Università degli Studi di Verona

FACOLTÀ DI SCIENZE E INGEGNERIA
Corso di laurea triennale in Matematica Applicata

TESI DI LAUREA TRIENNALE

Numerical Simulations of Vortex Dynamics in Quantum Fluids

Candidato:
Alessio Zanella
Matricola VR489417

Relatore:
Marco Caliari

Anno Accademico 2024–2025

Contents

Chapter 1

Introduction

1.1 Introduction to superfluidity

The equations of classical fluid dynamics, such as the Navier-Stokes equations, have been known for more than a century. However, the complete understanding of their solutions in a turbulent regime remains one of the most important unsolved problems in classical physics.

Strangely, in the context of quantum mechanics, there exists quantum entities called bosons, as they respond to Bose-Einstein statistics, unlike Fermions, which follows the Fermi-Dirac statistics. At temperatures approaching absolute zero,these exhibit peculiar behaviors, in particular they can occupy the same quantum state, which fermions cannot, due to the Pauli exclusion principle. As a result, when a large number of bosons are cooled, a condensate is formed. This collective behavior gives rise to remarkable phenomena, such as superfluidity.

In this condensed state, the behavior of the superfluid is described by a single wave function, which carries both density and motion information. The dynamical evolution of this function is given by the Schrödinger equation, which, in our case, is called the Gross-Pitaevskii equation (GPE), a nonlinear form of the above equation.

Of course, since it is still a fluid, vorticity phenomena can occur, with the latter evolving by moving and recombining, thus establishing the relationship with classical fluid dynamics. In this framework, numerical simulations play a fundamental role, as the GPE is difficult to solve analytically, especially in the presence of vortices and complex phenomena such as the evolution and recombination of the latter.

The aim of this thesis is to study in detail an efficient numerical method for the resolution of GPE in the presence of vortices, both from the point of view of numerical analysis and by performing some experiments.

1.2 The mathematical context

In this section we introduce the mathematical model for a superfluid and seek an analytical approximation for the density ρ.

We define a *two-dimensional vortex* as a vortex filament embedded in three-dimensional space that possesses cylindrical symmetry. Assuming its axis lies along the x_3-direction, the configuration depends only on the coordinates in the orthogonal plane, so that variations are observed exclusively in the (x_1, x_2)-plane.

The governing equation for a weakly interacting Bose-Einstein condendate is the *Gross-Pitaevskii equation* (GPE for short)

$$\mathrm{i}\hbar\frac{\partial\psi}{\partial t} = -\frac{\hbar^2}{2m}\Delta\psi + V_0\left|\psi\right|^2\psi - E_0\psi$$

where $\psi = \psi(\mathbf{r}, t)$ is the complex macroscopic wavefunction for a condensate of N bosons of mass m at position $\mathbf{r}$ and time t. The constant $\hbar$ is the reduced Planck's constant, V_0 is the strength of the repulsive interactions between bosons, and E_0 is the chemical potential of the system (the energy required to add a boson to the condensate).

The adimensional form of the GPE, which derivation can be found in [4], is

$$\frac{\partial\psi}{\partial t} = \frac{\mathrm{i}}{2}\Delta\psi + \frac{\mathrm{i}}{2}(1 - |\psi|^2)\psi \tag{1.1}$$

Let us introduce the Madelung transformation, writing the wavefunction in polar form

$$\psi(\mathbf{r}, t) = \rho^{1/2}\exp(\mathrm{i}S) \tag{1.2}$$

where $\rho = \rho(\mathbf{r}, t) = |\psi(\mathbf{r}, t)|^2$ is the density and $S = S(\mathbf{r}, t)$ is the phase of ψ. The vector field $\mathbf{u} = \nabla S$ can be interpreted as the superfluid velocity field.

Substituting (1.2) into (1.1) and separating real and imaginary parts, we obtain the following system of equations, written in Einstein notation

$$\begin{aligned}
&\frac{\partial\rho}{\partial t} + \frac{\partial\rho u_j}{\partial x_j} = 0\\
&\rho\left(\frac{\partial u_i}{\partial t} + u_j\frac{\partial u_i}{\partial x_j}\right) = -\frac{\partial p}{\partial x_i} + \frac{\partial\tau_{ij}}{\partial x_j}, \quad i = 1, 2, 3
\end{aligned}$$

where

$$p = \frac{\rho^2}{4} \quad \text{and} \quad \tau_{ij} = \frac{1}{4}\rho\frac{\partial^2\log\rho}{\partial x_i x_j^2}$$

are the pressure and the quantum stress tensor, respectively.

These equations resemble the classical Navier-Stokes equations for a barothopic, inviscid fluid, where the volume forces are conservative (in our case they are constantly zero). The main difference is in the stress term, since a quantum fluid is inviscid.

These similarities suggest, in some ways, that the study of quantum vortex dynamics, in particular vortex reconnections, could provide some insights into the classical case.

1.3 Properties of the solution

The GPE has several conserved quantities, which can be derived from its Hamiltonian structure. The most important ones are the mass, the momentum and the energy.

We refer to $\bar{\psi}$ as the complex conjugate of ψ.

1.3.1 Mass

The mass in a bounded domain Ω is defined as

$$m_\Omega(t) = \int_\Omega |\psi|^2 \, \mathrm{d}V \tag{1.3}$$

The derivative with respect to time, which derivation can be found in [3], is

$$m'_\Omega(t) = \Im \left[\int_{\partial\Omega} \psi \left(\nabla\bar{\psi} \cdot \hat{\mathbf{n}} \right) \mathrm{d}S \right]$$

We notice that if $\nabla\psi \cdot \hat{\mathbf{n}} = 0$ on the boundary $\partial\Omega$, then $m'_\Omega(t) = 0$, so that the mass is conserved.

1.3.2 Energy

The energy in a bounded domain Ω is defined as

$$E_\Omega(t) = \int_\Omega \left(\frac{1}{2} |\nabla\psi|^2 + \frac{1}{4}(1 - |\psi|^2)^2 \right) \mathrm{d}V \tag{1.4}$$

The derivative with respect to time, which derivation can also be found in [3], is

$$E'_\Omega(t) = \Im \left[\frac{1}{2} \int_{\partial\Omega} \left(\nabla^2\bar{\psi} + (1 - |\psi|^2)\bar{\psi} \right) \nabla\psi \cdot \hat{\mathbf{n}} \, \mathrm{d}S \right]$$

Again, if $\nabla\psi \cdot \hat{\mathbf{n}} = 0$ on the boundary $\partial\Omega$, then $E'_\Omega(t) = 0$.

1.4 Approximation of steady-state vortex

In a two-dimensional domain we set

$$\psi(x, y) = \rho\left(\sqrt{x^2 + y^2}\right) \exp\left(\mathrm{i}\,\theta(x, y)\right),$$

where $\rho(r)$ is a function to be determined. Requiring time-independence (steady-state solution) leads to the following ordinary differential equation for ρ:

$$\rho''(r) + \frac{\rho'(r)}{r} + \frac{(\rho'(r))^2}{2\rho(r)} - \frac{2\,\rho(r)}{r^2} + 2\big(1 - \rho(r)\big)\rho(r) = 0, \tag{1.5}$$

with boundary conditions $\rho(0) = 0$ and $\rho(\infty) = 1$. There are several approaches: one could solve the boundary-value problem directly on a truncated domain and interpolate the solution. In many applications, however, it is convenient to approximate $\rho(r)$ using a Padé approximant of arbitrary order. The Padé approximant used below is of fourth order; its coefficients are reported in [3].

Chapter 2

Numerical Methods

2.1 Space discretization

In this section we describe the spatial discretization methods used to simulate the GPE.

2.1.1 Finite differences on uniform grid

Let us discretize the interval $[-L, L]$ with m equispaced points, so that

$$x_1 = -L, \quad x_m = L, \quad x_j = -L + (j-1)h, \quad 1 \le j \le m,$$

where the mesh size is given by $h = \frac{2L}{m-1}$. Following [1], the second derivative along one spatial direction is approximated by the standard finite-difference matrix

$$D_2^x = \frac{1}{h^2} \begin{bmatrix} -2 & 2 & 0 & 0 & \cdots & 0 & 0 \\ 1 & -2 & 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & -2 & 1 & \cdots & 0 & 0 \\ 0 & 0 & 1 & -2 & \ddots & 0 & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 1 & -2 & 1 \\ 0 & 0 & 0 & 0 & \cdots & 2 & -2 \end{bmatrix}$$

This corresponds to the usual second-order central difference scheme with Neumann boundary conditions.

2.1.2 Finite differences on non uniform grid

Although the above numerical scheme is simple and easy to implement, in our case it is not the most suitable one, since, as we will see in the next chapter, it requires a very

large number of points to obtain physically reliable results, especially when integrating over long time intervals.

Given that the vortex density exhibits stronger variations near its core, while remaining almost constant towards the boundary of the domain, it would be preferable to employ a grid that clusters points around the vortex center(s). In this way, higher resolution is achieved in the regions where sharp variations occur.

Consider the interval $[a, b]$ and let $\mathbf{x} = [x_1, \ldots, x_m]^T$, with

$$x_1 = a, \quad x_m = b, \quad x_j = x_{j-1} + h_j,$$

where the grid spacings satisfy $h_j = (1 + \delta)h_{j-1}$ for some fixed parameter δ. This leads to the recurrence

$$\begin{aligned} h_j &= (1+\delta)h_{j-1} \\ &= (1+\delta)^2 h_{j-2} \\ &\vdots \\ &= (1+\delta)^{j-1} h_1. \end{aligned}$$

The constraint is that the sum of all step sizes must cover the entire interval length, which gives

$$\sum_{j=1}^{m-1} h_j = h_1 \sum_{j=1}^{m-1} (1+\delta)^{j-1} = b - a.$$

There are three natural ways to proceed, each consisting of fixing two among the parameters δ, m, and h_1, and then computing the remaining one. In what follows, we present two cases: computing m and computing h_1.

First of all, we observe that the sum is a geometric series with ratio $1 + \delta$, whose closed-form expression is known analytically:

$$\sum_{j=1}^{m-1} (1+\delta)^{j-1} = \frac{(1+\delta)^{m-1} - 1}{\delta}.$$

To compute m we proceed as follows:

$$
\begin{aligned}
h_1 \frac{(1+\delta)^{m-1}-1}{\delta} = b-a \iff (1+\delta)^{m-1} &= \frac{\delta(b-a)}{h_1}+1 \\
\iff (m-1)\log(1+\delta) &= \log\left(\frac{\delta(b-a)}{h_1}+1\right) \\
\iff m &= \left\lceil \frac{\log\left(\frac{\delta(b-a)}{h_1}+1\right)}{\log(1+\delta)}+1 \right\rceil .
\end{aligned}
$$

Since m must be an integer, we take the ceiling in the final expression.
The expression for h_1, after some trivial algebra, is

$$
h_1 = \frac{\delta(b-a)}{1-(1+\delta)^{m-1}}
$$

To construct the one-dimensional discretization, it is important to note that, in the case of non-uniform finite differences, the local error can be shown to be proportional to $O(h_j^2)$ provided that the difference between h_j and h_{j-1} is of order $O(\max\{h_j^2, h_{j-1}^2\})$, therefore, care must be taken in constructing the grid using the above formula, so as to keep the difference between two consecutive steps as smooth as possible. Otherwise, the error would not even be proportional to h_j, which in the worst case would lead to an inconsistent scheme and consequently to significant errors in the time integration.

With this in mind, we can construct the discretization along one direction by applying the procedure described above to the interval $[0, x_c]$, where x_c denotes the center of the vortex, clustering the points towards x_c. We then extend the grid from x_c to the boundary of the domain, using the same values of h and δ, so that the spacing over the interval $[0, 2x_c]$ remains consistent with the previous construction. The grid can be further extended up to the domain boundary within a small tolerance, since this procedure does not necessarily guarantee that the endpoints are reached exactly. However, this is not problematic, as small discrepancies at the domain boundaries do not significantly affect the evolution of the dynamics, being far from the regions of interest.

Once we have the step vector $\mathbf{h}$ constructed, we can discretize the second derivative along one direction using the tridiagonal matrix

$$
D_2^x = \begin{bmatrix}
-\frac{2}{h_1^2} & \frac{2}{h_1^2} & 0 & \cdots & 0 \\
\frac{2}{h_2(h_1+h_2)} & -\frac{2}{h_1 h_2} & \frac{2}{h_1(h_1+h_2)} & \cdots & 0 \\
0 & \frac{2}{h_3(h_2+h_3)} & -\frac{2}{h_2 h_3} & \ddots & 0 \\
\vdots & \ddots & \ddots & \ddots & \frac{2}{h_{m-1}(h_{m-2}+h_{m-1})} \\
0 & \cdots & 0 & \frac{2}{h_{m-1}^2} & -\frac{2}{h_{m-1}^2}
\end{bmatrix}
$$

2.1.3 Laplacian discretization

Now, in both cases (uniforn or not), the Laplacian operator in two dimension is approximated by

$$\Delta \approx I^y \otimes D_2^X + D_2^Y \otimes I^x \tag{2.1}$$

where $\otimes$ denotes the Kronecker product (see A.1 for further details), and I^* denotes the identity matrix of the dimension equal to the number of points in each direction.

2.2 Time integration

2.2.1 Strang splitting method

Consider the Cauhcy problem

$$\begin{cases} \frac{\partial u}{\partial t} = (A + B)u \\ u(0) = u_0 \end{cases} \tag{2.2}$$

where A and B are two differential operators.
The analytical solution for 2.2 is

$$u(t) = e^{(A+B)t}u_0$$

Proposition 2.1. *If A and B commute, then* $\exp(A + B) = \exp(A)\exp(B)$

Proof.

$$\begin{aligned} \exp(A + B) &= \sum_{n=0}^{\infty} \frac{(A + B)^n}{n!} \\ &= \sum_{n=0}^{\infty} \sum_{k=0}^{n} \binom{n}{k} \frac{A^k B^{n-k}}{n!} \\ &= \sum_{n=0}^{\infty} \sum_{k=0}^{n} \frac{A^k}{k!} \frac{B^{n-k}}{(n-k)!} \qquad \ell = n - k \\ &= \sum_{k=0}^{\infty} \frac{A^k}{k!} \sum_{\ell=0}^{\infty} \frac{B^\ell}{\ell!} \\ &= \exp(A)\exp(B) \end{aligned}$$

■

So, for the previous proposition, if A and B commute, then $u(t) = e^{tA}e^{tB}u_0 \implies u(t + \tau) = e^{\tau A}e^{\tau B}u(t)$.

This is equivalent to solve separately two initial value problems:

$$\begin{cases} \frac{\partial u}{\partial t} = Au \\ u(0) = u_0 \end{cases}$$

and

$$\begin{cases} \frac{\partial u}{\partial t} = Bu \\ u(0) = u_0 \end{cases}$$

This gives rise to a numerical scheme:

$$\begin{aligned}\bar{u}_{n+1} &= \exp(\tau A)u_n \\ u_{n+1} &= \exp(\tau B)\bar{u}_{n+1}\end{aligned}$$

This approach is known as *Lie splitting*. It is exact when the operators A and B commute, otherwise it achieves first-order accuracy with respect to the time step τ.

Other methods can be constructed by considering alternative ways of splitting the operators. One widely used approach is the *Strang time splitting*, which improves the accuracy by symmetrically composing the substeps. Specifically, it provides the numerical scheme

$$\begin{aligned}\bar{u}_{n+1} &= \exp\left(\frac{\tau}{2}A\right)u_n \\ \tilde{u}_{n+1} &= \exp(\tau B)\bar{u}_{n+1} \\ u_{n+1} &= \exp\left(\frac{\tau}{2}A\right)\tilde{u}_{n+1}\end{aligned} \tag{2.3}$$

where $u_n \approx u(t_n)$.

Proposition 2.2 (Error of Strang splitting)**.** *The Strang time-splitting method is* exact *if the operators A and B commute, and it is of order* 2 *with respect to the time step τ otherwise.*

Remark 2.3. In the special case where A and B commute, the Strang splitting becomes exact. Indeed, we have

$$e^{(A+B)\tau} = e^{\tau A}e^{\tau B} = e^{\frac{\tau}{2}A}e^{\frac{\tau}{2}A}e^{\tau B} = e^{\frac{\tau}{2}A}e^{\tau B}e^{\frac{\tau}{2}A},$$

which coincides with the splitting formulation.

Order 2. Let us now consider the general case where A and B do not commute. The exact solution after one time step is given by

$$u(t+\tau) = e^{(A+B)\tau}u(t),$$

whereas the Strang approximation reads

$$u(t+\tau) \approx e^{\frac{\tau}{2}A}e^{\tau B}e^{\frac{\tau}{2}A}u(t).$$

The difference between the two is entirely contained in the discrepancy between the operators

$$e^{(A+B)\tau} \quad \text{and} \quad e^{\frac{\tau}{2}A}e^{\tau B}e^{\frac{\tau}{2}A}.$$

To quantify this, we expand the Strang operator in a Taylor series:

$$\left(I + \tfrac{\tau}{2}A + \tfrac{\tau^2}{8}A^2 + O(\tau^3)\right)\left(I + \tau B + \tfrac{\tau^2}{2}B^2 + O(\tau^3)\right)\left(I + \tfrac{\tau}{2}A + \tfrac{\tau^2}{8}A^2 + O(\tau^3)\right).$$

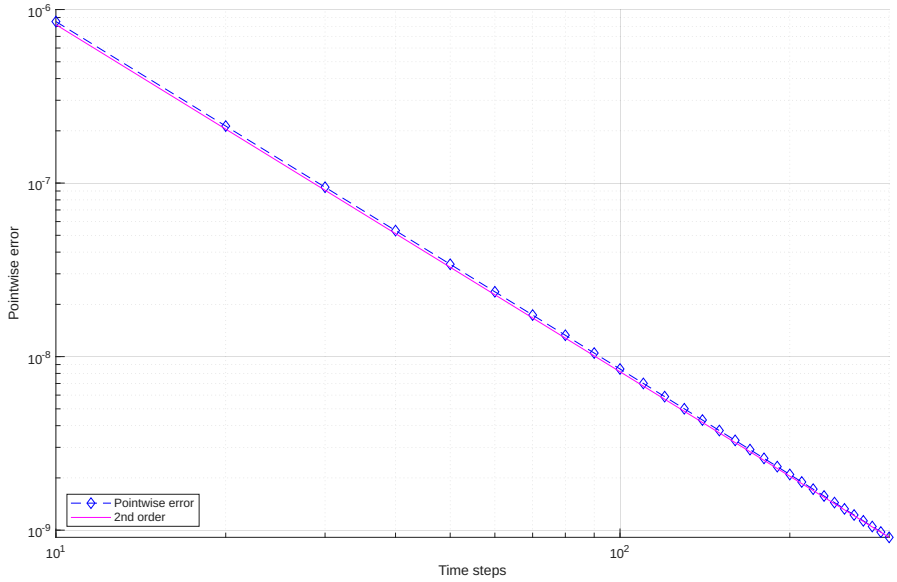

Figure 2.1: Convergence order

Multiplying and collecting terms up to order τ^2, we obtain

$$I + \tau(A + B) + \frac{\tau^2}{2}(A + B)^2 + O(\tau^3).$$

On the other hand, the Taylor expansion of the exact propagator is

$$e^{(A+B)\tau} = I + \tau(A + B) + \frac{\tau^2}{2}(A + B)^2 + \frac{\tau^3}{3}(A + B)^3 + O(\tau^4).$$

Subtracting the two expressions shows that the leading term of the local error is of order τ^3. Hence, the method is (at least) second-order accurate.

To prove that it is *exactly* 2, we need to expand up to the third order and subtract. After some computation, one finds that the local error is

$$\tau^3 \left(\frac{1}{12} [B, [B, A]] - \frac{1}{24} [A, [A, B]] \right) + O(\tau^4) = O(\tau^3)$$

Since, as $[A, B] \neq 0$, the cubic term does not cancel, hence the method has a local error proportional to τ^3, and is therefore of order 2. ■

2.2.2 Method applied to GPE

In the case of GPE, we have $A = \frac{\mathrm{i}}{2}\Delta$ and $B = \frac{\mathrm{i}}{2}\left(1 - |u|^2\right)$, this results in the numerical scheme

$$\bar{\mathbf{u}}_{n+1} = \exp\left(\frac{\tau \mathrm{i}}{2}\Delta\right)\mathbf{u}_n$$
$$\tilde{\mathbf{u}}_{n+1} = \exp\left(\frac{\tau \mathrm{i}}{2}\left(1 - |\bar{\mathbf{u}}_{n+1}|^2\right)\right)\bar{\mathbf{u}}_{n+1}$$
$$\mathbf{u}_{n+1} = \exp\left(\frac{\tau \mathrm{i}}{2}\Delta\right)\tilde{\mathbf{u}}_{n+1}$$

For B we directly approximate the solution. The treatment for the first is more delicate: since in fact the Laplacian operator is approximated by the formula 2.1, we need to compute the matrix-vector product

$$\exp\left(\frac{\tau}{2}\left(I^y \otimes D_2^x + D_2^y \otimes I^x\right)\right)\mathbf{u}_n$$

which turns out to be a full matrix-vector product of large dimension, leading to a huge memory requirement and, consequently, to an unreasonable computational time. Our goal would be avoiding the construction of any Kroneker product; instead we directly compute the action of the latter on the vector $\mathbf{u}$. First we notice that $I^y \otimes D_2^x$ and $D_2^y \otimes I^x$ commute, as trivial consequence of the formula A.3. This implies that

$$\exp\left(I^y \otimes D_2^x + D_2^y \otimes I^x\right) = \exp\left(I^y \otimes D_2^x\right)\exp\left(D_2^y \otimes I^x\right)$$

Since the exponential function is analytical, using A.8, we have

$$\exp\left(I^y \otimes D_2^x\right) = I^y \otimes \exp\left(D_2^x\right)$$

and the same for the other. Using again the formula A.3, we obtain

$$\exp\left(I^y \otimes D_2^x + D_2^y \otimes I^x\right)\mathbf{u} = \left(\exp\left(D_2^y\right) \otimes \exp\left(D_2^x\right)\right)\mathbf{u}$$

This is way better than directly compute the full exponential of Kroneker product, but it still requires the computation of a large full matrix. This can be avoided, since we can take into account the matrix $U = (u_{i,j}) \in \mathbb{C}^{m_x \times m_y}$, with the relation $U_{i,j} = \mathbf{u}_{i+m_x(j-1)}$, also denoted with $\mathrm{vec}(U) = \mathbf{u}$. Let now $E_x = \exp\left(\tau D_2^x\right)$ and $E_y = \exp\left(\tau D_2^y\right)$. Thanks to A.4, we have that

$$\left(\exp\left(D_2^y\right) \otimes \exp\left(D_2^x\right)\right)\mathbf{u} = \mathrm{vec}\left(E_x\, U\, E_y^T\right)$$

This result is way better than the previous one, as now we only have to compute two matrix-matrix products.

Chapter 3

Numerical experiments

3.1 Stationary vortex preservation over long time integration

The first numerical experiment we consider concerns the preservation of the steady-state solution of 1.1. Since this solution is stationary, a natural way to assess the quality of our numerical method is to examine how much the approximate solution deviates from the initial one over long integration times.

In all the following experiments, we approximate the density ρ by a 4-th order Padé approximation of the equation 1.5

3.1.1 Uniform grid

We now present some results from the performed experiments. The parameters used in the following simulations are: $m_x = m_y = 347$, $n_t = 450$, $t^* = 20$, $L_x = L_y = 30$.

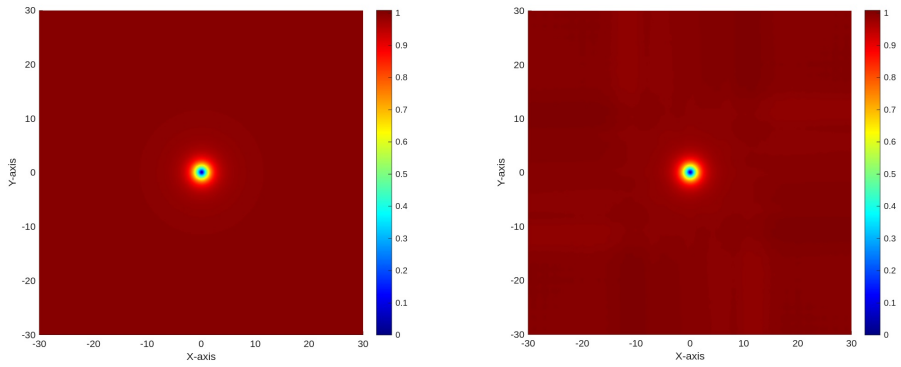

Figure 3.1: $|\psi_0|$ (left) and $|\psi(20)|$ (right), finite differences on a uniform grid.

In this figure, we can observe that the numerical solution integrated up to $t = 20$ shows slight variations in the norm outside the vortex core. To highlight this difference more clearly, we employ an alternative colormap with a steeper gradient for values close to 1:

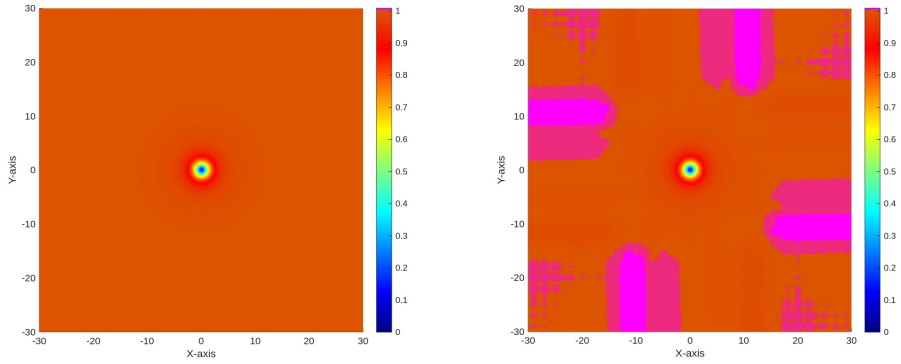

Figure 3.2: Initial solution (left), solution at time $t = 20$ (right)

We observe that the variations in the norm are localized near the limit of the domain, outside the vortex core. To highlight this, Figure 3.3 shows the truncated domain $[-5, 5]$, emphasizing the vortex center.

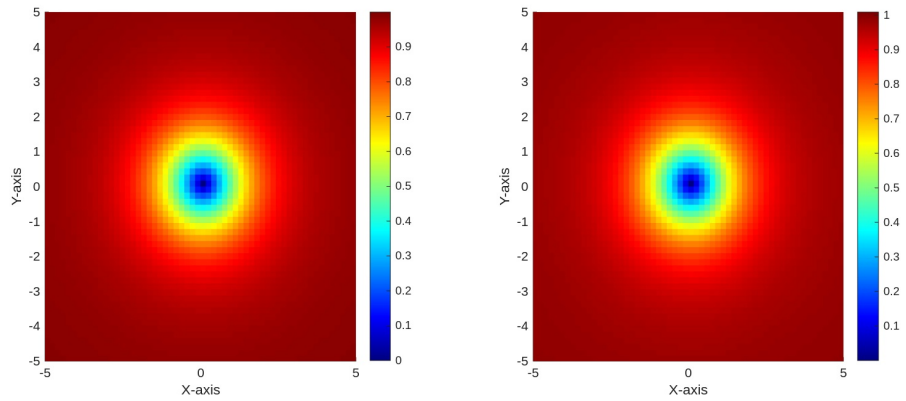

Figure 3.3: Zoomed domain

From this figure, we can see that no visually appreciable variations occur near the vortex core.

3.1.2 Non-uniform grid

We now consider a non-uniformly spaced grid as described in 2.1.2, with $\delta = h_1 = 1/50$. This results in a grid composed of 347 points per direction.

We directly report the comparison between the uniform and non-uniform grids.

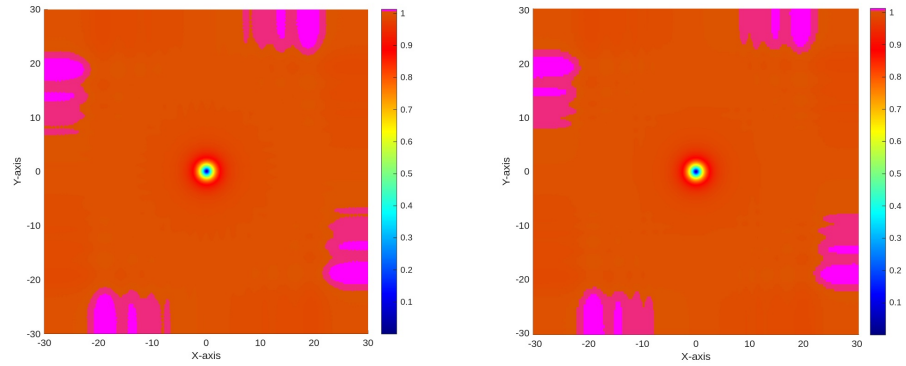

Figure 3.4: Comparison between uniform (left) and non-uniform (right) grids.

We can observe that the solution on the two grids is very similar, showing some slight differences near the boundary of the domain.

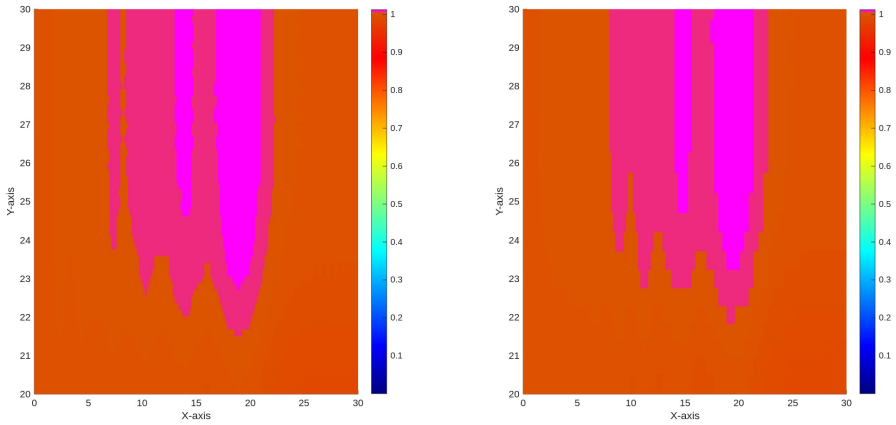

Figure 3.5: Zoom near the boundary of the domain.

In this zoomed view near the boudary of the domain, we can see that the solution on the non-uniform grid appears to be sligtly less accurate, due to the coarser discratization in that region. However, the differences are minimal, and does not significantly affect the overall quality of the solution.

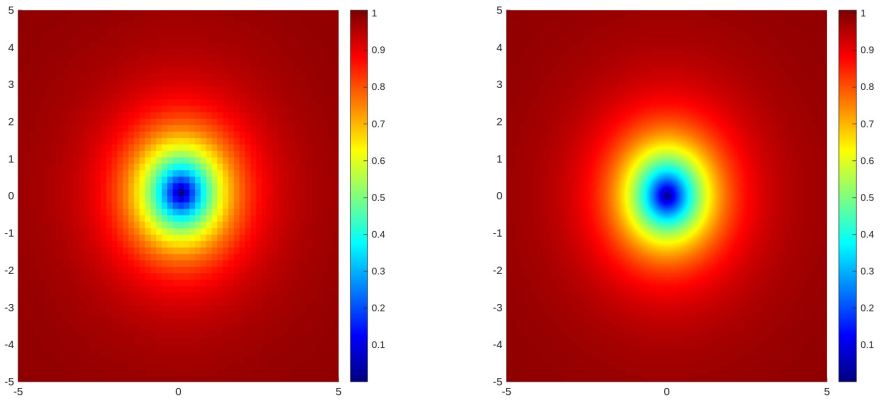

Figure 3.6: Zoom near the center of the domain.

Zooming into the center of the domain, again, no significant differences are observed. However, the refinement of the grid towards the origin in the non-uniform construction is clearly visible.

3.2 Simulations of two straight vortices

In this section, we simulate the dynamics of two interacting straight vortices. We perform experiments by imposing both concordant and opposite initial phases, and compare the different spatial discretization methods introduced so far.

3.2.1 Simulation of marching vortices

We now proceed to perform a set of simulations involving two straight vortices, centered respectively at (x_{c_1}, y_{c_1}) and (x_{c_2}, y_{c_2}). In this case we impose counter-rotating phases, meaning that, given the initial condition

$$\psi_0 = \sqrt{\rho_1 \rho_2}\, \exp\big(\mathrm{i}(S_1 + S_2)\big),$$

we have $\rho_1 = \rho(x - x_{c_1}, y - y_{c_1})$ and $S_1 = \mathrm{atan2}(y - y_{c_1}, x - x_{c_1})$, while $\rho_2 = \rho(x - x_{c_2}, y - y_{c_2})$ and $S_2 = -\mathrm{atan2}(y - y_{c_2}, x - x_{c_2})$.

All the following experiments have been carried out under the assumption of symmetry with respect to the y-axis, so that $x_c = x_{c_1} = -x_{c_2}$.

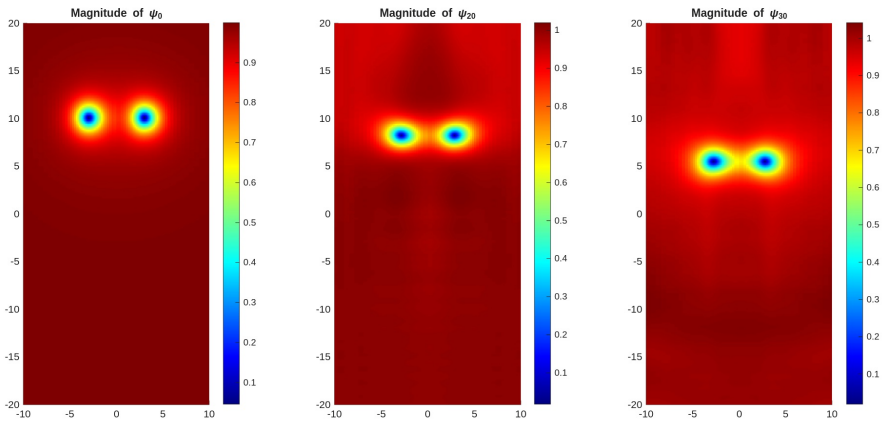

Figure 3.7: Comparison between ψ_0 (left), ψ_{10} (center), and ψ_{30} (right), for $x_c = 3$.

With this configuration, the vortices appear to march steadily towards the lateral boundaries of the computational domain. This drift is a direct consequence of the imposed counter-rotation, which breaks the otherwise stationary balance. In the following figure we can appreciate how the translation speed depends strongly on the initial separation between the vortex centers.

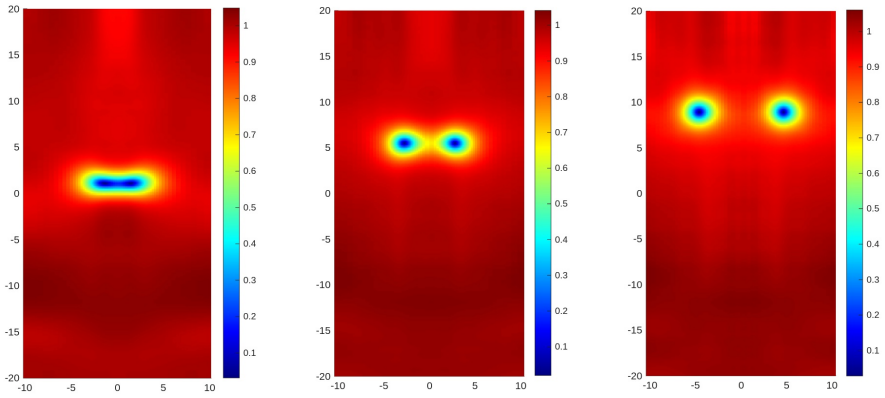

Figure 3.8: Solutions at time $t = 30$ for $x_c = 2, 3$, and 5.

All simulations reported here were performed on a nonuniform grid along the x-direction, reported in 3.12 with the spacing defined in 2.1.2, while in the y-direction equispaced nodes were employed. This choice of discretization allows one to concentrate grid resolution where the vortex dynamics are more intense, without excessively increasing the overall computational cost.

3.3 Simulation of two rotating vortex

We now employ the initial condition given by

$$\psi_0 = \sqrt{\rho_1 \rho_2}\exp\left(\mathrm{i}(S_1 - S_2)\right),$$

with $\rho_1 = \rho(x - x_{c_1}, y - y_{c_1})$ and $S_1 = \mathrm{atan2}(y - y_{c_1}, x - x_{c_1})$, while $\rho_2 = \rho(x - x_{c_2}, y - y_{c_2})$ and $S_2 = -\mathrm{atan2}(y - y_{c_2}, x - x_{c_2})$.

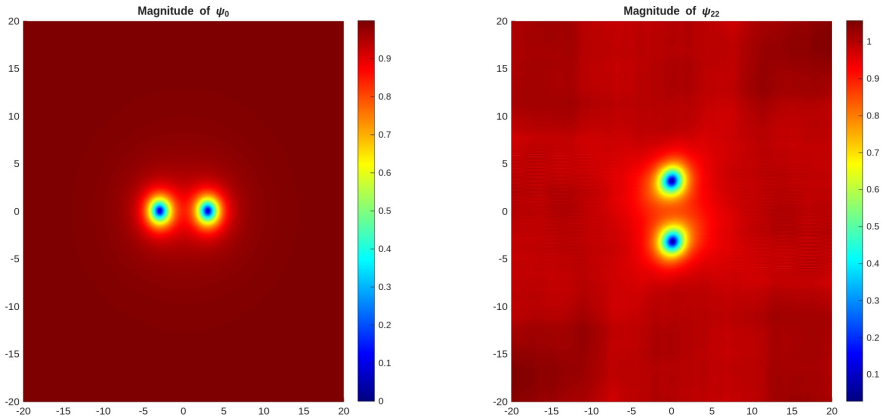

Figure 3.9: Initial solution (left) and solution at time $t = 22$ (right).

In this simulation, the grid was constructed according to 2.1.2 in both directions, with parameters $x_c = 3$, $h_1 = 0.05$, and $\delta = 0.025$, resulting in $m_x = m_y = 255$ grid points. The time step was set to $\tau = 0.05$.

3.3.1 Comparison between uniform and nonuniform grids

It is particularly interesting to examine how the solution changes depending on the chosen grid, while keeping the parameters fixed. We shall carry out one final experiment to compare the two grids, since in the non-stationary case we observed that the differences are much more pronounced than in the stationary one.

The following experiment was carried out using a nonuniform grid, advancing the simulation up to the time when the two vortices complete approximately a $\pi/2$ rotation. We then compare the vortex positions obtained on the uniform and nonuniform discretizations.

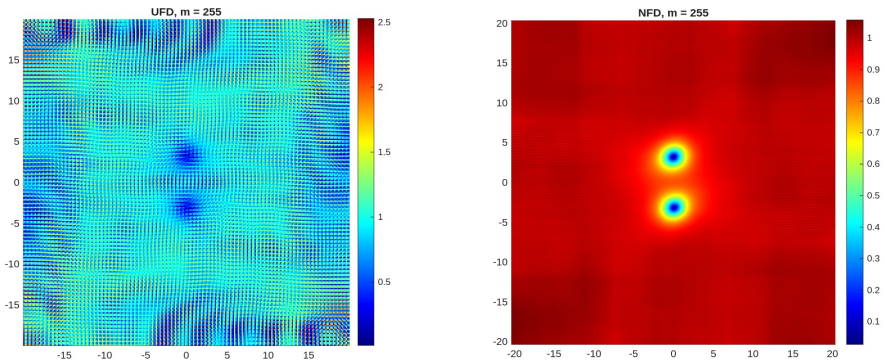

Figure 3.10: Comparison between equispaced grid and grid 3.13

It is clearly visible that the solution obtained with the nonuniform grid is significantly more accurate than the one computed on a uniform grid of the same size.

In order to achieve a comparable level of accuracy with a uniform grid, it is necessary to substantially increase the number of points, reaching as high as $m_x = m_y = 771$, resulting, as well, in a much larger computational time.

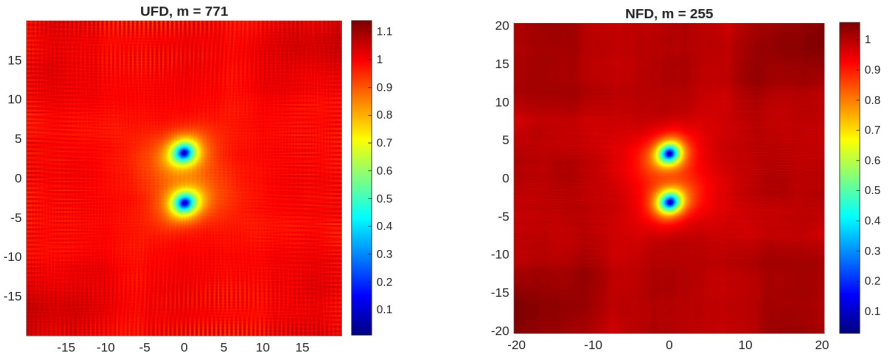

Figure 3.11: Comparison between uniform grid with $m = 771$ (left) and non uniform grid with $m = 255$ (right)

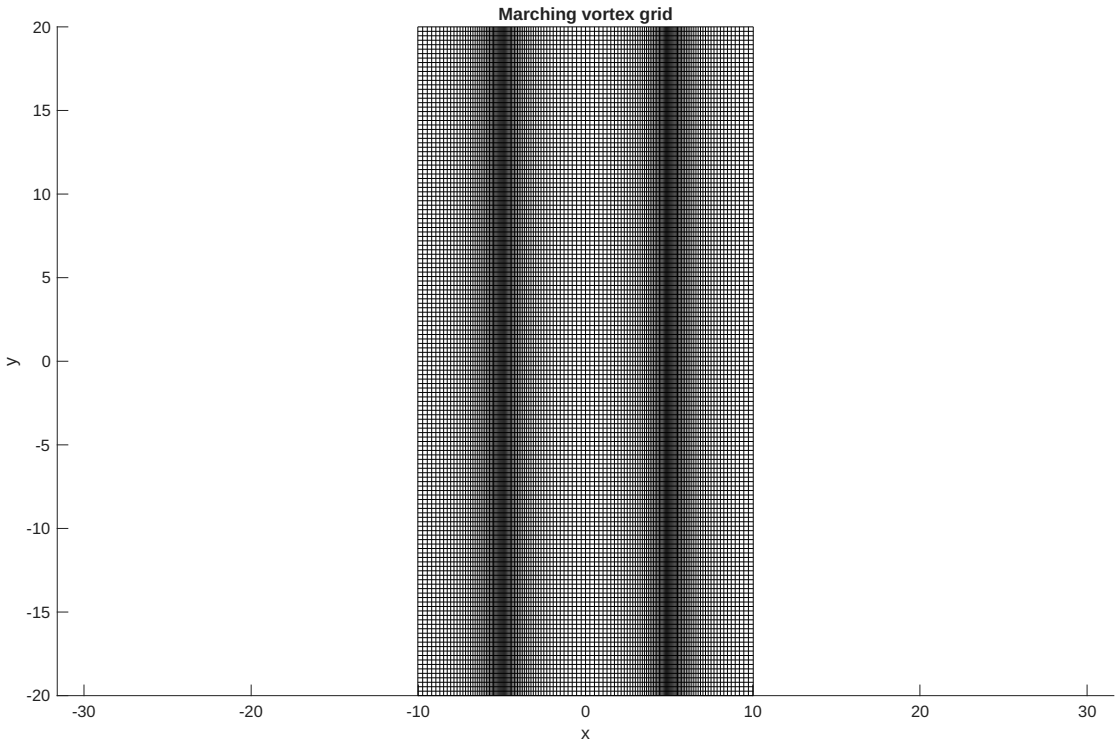

Figure 3.12: Grid used for the marching vortex simulations

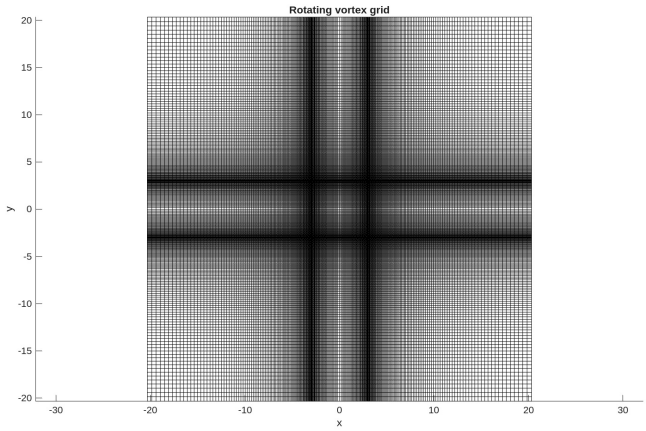

Figure 3.13: Grid used for rotating vortex simulation

3.4 Numerical conservation of phiysical quantities

3.4.1 Mass and energy conservation

Another insight into the quality of the proposed methods can be retrived by analyzing the conservation of physical invariants, such as mass and energy. It can be shown [2] that the time splitting method conserves mass. We employ the definition of mass given in 1.3 to compute the mass in the computational domain at each time step, with the trapezoidal quadrature rule, as

$$m(t_n) \approx h_x h_y \sum_{i=1}^{m_x} \sum_{j=1}^{m_y} \left|\psi(x_i, y_j, t_n)\right|^2 .$$

Appendix A

Kronecker products and its properties

From now on, in this section I denotes the identity matrix.

Definition A.1. Let $A \in \mathbb{R}^{m\times n}$ and $B \in \mathbb{R}^{p\times q}$. The *Kronecker product* is defined as

$$A \otimes B = \begin{bmatrix} a_{11}B & \cdots & a_{1n}B \\ \vdots & \ddots & \vdots \\ a_{m1}B & \cdots & a_{mn}B \end{bmatrix} \in \mathbb{R}^{mp\times nq}.$$

Example A.2.

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \otimes \begin{bmatrix} 0 & 5 \\ 6 & 7 \end{bmatrix} = \begin{bmatrix} 0 & 5 & 0 & 10 \\ 6 & 7 & 12 & 14 \\ 0 & 15 & 0 & 20 \\ 18 & 21 & 24 & 28 \end{bmatrix}.$$

Proposition A.3. *If A, B, C, D are matrices of compatible dimensions, then*

$$(A \otimes B)(C \otimes D) = (AC) \otimes (BD).$$

Proof. The (i, j) block of $(A \otimes B)(C \otimes D)$ is

$$\sum_k (a_{ik}B)(c_{kj}D) = \Big(\sum_k a_{ik}c_{kj}\Big)BD = (AC)_{ij}BD,$$

which coincides with the (i, j) block of $(AC) \otimes (BD)$. ∎

Proposition A.4. *Let $A \in \mathbb{C}^{m\times m}$, $B \in \mathbb{C}^{n\times n}$, such that $[A, B] = 0$ and $\mathbf{u} \in \mathbb{C}^{mn}$. Then*

$$(A \otimes B)\mathbf{u} = \mathrm{vec}\left(BUA^T\right)$$

where $\mathbf{u} = \mathrm{vec}(U)$, $U_{i,j} = u_{i+m(j-1)}$

Proof.

$$(A \otimes B)_{(i-1)m+k,\,(j-1)m+\ell} = a_{i,j}\, b_{k,\ell},$$

where $1 \le i, j \le n$ and $1 \le k, \ell \le m$.

The component of the product $(A \otimes B)\,\mathbf{u}$ corresponding to row index $(i-1)\,m + k$ is, by definition,

$$\begin{aligned}((A \otimes B)\,\mathbf{u})_{(i-1)m+k} &= \sum_{j=1}^{n} \sum_{\ell=1}^{m} (A \otimes B)_{(i-1)m+k,\,(j-1)m+\ell}\;\; u_{(j-1)m+\ell} \\ &= \sum_{j=1}^{n} \sum_{\ell=1}^{m} a_{i,j}\, b_{k,\ell}\, U_{\ell,j},\end{aligned}$$

Reordering the sums and factoring out $a_{i,j}$ gives

$$((A \otimes B)\,\mathbf{u})_{(i-1)m+k} = \sum_{\ell=1}^{m} b_{k,\ell} \left(\sum_{j=1}^{n} U_{\ell,j}\, a_{i,j} \right).$$

The inner sum is exactly the (ℓ, i)-entry of the matrix UA^T, since

$$\left(UA^T\right)_{\ell,i} = \sum_{j=1}^{n} U_{\ell,j}\, \left(A^T\right)_{j,i} = \sum_{j=1}^{n} U_{\ell,j}\, a_{i,j}.$$

Therefore,

$$((A \otimes B)\,\mathbf{u})_{(i-1)m+k} = \sum_{\ell=1}^{m} b_{k,\ell}\, \left(UA^T\right)_{\ell,i} = \left(B\,UA^T\right)_{k,i}.$$

Since this holds for all $i = 1, \ldots, n$ and $k = 1, \ldots, m$, we conclude that

$$(A \otimes B)\;\mathrm{vec}(U) = \mathrm{vec}\Big(B\,U\,A^T\Big).$$

■

Lemma A.5. *For every $n \in \mathbb{N}$, one has*

$$(I \otimes A)^n = I \otimes A^n.$$

Proof. By induction on n. For $n = 1$ the claim is trivial. Suppose it holds for $n - 1$; then

$$(I \otimes A)^n = (I \otimes A)(I \otimes A)^{n-1} = (I \otimes A)(I \otimes A^{n-1}) = I \cdot I \otimes A \cdot A^{n-1} = I \otimes A^n$$

where we used the previous proposition. ■

Lemma A.6. *Let $A, B \in \mathbb{C}^{m\times n}$ and X a matrix. Then $X \otimes A + X \otimes B = X \otimes (A + B)$.*

Proposition A.7. *Let $\{A_k\}_{k=1}^{\infty}$ be a sequence of matrices, where $A_j \in \mathbb{C}^{m\times n}$ for all j. Let I be the identity matrix of arbitrary dimensions. Then*

$$\sum_{k=1}^{\infty} (I_k \otimes A_k) = I_k \otimes \sum_{k=1}^{\infty} A_k$$

Proof. We consider the infinite sum as the limit of a partial sum as $\sum_{k=1}^{\infty} (I \otimes A_k) = \lim_{n\to\infty} \sum_{k=1}^{n} (I \otimes A_k)$.

We now proceed for induction on n. For $n = 1$ the claim is trivial. Suppose it holds for $n - 1$, then

$$\sum_{k=1}^{n} (I \otimes A_k) = I \otimes A_n + \sum_{k=1}^{n-1} (I \otimes A_k) = I \otimes A_n + I \otimes \sum_{k=1}^{n-1} A_k = I \otimes \sum_{k=1}^{n} A_k$$

■

Theorem A.8. *Let f be an analytic function. Then*

$$f(I \otimes A) = I \otimes f(A) \quad \textit{and} \quad f(A \otimes I) = f(A) \otimes I$$

Proof. Write $f(X) = \sum_{k=0}^{\infty} \alpha_k X^k$. Then

$$f(I \otimes A) = \sum_{k=0}^{\infty} \alpha_k (I \otimes A)^k = \sum_{k=0}^{\infty} \alpha_k (I \otimes A^k) = I \otimes \left(\sum_{k=0}^{\infty} \alpha_k A^k \right) = I \otimes f(A).$$

We proceed in the same way for the other. ■

Appendix B

MATLAB® implementation

In this chapter we describe the implementation of the numerical method in MATLAB®, which was used to perform the experiments described in Chapter 3.

The code is available at the following link. Here we report the main functions and implementation details.

B.1 Non uniform grid generation

The function `neqdeltagrid0.m` generates a non-uniform grid as described in Section 2.1.2. The function takes as input the parameters δ, h_1, L and m, and returns the grid points in the interval $[0, L]$.

```
% from xc to 0
for i = m-1:-1:2
    h(i) = (1 + delta)*h(i+1);
    x(i) = x(i+1) - h(i);
end

% from xc to xL
i = m + 1;
while x(i - 1) < xL
    h(i) = (1 + delta)*h(i-1);
    x(i) = x(i-1) + h(i);
    i = i + 1;
end

% always add 0
x = [0, x];
h = diff(x);

% check for the nearest point
    to 0
if numel(x) > 2
    if h(1) < 0.7*h(2)
        x(2) = [];
        h = diff(x);
    end
end
```

We then construct the full grid in $[-L, L]$ by mirroring the points around zero as

```
x = neqdeltagrid0(xc,Lx, delta, h1);
x = unique([-flip(x), x]); x = x(:);
hx = diff(x);
```

```
mx = length(x);
```

Once we have the grid points, we can construct the finite difference matrix as

```
%% matrix
dux = 2./(hx(1:mx-2) .* (hx(1:mx-2) + hx(2:mx-1))); % upper diag
dmx = -2./(hx(1:mx-2).*hx(2:mx-1));                 % main diag
dlx = 2./(hx(2:mx-1).*(hx(1:mx-2) + hx(2:mx-1)));   % lower diag

Dxx = spdiags([[dux;0;0], [0; dmx; 0], [0;0;dlx]], -1:1, mx,mx);

% Homogeneus Neumann conditions
Dxx(1,1:2) = [-2,2]/(hx(1)^2);
Dxx(mx,mx-1:mx) = [2,-2]/(hx(mx-1)^2);
```

B.2 Time splitting

We report a brief section of the code used to implement the time integration

```
for i = 1:nt - 1

    U = Ex*U*Ey.';
    U = exp(1i*tau/2*(1 - abs(U).^2)).*U;
    U = Ex*U*Ey.';

end
```

Bibliography

[1] M. Caliari. Dispense di metodi numerici per le equazioni differenziali. 2024.

[2] M. Caliari and S. Zuccher. Reliability of the time splitting fourier method for singular solutions in quantum fluids. *Computer Physics Communications*, 222:46–58, 2018.

[3] M. Caliari and S. Zuccher. A fast time splitting finite difference approach to gross-pitaevskii equations. *Communications in Computational Physics*, 29(5):1336–1364, May 2021.

[4] A.W. Baggaley S. Zuccher, M. Caliari and C.F. Barenghi. Quantum vortex reconnections. *Physics of Fluids*, 24(12):125108, 2012.