_{i-1}$ and $l_i$ are the arc lengths of the curve between points $\mathbf{s}_{i-1}$ and $\mathbf{s}_i$ and between $\mathbf{s}_i$ and $\mathbf{s}_{i+1}$ respectively, as shown in **Figure 6**.

Filament discretisation allows the Biot-Savart integral to be rewritten into the sum of single-line contributions (**Figure 7**) between each $j$-th and $(j+1)$-th segment (except for the ones attached to the $i$-th point):

$$\mathbf{v}_{\mathrm{BIOT}}^{(i)} \approx \frac{\varkappa}{4\pi} \sum_{j \notin \{i-1, i\}} \frac{(R_j + R_{j+1})(\mathbf{R}_j \times \mathbf{R}_{j+1})}{R_j R_{j+1}(R_j R_{j+1} + \mathbf{R}_j \cdot \mathbf{R}_{j+1})}, \quad (14)$$

where $\mathbf{R}_j = \mathbf{s}_j - \mathbf{s}_i$ and $\mathbf{R}_{j+1} = \mathbf{s}_{j+1} - \mathbf{s}_i$ are the relative vectors from a given point.

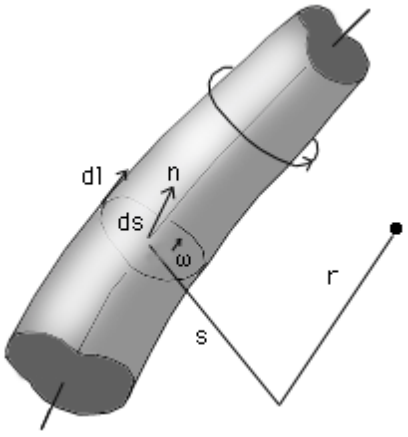


**Figure 7:** An infinitesimal contribution of a $j$-th segment line between two points $\mathbf{s}_j$ and $\mathbf{s}_{j+1}$ at a given point $\mathbf{r}$. Source: Internet

Note that the computational time of Biot-Savart part is proportional to $\mathscr{O}(N^2)$, while, if one uses just the LIA term, it is only $\mathscr{O}(N)$. Numerical simulations based on Biot-Savart are therefore far expensive and not practical.

One way to get around this difficulty is to update the LIA term (12). In code, when choosing BIOT = False in `config.py`, we neglect completely the non-local integral and keep just the local term, which is typically[11] adjusted within the log term:

$$\mathbf{v}_{\mathrm{LIA}}^{(i)} = \frac{\varkappa}{4\pi}(\mathbf{s}_i' \times \mathbf{s}_i'') \ln\left(\frac{2R_i}{a}\right), \qquad (15)$$

where $R_i$ is a filament length scale, which may be taken as a local curvature of $i-th$ segment: $R_i = 1/|\mathbf{s}_i''|$. Updated LIA is a convenient approximation and works very well for calculating the motion of a single vortex ring, as shown in Results part.

## State definition

In code, a single vortex ring object is represented as a `class` structure. This structure (*state*) is partially updated after each time step and is defined with following properties:

- shape - ring center coordinates $[x_c, y_c, z_c]$, radius $R$ and the direction of desired motion $\{x, y, z\}$ (three possible axis)
- velocity - the actual velocity magnitude of vortex ring center $|\mathbf{v}_c|$
- segments - an array of all mesh points, each one with following attributes:
  - coordinates - an array of segment coordinates $\mathbf{s}_i = [x_i, y_i, z_i]$
  - previous/next neighbour - array localisation indices of the *previous* $(i-1)$ and the *next* $(i+1)$ segment within the context of the directed vortex
  - tangent/curvature - a tangential and normal vectors $\mathbf{s}_i'$ and $\mathbf{s}_i''$
  - LIA velocity - a self-induced velocity $\mathbf{v}_{\mathrm{LIA}}^{(i)}$ driven by the local curvature.
  - BIOT velocity - a self-induced velocity driven by the farther segment lines of the vortex ring $\mathbf{v}_{\mathrm{BIOT}}^{(i)}$
  - Drive velocity - a velocity given by the mutual friction force $\mathbf{v}_{\mathrm{drive}}^{(i)}$
  - Full velocity - the sum of LIA velocity $\mathbf{v}_{\mathrm{LIA}}^{(i)}$, BIOT velocity $\mathbf{v}_{\mathrm{BIOT}}^{(i)}$ and the drive velocity $\mathbf{v}_{\mathrm{drive}}^{(i)}$, resulting in $\mathrm{d}\mathbf{s}_i/\mathrm{d}t$

The initialisation of the *state* in the very beginning of simulation includes the following steps:

1. an input from user: *center, radius, direction* and *resolution*. Resolution parameter is the initial desired distance $\delta$ between each two neighbouring segments.
2. The number of segments $N$ is calculated and the neigbour indices ($i-1$ and $i+1$ are assigned $i-1$ to the $i$-th segment) are gradually initialised
3. Calculation of all $\mathbf{s}'$ and $\mathbf{s}''$ using FD4
4. Calculation of all segment velocities using motion equations and their approximative forms
5. Calculation of the center velocity by taking the mean of all segments' full velocites in corresponding direction

## Time evolution

Time evolution (**Figure 8**) is based on an explicit iterative method: the fourth-order Runge-Kutta (RK4) scheme.

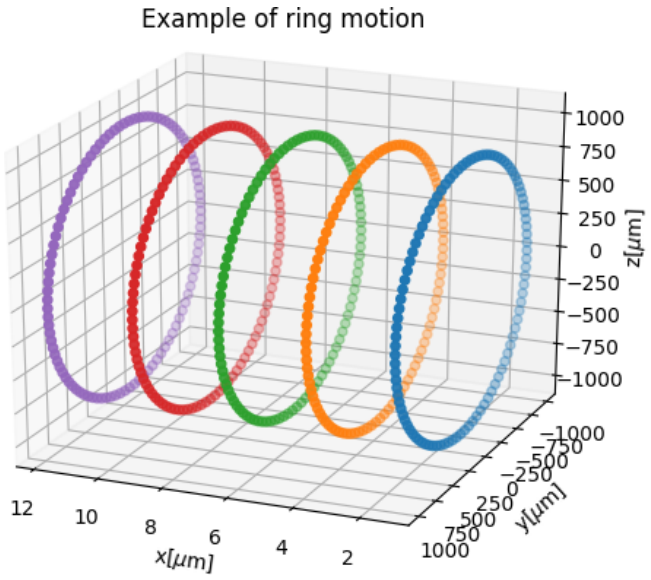


**Figure 8:** Example of a simulated moving vortex ring (to the left)

When we consider the Schwarz's equation $\mathrm{d}\mathbf{s}_i/\mathrm{d}t \equiv \mathbf{v}_{\mathrm{full}}^{(i)}$, the stepping algorithm is given as:

$$\mathbf{s}_i(t+\mathrm{d}t) = \mathbf{s}_i(t) + \frac{\mathrm{d}t}{6}(\mathbf{v}_1^{(i)} + 2\mathbf{v}_2^{(i)} + 2\mathbf{v}_3^{(i)} + \mathbf{v}_4^{(i)}), \qquad (16)$$

where $\mathrm{d}t$ is the time step and the velocities $\mathbf{v}_1^{(i)}, \mathbf{v}_2^{(i)}, \mathbf{v}_3^{(i)}, \mathbf{v}_4^{(i)}$ are the induced velocites of partial steps:

$$\mathbf{v}_1^{(i)} = \mathbf{v}_{\text{full}}^{(i)}(\mathbf{s}_i, t)\,, \tag{17}$$
$$\mathbf{v}_2^{(i)} = \mathbf{v}_{\text{full}}^{(i)}(\mathbf{s}_i + \mathbf{v}_1^{(i)}\mathrm{d}t/2, t + \mathrm{d}t/2)\,, \tag{18}$$
$$\mathbf{v}_3^{(i)} = \mathbf{v}_{\text{full}}^{(i)}(\mathbf{s}_i + \mathbf{v}_2^{(i)}\mathrm{d}t/2, t + \mathrm{d}t/2)\,, \tag{19}$$
$$\mathbf{v}_4^{(i)} = \mathbf{v}_{\text{full}}^{(i)}(\mathbf{s}_i + \mathbf{v}_3^{(i)}\mathrm{d}t, t + \mathrm{d}t) \tag{20}$$

Lower-order schemes such as basic Euler method is also implemented in code, however, not recommended to use due to its instability. The purpose is to use it only for stability testing.

The time step $\mathrm{d}t$ is chosen so that the vortex ring cannot move faster than a 1% of its size in a single step. Since the radius of vortex ring can vary in time, the time step $\mathrm{d}t$ is iteratively changing after each break of the above rule:

$$\mathrm{d}t \leftarrow \frac{0.01R}{|\mathbf{v}_c|} \tag{21}$$

## Re-segmentation of vortex

To obtain the most realistic simulation (catch effects on any length scale), the natural tendency would be to set the *resolution* parameter $\delta$ as low as possible. However, the CPU time cost rises rapidly as the number of segments $N$ increases, so there is need to find the best trade-off.

As the distance between neighbouring segments is compressed/enlarged with time due to physics and numerical inaccuracies, there is need to remove/add segments (*re-segment*) to conserve the vortex resolution $\delta$. The closeness (in terms of arclength) of neighbouring segments is therefore measured after each simulation time-step.

We used the simplest re-segmenting criteria - keeping an approximately *uniform distance* between the segments. To ensure this, two boundary conditions were implemented:

1. The segment $\mathbf{s}_{j+1}$ would be removed if:

$$|\mathbf{s}_{j+1} - \mathbf{s}_j| < \delta_{\min}\,, \tag{22}$$

where $\delta_{\min}$ is the minimal distance between the neighbouring segments (example in **Figure 9**). Also, the segment $\mathbf{s}_j$ would take place somewhere between $\mathbf{s}_{j-1}$ and $\mathbf{s}_{j+2}$ so that it will not damage the local curvature of vortex. This is ensured using spline interpolation of nearest 4 points (in our context $\mathbf{s}_{j-2}$, $\mathbf{s}_{j-1}$, $\mathbf{s}_{j+1}$, $\mathbf{s}_{j+2}$).

2. In a similar manner, we add a new segment $\mathbf{s}_{\text{new}}$ between $\mathbf{s}_j$ and $\mathbf{s}_{j+1}$ if:

$$|\mathbf{s}_{j+1} - \mathbf{s}_j| > \delta_{\max}\,, \tag{23}$$

where $\delta_{\max}$ is the maximal allowed distance between two neigbouring segments.

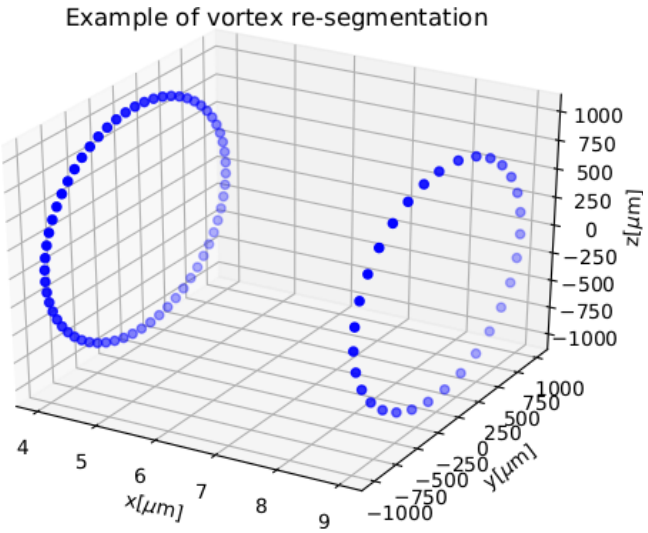


**Figure 9:** Example of simulated resegmentation process of vortex ring, after the 22 condition was violated

This method keeps all the distances along the vortex roughly around $\delta \in \langle \delta_{\min}, \delta_{\max} \rangle$ and also keeps the geometrical properties.

## Real-time tests

After each time step is done, a few tests are performed, to ensure that vortex itself is behaving according to our expectations. The tests are following:

- Length test - This test calculates the vortex circumference as $l = \sum_j |\mathbf{s}_j - \mathbf{s}_{j+1}|$ and compare it with the theoretical one $2\pi R$. If the deviation from the theoretical value is too high $> 1\%$, the segments are noisy and the process is killed. In case of deviation below $< -1\%$, there is clearly too less segments and resegmentation is called.

- Segmentation test - Here we check the value $l_j|\mathbf{s}_j - \mathbf{s}_{j+1}|$ for each $j$ and ask whether $l_j \in \langle \delta_{\min}, \delta_{\max} \rangle$. If not, resegmentation is called.
- Smallness test - If the vortex ring radius would decrease below $R < \delta_{\min}$, the ring is deleted from the simulation.

# 4. Results

All presented measurements and results were done in purpose of setup the `config` file. We performed a tests with physical motivation and tests focused on precision and stability. With presented findings, there should be ensured the correctness and stability of any further high-scale simulation.

## Frictionless test

In case of zero temperature $T = 0\,\mathrm{K}$, there should be no *normal component* in superfluid He-II and therefore also no mutual friction. In such case, velocity and energy of vortex ring is conserved due to the lack of energy dissipation processes.

We plotted in **Figure 10** the ring velocity $|\mathbf{v}_c|$ and energy evoluting in time, for the case of $T = 0$ and $T = 1.5\,\mathrm{K}$. We tested the mutual friction effect during 1000 time-steps (*epochs*) with varying d$t$.

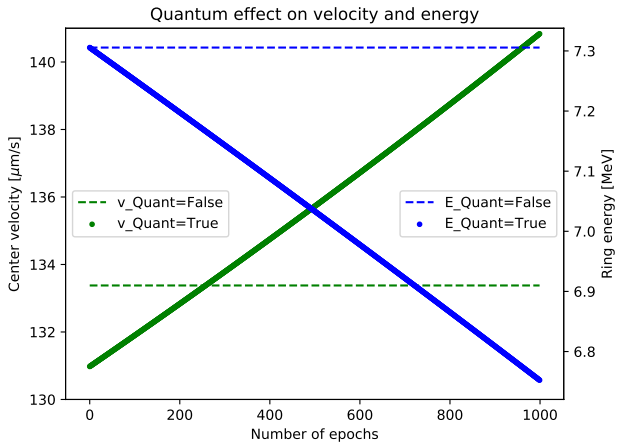


**Figure 10:** Dashed curves show the constant behaviour for temperature $T = 0$), full lines the dissipation process for $T = 1.5\mathrm{K}$. On `x` axis, we plot the number of epochs (time steps) the vortex ring ran over and on `y` axis the velocity of ring center $|\mathbf{v}_c|$.

As we see, at $T = 0\,\mathrm{K}$ the velocity and energy is conserved even after 1000 epochs of simulation. In case of $T > 0$, energy is falling down as expected, whereas the velocity is increasing. This increase is physically well-explained by the fact that the radius is decreasing with time (2), also due to mutual friction.

## Velocity precision test

Our first velocity test compares the various approaches at $T = 1.5\mathrm{K}$ how can the ring velocity $|\mathbf{v}_c|$ be calculated. Here, we recognize between four approaches, based on its theoretical motivation, computational complexity and precision:

- LIA (12): motivated, cheap, not precise
- LIA + BIOT (12 + 13): motivated, but expensive
- updated LIA (15): not well motivated, but cheap and precise
- Theoretical (scaled 2): motivated and precise

Of course, the theoretical velocity (2) is taken as a baseline that other velocites are compared with. All velocites were evoluted and measured during 5000 epochs (time steps), as we see in **Figure 11**.

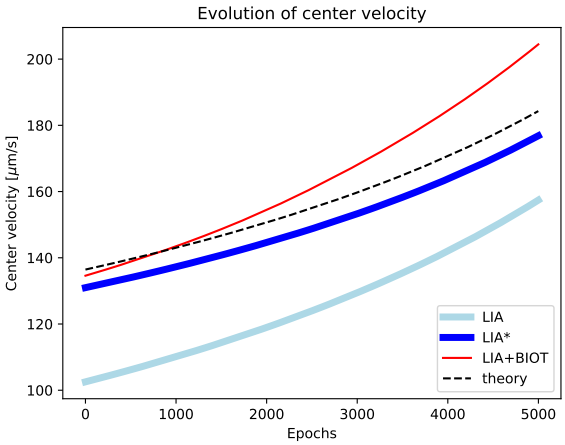


**Figure 11:** Comparison of all implemented velocity approaches (full lines) with the theoretical one (black dashed line). Velocities are always computed based on ring radius. On `x` axis, we plot the number of epochs (time steps) the vortex ring ran over and on `y` axis the velocity of ring center $|\mathbf{v}_c|$.

As we see, the (LIA + BIOT) velocity grows much faster than it should according to theory. Therefore, for all further measurements we used the updated LIA velocity (LIA*) due to its precision and speed.

## Velocity convergence test

Next, we investigated the magnitude of ring velocity $|\mathbf{v}_c|$ for various resolutions $\delta \in \langle 50, 200 \rangle \mu$m, immediately after initialisation and then after 100 epochs. Ring radius was set at $R = 1000\mu$m, so the number of discretisation points was given by $R$ and $\delta$ as $N \approx 2\pi R/\delta \in \langle 30, 120 \rangle$.

We see in **Figure 12** the expected convergent behaviour of both measured velocities in an area of good resolutions. Below $\delta < 100\mu$m, the velocites are enough convergent (corresponding to $\approx 60$ number of segments), which gives us the upper boundary.

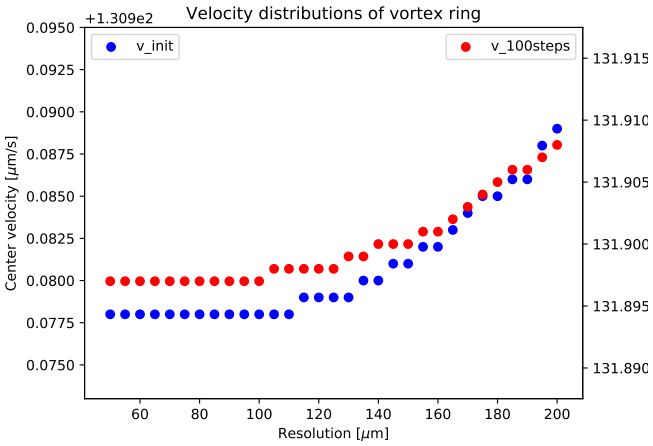


**Figure 12:** On x axis, we plot the resolutions $\delta$ and on y axis the velocity of ring center $|\mathbf{v}_c|$. Blue dots - initial ring velocities, Red dots - ring velocities after 100 epochs.

Even if it is intuitive that good resolutions lead to more stable velocities, the high number of vortex segments worsens the stability of simulation in time. Therefore, we propose also a lower boundary $\delta_{\text{min}}$, ensuring the stability, in a following test.

## Stability test

Stability of simulation was measured for three values of vortex radius $R \in \{500, 1000, 2000\}\mu$m using *Euler* and *RK4* stepping in various resolutions. In all cases, the stability is described by the number of reached epochs, which was determined by the violation of length condition - the vortex circumference cannot deviate more than 1% from the geometrical value $2\pi R$.

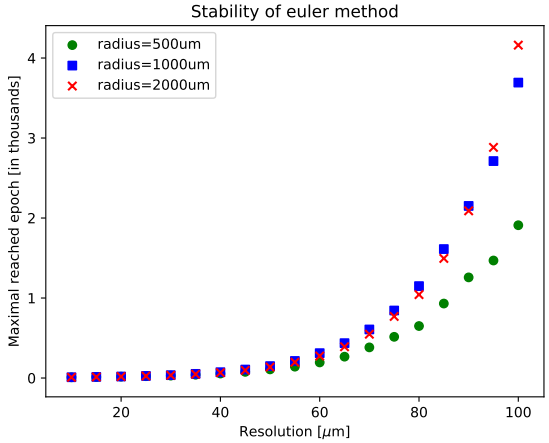


**Figure 13:** Maximal reached epoch (time step) with Euler method till the simulation was killed by violating the length condition, for various radii $R$ and resolutions $\delta$.

Euler method seems to be instable for whatever resolution (**Figure 13**). Therefore it is useful only for test purposes. We recommend to use a way more stable method RK4 in a real high-scale simulation:

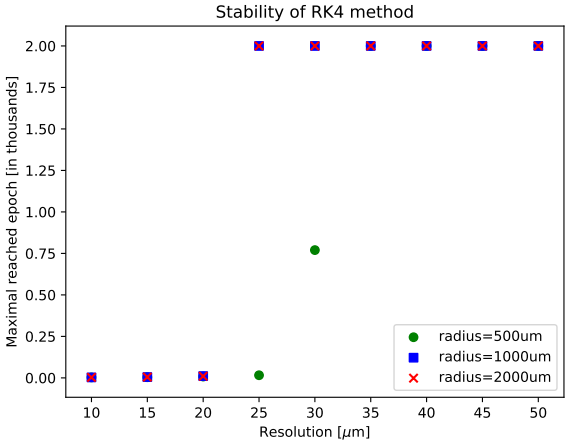


**Figure 14:** Maximal reached epoch (time step) with RK4 method till the simulation was killed by violating the length condition, for various radii $R$ and resolutions $\delta$. A threshold is set on 2000 epochs.

The plot in **Figure 14** suggests the minimal resolution to be at least $\delta > 30\mu$m. This boundary is still quite consevative, since the radius of vortex is decreasing in time (see Apendix part), so the resegmentation will happen. Resegmantation processes heavily help the simulation to be stable (deletes any forwarding numerical errors).

## 5. Conclusions

In this work, we proposed an effective numerical method to compute the time evolution of vortex ring in superfluid He-II. VF's performance was improved by neglegcting the Biot-Savart integral and updating the LIA calculation instead (15). Simulation well replicates the physical processes and performs sufficient stability when using testing vortex ring of radius $R \in \langle 500, 2000 \rangle \mu$m in resolution $\delta \in (60 - 100)\mu$m.

### Future improvements

To make `Pyvort` a full-fledged quantum vortex simulation, there should be implemented a few improvements.

- Performance - Recent numerical research presented [12] a new numerical method to compute the evolution of vortex filament. The method is based on a *tree algorithm* which considerably speeds up the calculation of Biot-Savart integrals - computational cost scales as $\mathcal{O}(N \log(N))$ rather than $N^2$. Certain physical quantities are heavily dependent on ring velocity, so the improvement of the simulation performance and precision would produce even more useful results.
- Interaction of vortices - If any lines of two vortices (or even the single one) become very close, the filaments can reconnect, changing the topology of the system. Many researchers experimentally reported this is happening and also found analogies with vortex dynamics in the Navier–Stokes equation. The VF model itself cannot describe the reconnection process because the vortex core structure is generally neglected. Hence, an artificial procedures must be introduced, for instance, when two vortices approach within a critical distance $\delta_{\text{min}}$, we will reconnect the vortices in the way based on energy minimalisation.
- High-order tests - Once the code of interacting VFs is developed, the analysis of all that data has to be improved. The simplest measurable quantity is the total lenght of all vortex lines. In a finite volume this would be $L = (1/V) \int \mathrm{d}\xi$ for the vortex line per unit volume. Of course, there can be defined more complicated metrics, measuring the isotropy of the vortex tangle. [11]

## Appendix

Here we briefly present a few visualisations of observed instabilities and physical processes.

### Instability in good resolution

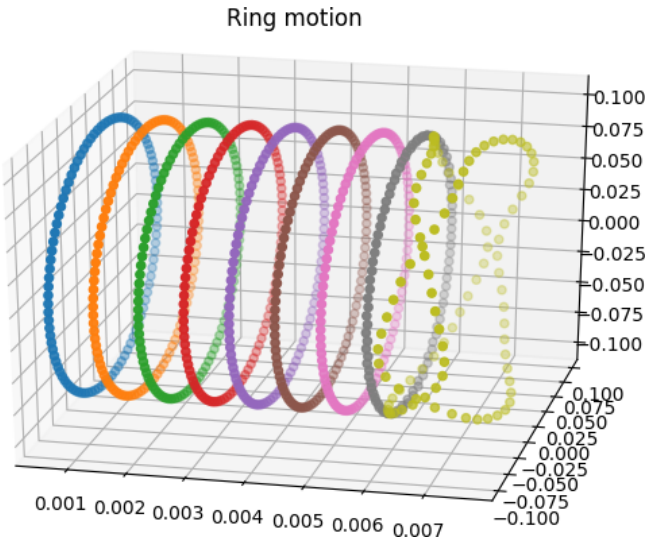


**Figure 15:** A graphical example of ring instability when Euler stepping was used with good resolution (small $\delta$). Ring is moving from the left side to the right with exponentially forwarding error.

### Noisy instability in worse resolution

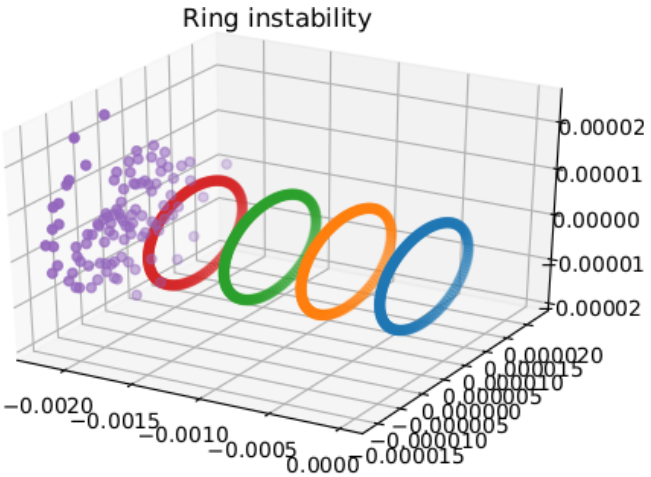


**Figure 16:** Another example of ring instability due to forwarded numerical error using Euler stepping. In this case, resolution $\delta$ is higher (worse resolution), so the instability itself is much more noisy.

### Mutual frtiction effects

Also, we present one of the bigger simulations (**Figure 17**) we tested. After we set the ring radius $R = 1000\mu$m, resolution $\delta = 60\mu$m (resulting in around $\approx 100$ mesh points), re-segmentation boundaries $\delta_{\text{min}} = \delta/2 = 30\mu$m, $\delta_{\text{max}} = 2\delta = 120\mu$m, the updated LIA calculation, RK4 stepping method and a temperature $T = 1.5$K, we obtained a stable simulation of vortex ring with a decreasing radius. More on particular physical processes can be found in [8].

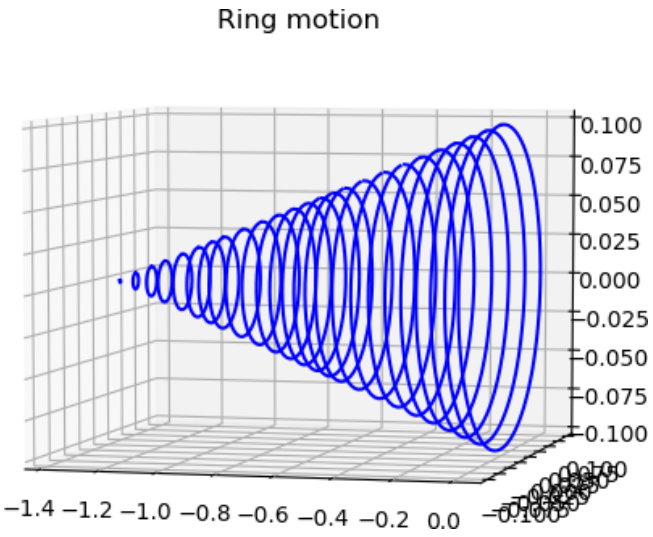


**Figure 17:** Here, a vortex ring is moving to the left, its energy is dissipating due to the mutual friction and therefore, the radius is decreasing in time. The ring re-segmented itself a few times (due to the violation of $\delta_{\text{min}}$ condition and died (the simulation was stopped) when the length error was too high.

## References


[1] LANDAU, L.D. *The theory of superfluidity of helium II* J. Phys. USSR, Vol. 11, 91 (1947)

[2] LANDAU, L.D. and LIFSHITZ, E.M. *Fluid Mechanics*. Second English Edition. Pergamon Books Ltd., 1987. ISBN 0-08-033933-6.

[3] ANDRONIKASHVILI, E.L. J. Phys. USSR, **10**, 201 (1946)

[4] BAHYL, J. *Measurement of quantum turbulence in superfluid He-4*. Student conference, FMPH UK, Bratislava (2016)

[5] FEYNMAN, R. *Application of quantum mechanics to liquid helium*.
Progress in Low Temp Phys, **1**, 17-53 (1957)

[6] TSUBOTA, M., FUJIMOTO, K., YUI S. *Numerical Studies of Quantum Turbulence*.
Journal of Low Temp Phys, 188 (2017)

[7] ROBERTS, P.H. Phys. Rev. A, 55, 1971

[8] BAGGALEY, A.W. *Superfluid vortices and Turbulence*. Quantized Vortex Dynamics and Superfluid Turbulence, Chap.1, Barenghi C.F. (2001)

[9] VINEN, W.F. and HALL, H.E. *The theory of mutual friction in uniformly rotating helium II*. Proc. Royal Soc. London 238 (1957) 204

[10] DONNELLY, R.J., BARENGHI, C.F *The Observed Properties of Liquid Helium at the Saturated Vapor Pressure*. American Ins. of Phys. and Chem. Soc. (1998)

[11] SAMUELS, D.C. *Vortex Filament Methods for Superfluids*. Quantized Vortex Dynamics and Superfluid Turbulence, Chap.9, Barenghi C.F. (2001)

[12] BAGGALEY, A.W., BARENGHI, C.F. *Tree Method for Quantum Vortex Dynamics*. Journal of Low Temp Phys, 166 (2012)