Master Thesis

2D Turbulence spreading

Author:
Víctor Ballester

Supervisors:
Alexandros Alexakis*
Emmanuel Dormy†

Abstract

This thesis investigates the propagation of locally injected turbulence within a doubly periodic 2D domain. Numerical simulations of the Navier-Stokes equations are conducted with inhomogeneous zero-mean forcing applied. Additionally, a point-vortex model is employed to compare vortex evolution between the two approaches. The results demonstrate that the energy carried by vortices spreads across the domain, as opposed to previous works in 3D domains, and reaches domain boundaries regardless of the size of the perturbation region if the Reynolds number is sufficiently high. Power laws for the energy and enstrophy distribution as a function of distance from the forcing region are identified, which are also consistent with the results found in the point-vortex model.

M2 Mathématiques Appliquées et Théoriques
Université Paris Dauphine - PSL
Laboratoire de Physique de l'École Normale Supérieure

June 30, 2024

*Professor in the Department of Physics at École Normale Supérieure. Webpage: https://www.phys.ens.fr/~alexakis/

†Professor in the Department of Mathematics at École Normale Supérieure. Webpage: https://www.math.ens.fr/~dormy/

Contents

1 Introduction

Turbulence in fluids is a ubiquitous phenomenon in our daily lives, manifesting in everything from the flow of water in a kitchen sink to the swirling smoke of a cigarette. These examples naturally occur in a three-dimensional (3D) space, as it is the intrinsic dimension of our world. However, this thesis focuses on the study of certain aspects of two-dimensional (2D) turbulence, which might initially seem less realistic. Despite this, 2D turbulence holds significant importance in the field of fluid dynamics. For instance, as explained in Boffetta and Ecke 2012, many large-scale atmospheric and oceanic phenomena exhibit properties closely related to those observed in 2D turbulence, making such phenomena well-approximated at first order by simplified 2D models.

Theoretically, 2D turbulence behaves quite differently from its 3D counterpart, contrary to natural intuition. While in 3D turbulence, energy cascades from large scales to small scales, in 2D turbulence, energy transfers from small scales to large scales. This fundamental difference is directly related to the primary motivation of this work: understanding the phase transitions of turbulent flows in the atmosphere. Although the atmosphere is a 3D domain, it is relatively thin in height, allowing for the observation of phenomena typical of 2D turbulence.

This thesis investigates the behavior of 2D turbulence under the influence of a localized inhomogeneous forcing continuously applied to the system. Several questions arise initially: Will the vortices reach the domain boundaries, or will they dissipate as they spread? How does the energy distribution evolve with distance from the forcing region? These questions are thoroughly addressed in this study. Similar question were explored in Alexakis 2023. In that study the author considered a long periodic 3D box and found that energy does not reach the domain boundaries if the boundaries are sufficiently large, regardless of the Reynolds number. This work tries to extend that study to the 2D case.

In the present study, a doubly periodic 2D box is analyzed with an inhomogeneous, but zero-mean, forcing applied to the system. The zero-mean nature of the forcing ensures no net momentum injection into the system. Additionally, a point-vortex model is examined to compare its results with those obtained from the 2D Navier-Stokes equations.

The structure of this thesis is organized as follows: Section 2 details the theoretical framework of our problem and defines all the relevant quantities. Section 3 describes the numerical setup and presents the results obtained during the study. Finally, Section 4 provides conclusions and discusses the implications of our simulations.

2 Theoretical background

The primary focus of this work is the integration of the incompressible Navier-Stokes equations with a random forcing term:

$$\partial_t \boldsymbol{u} + (\boldsymbol{u}\cdot\boldsymbol{\nabla})\boldsymbol{u} = -\boldsymbol{\nabla}p + \nu\Delta\boldsymbol{u} + f_0\boldsymbol{f} \tag{2.1}$$

$$\boldsymbol{\nabla}\cdot\boldsymbol{u} = 0 \tag{2.2}$$

where $\boldsymbol{u}$ is the velocity field, p is the pressure, ν is the kinematic viscosity, and $f_0\boldsymbol{f}$ is a random forcing term satisfying $\boldsymbol{\nabla}\cdot\boldsymbol{f} = 0$. For the sake of clarity and simplicity the amplitude factor f_0 of the forcing have been factor out from the force itself. The second equation is called *incompressibility condition* and it translates the fact that the fluid cannot be compressed. The forcing term is chosen to be Gaussian for the vorticity formulation, which is detailed below. Throughout the project the density of the fluid is assumed to be constant and equal to 1.

2.1. Stream function formulation

In Batchelor 2000, a new variable is introduced in order to simplify the integration of the 2D incompressible Navier-Stokes equations. This quantity, called *stream function* and denoted by ψ, is defined as the flow rate across a given line. More accurately, if $\mathcal{C}$ is a curve joining two points O (fixed) and $P = (x, y)$, the stream function streams function as a function of the coordinates of the point P is then

$$\psi(x,y) - \psi_0 = \int_{\mathcal{C}} \boldsymbol{u}^\perp \cdot \mathrm{d}\boldsymbol{s} = \int_O^P -v\,\mathrm{d}x + u\,\mathrm{d}y \tag{2.3}$$

where $\boldsymbol{u} = (u, v)$ is the velocity field, $\mathrm{d}\boldsymbol{s} = (\mathrm{d}x\,, \mathrm{d}y)$ is the tangent vector to the curve, and ψ_0 is a reference value. In differential form, it can be written as

$$\mathrm{d}\psi = -v\,\mathrm{d}x + u\,\mathrm{d}y = \frac{\partial\psi}{\partial x}\,\mathrm{d}x + \frac{\partial\psi}{\partial y}\,\mathrm{d}y \tag{2.4}$$

where the last equality follows from the exact differential property. Thus, one obtain the following useful relations:

$$u = \frac{\partial\psi}{\partial y} \quad \text{and} \quad v = -\frac{\partial\psi}{\partial x} \tag{2.5}$$

Note the arbitrary choice of $\boldsymbol{u}^\perp$ in the definition of the stream function. In the present work, the choice is made to be $\boldsymbol{u}^\perp = (-v, u)$, in order to keep the same sign convention as in similar works (Boffetta and Ecke 2012; Alexakis and Biferale 2018). The formulation with the alternative stream function $\psi' := -\psi$ is sometimes used in other fields of fluid dynamics, mostly in meteorology and oceanography.

Note that using this definition, the incompressible condition $\boldsymbol{\nabla}\cdot\boldsymbol{u} = 0$ is automatically satisfied. Finally, introducing the scalar vorticity $\omega := \boldsymbol{\nabla}\times\boldsymbol{u} = \frac{\partial v}{\partial x} - \frac{\partial u}{\partial y} = -\Delta\psi$, one can rewrite the Navier-stokes equations in terms of the this latter variable:

$$\partial_t\omega + (\boldsymbol{u}\cdot\boldsymbol{\nabla})\omega = \nu\Delta\omega + f_0 f_\omega \tag{2.6}$$

$$\boldsymbol{\nabla}\cdot\boldsymbol{\omega} = 0 \tag{2.7}$$

where the rotational has been taken to both sides of Eqs. (2.1) and (2.2) and basic vector identities have been used. The main objects of interest in the vorticity formulation are the *vortices* (see POSAR FIGURA D'ALGUNS VORTICES), which according to Saffman 1993 are the local regions on the plane with non vanishing vorticity and surrounded with an irrotational flow.

Now, using the relation between the stream function and the vorticity one obtains:

$$\partial_t\psi + \Delta^{-1}(\boldsymbol{u}\cdot\boldsymbol{\nabla})\Delta\psi = \nu\Delta\psi + f_0 f_\psi \tag{2.8}$$

The reader may quickly notice that this equation appears to be more complicated than the first one. However, when transforming the equation to Fourier space, it becomes much more simpler (see Eq. (2.13)), with the advantage of having a scalar function as the main unknown variable (as opposed to the velocity formulation) and removing the incompressible condition (as opposed to the vorticity formulation), which is implicit in the definition of the stream function.

The forcing term is assumed to be Gaussian, in particular it is taken of the form:

$$f_\omega(x,y) = \sum_{i=1}^{10} A_i \exp\left(-\frac{k_\ell^2}{2}\left[(x-x_{i,1})^2 + (y-y_{i,1})^2\right]\right) - A_i \exp\left(-\frac{k_\ell^2}{2}\left[(x-x_{i,2})^2 + (y-y_{i,2})^2\right]\right) \quad (2.9)$$

The quantities A_i follow a uniform distribution between 0 and 1; k_ℓ quantifies the size of the vortices (thought in Fourier space) so that the vortices injected have size $\sim 1/k_\ell$, and the coordinates $x_{i,j}$, $y_{i,j}$, for $j = 1, 2$ are random variables that position the vortices inside a small disk of radius k_r centered at the origin such that the density of vortices is (almost) constant. To be more clear, if one expresses the coordinates of the vortices in polar coordinates, the angular variable is uniformly distributed between 0 and 2π and the radial variable follows a distribution $\sqrt{\mathcal{U}(0, \pi/k_r)}$ where $\mathcal{U}(a, b)$ is the uniform distribution between a and b. Indeed, one can easily check that the probability of finding a vortex inside a thin annulus within the perturbation region does not depend on the radius of the annulus r:

$$\mathbb{P}\left(r - \mathrm{d}r < \sqrt{\mathcal{U}(0,\pi/k_r)} \le r\right) = \int_{(r-\mathrm{d}r)^2}^{r^2} \frac{1}{\pi/k_r}\,\mathrm{d}s \simeq Cr\,\mathrm{d}r + \mathcal{O}(\mathrm{d}r^2) \quad (2.10)$$

Thus, since the area of the annulus $\{(x,y) \in \mathbb{R}^2 : r - \mathrm{d}r < \sqrt{x^2+y^2} \le r\}$ is proportional to $r\,\mathrm{d}r$ at first order, the density of vortices is constant up to a small error of order $\mathrm{d}r$ across the perturbation region.

With this forcing the aim is to introduce 10 pairs of vortices with opposite vorticity in a non-homogeneous way which a priori may introduce a nonzero momentum to the system. To correct that, the first Fourier coefficient is set to zero once transformed the forcing term to Fourier space. This implies that the actual force differs up to a constant factor from the one given above, which from now on we will assume that this factor is implicit in f_ω, that is $\mathbb{E}(f_\omega) = 0$.

The amplitude of the forcing is controlled by the parameter f_0, which is chosen in such a way to keep the injection rate of energy constant and equal to 1. Because of that it has an implicit dependence on time, as explained below.

Forward-transforming f_ω to Fourier space, from the relation $\omega = -\Delta\psi$, one can easily get $\widehat{f_\psi}$ by dividing by k^2 each mode of $\widehat{f_\omega}$, being k the norm value of the wave vector.

2.2. Fourier space

The *Fourier transform* (FT) of a function $f : \mathbb{R}^2 \to \mathbb{R}$ is defined as

$$\widehat{f}(\boldsymbol{\xi}) = \int_{\mathbb{R}^2} f(\boldsymbol{x}) \mathrm{e}^{-\mathrm{i}\boldsymbol{\xi}\cdot\boldsymbol{x}}\,\mathrm{d}\boldsymbol{x} \quad (2.11)$$

for all $\boldsymbol{\xi} \in \mathbb{R}^2$ and its discrete version (DFT) for a square domain with N points in each direction is

$$\widehat{f}(\boldsymbol{k}) = \sum_{\boldsymbol{n}\in\boldsymbol{N}} f_{\boldsymbol{n}} \mathrm{e}^{-\mathrm{i}\boldsymbol{k}\cdot\boldsymbol{n}/N} \quad (2.12)$$

where $\boldsymbol{N} = \{0, 1, \ldots, N-1\}^2$ is the set of points in the Fourier grid, $f_{\boldsymbol{n}}$ is the value of the function in the physical space at the point $\boldsymbol{n}$, and $\boldsymbol{k} = (k_x, k_y)$ is the wave vector.

Taking the Fourier transform on both sides of Eq. (2.8) and using the well-known properties of the Fourier transform, one obtains:

$$\frac{\mathrm{d}}{\mathrm{d}t}\widehat{\psi} - k^{-2}\widehat{(\boldsymbol{u}\cdot\boldsymbol{\nabla})\Delta\psi} = -\nu k^2 \widehat{\psi} + \widehat{f_\psi} \quad (2.13)$$

where $k := \|\boldsymbol{k}\|$. Note that the non-linear term in the above equation has not been simplified. This is because in the simulation that term is backward-transformed to the physical space, computed, and then transformed back to the Fourier space, as it is less expensive than computing the non-linear term in Fourier space.

2.3. *Reynolds number*

The Reynolds number is a dimensionless quantity that characterizes the ratio of inertial forces to viscous forces in a fluid. It is usually defined as $\mathrm{Re} := UL/\nu$, where U is a characteristic velocity of the flow, L is the characteristic length scale associated to it and ν is the kinematic viscosity. In the cases where there is no control on the injection velocity, but instead one can control the rate of energy injection ϵ, this equation is not useful anymore. To derive an alternative equation, the scaling theory of Kolmogorov is used (Frisch 1995). Let $\epsilon_\ell \sim u_\ell^2/\tau_\ell$ be the rate of change of energy at the scale ℓ, where u_ℓ is the typical velocity at that scale and τ_ℓ is the characteristic time at that scale. Using that $\tau_\ell \sim \ell/u_\ell$, one obtains $\epsilon_\ell \sim u_\ell^3/\ell$. Assuming that the energy transferred from the scale ℓ to smaller scales is the same as the energy received by the scale ℓ from larger scales (that is, the flux of energy across scales is constant), one obtains that the value ϵ does not depend on L and moreover $\epsilon \sim U^3/L$, i.e. $U \sim (\epsilon L)^{1/3}$. Thus, one obtains a new formula for the Reynolds number:

$$\mathrm{Re} = \frac{\epsilon^{1/3} L^{4/3}}{\nu} \tag{2.14}$$

In the present work, the length scale L is determined by the size of the vortices injected in the disk, which in this case is $L = 1/k_\ell$. The injection rate of energy is determined by the forcing term, as follows. The amplitude f_0 is time dependent and is chosen in such a way that f_0^2 is the rate at which energy is injected per unit of area in the domain. More precisely, the amplitude of the forcing is taken as $f_0/\sqrt{\Delta t/2}$, where Δt is the time step of the simulation, a priori varying with time.

To check that indeed this is the case, let $\boldsymbol{f} = (f_u, f_v)$ be the forcing term for the velocity equation in the physical space. Recall that since $\mathbb{E}(f_\omega) = 0$ and $f_\omega = \boldsymbol{\nabla}\times \boldsymbol{f}$, using the inversion formula for the curl (see Eq. (2.26)) and Fubini's theorem that $\mathbb{E}(f_u) = \mathbb{E}(f_v) = 0$. Thus, at time $t = T$ and time step Δt one has, accounting only for the forcing terms:

$$u(\boldsymbol{x}, T + \Delta t) = u(\boldsymbol{x}, T) + \Delta t \, \frac{f_0}{\sqrt{\Delta t/2}} \, f_u(\boldsymbol{x}) + \cdots \tag{2.15}$$

Squaring both sides of the equation one obtains:

$$u(\boldsymbol{x}, T + \Delta t)^2 = u(\boldsymbol{x}, T)^2 + 2\sqrt{2\Delta t}\, f_0 f_u(\boldsymbol{x}) u(\boldsymbol{x}, T) + 2\Delta t\, f_0^2 f_u(\boldsymbol{x})^2 + \mathcal{O}(\Delta t^{3/2}) \tag{2.16}$$

Taking expectation on both sides we conclude:

$$\frac{u(\boldsymbol{x}, T + \Delta t)^2 - u(\boldsymbol{x}, T)^2}{\Delta t} = 2 f_0^2 \, \mathbb{E}(f_u(\boldsymbol{x})^2) \tag{2.17}$$

where the second term on Eq. (2.16) vanishes because of the zero mean of the forcing term. Doing a similar computation for the v component, summing both equations and taking the limit $\Delta t \to 0$, one obtains the rate of energy injected in the domain.

$$\frac{\partial E}{\partial t} = \frac{1}{2}\frac{\partial}{\partial t}(u^2 + v^2) = f_0^2 \, \mathbb{E}(f_u^2 + f_v^2) \tag{2.18}$$

The implicit assumption that $\rho = 1$ in the equation ensures that it precisely represents the rate of energy injected per unit area in the domain. In the simulations at each step we adjust the amplitude of the forcing term so that its variance is equal to 1, making the rate of energy injection become f_0^2. Since the interest is, not on the injection rate of energy per unit of area, but on the injection rate of energy per unit of area in the forcing region, the following formula for ϵ is used:

$$\epsilon = f_0^2 \, \frac{4\pi^2}{\pi(\pi/k_r)^2} \tag{2.19}$$

Consequently the Reynolds number becomes:

$$\mathrm{Re} = \frac{\left(f_0^2 \, \frac{4k_r^2}{\pi}\right)^{1/3} k_\ell^{-4/3}}{\nu} \tag{2.20}$$

2.4. Point-vortex model

The study has accomplished another simulation, far from the Navier-Stokes equations, but aiming to obtain qualitatively and quantitatively similar results. This simulation is based on the point-vortex model, which is a simplified version of the dynamics of a set of vortices.

More precisely, in the point-vortex model that follows, the viscous term is neglected. Thus, this problem aims to mimic in some way the dynamics of the Navier-Stokes equations as $\nu \to 0$ (or equivalently $\mathrm{Re} \to \infty$, see Eq. (2.20)). Moreover, the vortices are considered as point-like objects, that is to say, being the vorticity singular at the position of the vortices. Because of that, the vortices are characterized by their circulation

$$\Gamma = \lim_{\mathrm{diam}(\mathcal{C})\to 0} \int_{\mathcal{C}} \boldsymbol{u} \cdot \mathrm{d}\boldsymbol{s} \tag{2.21}$$

where $\mathcal{C}$ is a curve surrounding the vortex. In those conditions and in absence of external forces, the vorticity field ω is only advected and its evolution in time is described by the following transport equation:

$$\partial_t \omega + (\boldsymbol{u} \cdot \boldsymbol{\nabla})\omega = 0 \tag{2.22}$$

together with the incompressible condition $\boldsymbol{\nabla} \cdot \boldsymbol{u} = 0$ (Ceci and Seis 2022). Here $\boldsymbol{u}$ is the velocity field generated by the vortices. An interesting interpretation of the vortices involves the Dirac's delta distribution. Specifically, the vorticity field generated by a point vortex located at $\boldsymbol{y}$ can be represented as a distribution of the form:

$$\omega(\boldsymbol{x}, t) = \Gamma \delta(\boldsymbol{x} - \boldsymbol{y}) \tag{2.23}$$

where Γ is the circulation of the vortex (Saffman 1993). The dynamics of N point vortices in the $\mathbb{R}^2$ plane are described in the following theorem:

Theorem 1. Consider N vortices at positions $\boldsymbol{z}_1, \ldots, \boldsymbol{z}_N \in \mathbb{R}^2$ and with circulations $\Gamma_1, \ldots, \Gamma_N$. Then, their evolution in time is described by the following system of ordinary differential equations (Aref 2007):

$$\dot{x}_i = -\sum_{j \neq i} \frac{\Gamma_j}{2\pi} \frac{y_i - y_j}{(x_i - x_j)^2 + (y_i - y_j)^2} \tag{2.24}$$

$$\dot{y}_i = \sum_{j \neq i} \frac{\Gamma_j}{2\pi} \frac{x_i - x_j}{(x_i - x_j)^2 + (y_i - y_j)^2} \tag{2.25}$$

for $i = 1, \ldots, N$, where $\boldsymbol{z}_i = (x_i, y_i)$.

Proof. Since there is no viscosity in the equations, each vortex is advected with the velocity field generated by all the other vortices. Now, given $\boldsymbol{f} \in \mathcal{C}^1(\mathbb{R}^2, \mathbb{R}^2)$ such that $\boldsymbol{\nabla} \cdot \boldsymbol{f} = 0$ and $g := \boldsymbol{\nabla} \times f$, by Biot-Savart law one can invert the curl operator using the Biot-Savart kernel (see Griffiths 2023):

$$\boldsymbol{f}(\boldsymbol{x}) = (\boldsymbol{K} * g)(\boldsymbol{x}) = \int_{\mathbb{R}^2} \boldsymbol{K}(\boldsymbol{x} - \boldsymbol{y}) g(\boldsymbol{y}) \, \mathrm{d}\boldsymbol{y} \tag{2.26}$$

where $\boldsymbol{K}(\boldsymbol{x}) = \dfrac{1}{2\pi} \dfrac{(-x_2, x_1)}{\|\boldsymbol{x}\|^2}$ is the Biot-Savart kernel and $\boldsymbol{x} = (x_1, x_2)$. In our case, $\boldsymbol{g}$ is the vorticity field which is a sum of δ's. Thus, taking into consideration Eq. (2.23), the velocity field generated by a vortex at $\boldsymbol{z}_j$ with circulation Γ_j is:

$$\boldsymbol{u}_j(\boldsymbol{x}) = \boldsymbol{K}(\boldsymbol{x} - \boldsymbol{z}_j)\Gamma_j \tag{2.27}$$

The proof concludes using the superposition principle. □

Another aspect worth-mentioning is the numerical addition of a softening parameter ε to Eqs. (2.24) and (2.25) in order to prevent the positions of the vortices from blowing up when two vortices get too close to each other. The equations then take the following form:

$$\dot{x}_i = -\sum_{j \neq i} \frac{\Gamma_j}{2\pi} \frac{y_i - y_j}{(x_i - x_j)^2 + (y_i - y_j)^2 + \varepsilon^2} \tag{2.28}$$

$$\dot{y}_i = \sum_{j \neq i} \frac{\Gamma_j}{2\pi} \frac{x_i - x_j}{(x_i - x_j)^2 + (y_i - y_j)^2 + \varepsilon^2} \tag{2.29}$$

This adjustment avoids the singularities of the system which happen when two vortices get too close to each other, which in Section 3.2 is shown to be the common behavior of the system. The softening parameter is chosen to be $\varepsilon = 0.001$, being this value small enough not to affect the dynamics of the system but large enough to prevent a numerical blow-up.

In order to narrow the gap between the point-vortex model and the Navier-Stokes equations while keeping the simplicity of the former, a set of vortices is added regularly in time to simulate the action of a forcing term in the equations. The point vortices are added in a similar way as in the Navier-Stokes, but this time making the addition completely symmetric to cancel out any nonzero momentum that could be introduced in the system.

2.5. Monitored quantities

In order to keep track of the evolution of the system, several variables are monitored during the simulation.

For the Navier-Stokes equations, the main quantity of interest is the total energy and vorticity in the system. Since the work environment is the Fourier space (see Section 3.1), the energy is computed as

$$E = \sum_{\boldsymbol{k}\in\boldsymbol{K}} \|\widehat{\boldsymbol{u}}(\boldsymbol{k})\|^2 = \sum_{\boldsymbol{k}\in\boldsymbol{K}} k^2 |\widehat{\psi}(\boldsymbol{k})|^2 \tag{2.30}$$

which by the Parseval identity is equivalent to the total energy in the physical space. Here $\boldsymbol{K} = \{0, 1, \dots, K-1\}^2$ is the set of points in the Fourier grid, and $K = N/3$ is the maximum wave number chosen in order to control the aliasing effects. Another important quantity worth-considering is *enstrophy*, which is analogous to energy but uses vorticity instead of velocity as the primary variable. It is defined as

$$\Omega = \sum_{\boldsymbol{k}\in\boldsymbol{K}} |\widehat{\omega}(\boldsymbol{k})|^2 = \sum_{\boldsymbol{k}\in\boldsymbol{K}} k^4 |\widehat{\psi}(\boldsymbol{k})|^2 \tag{2.31}$$

where the second equality follows from the relation $\omega = -\Delta\psi$.

As the title suggests, the main purpose of this work is to study how turbulence is spread across the domain. Thus, quantities relating the energy (and enstrophy) contained in rings as a function of the radius are also taken into account. These variables are respectively denoted by E_k and Ω_k, and they are given by the following expressions:

$$E_k = \sum_{k-1<\|\boldsymbol{k}\|\le k} \|\widehat{\boldsymbol{u}}(\boldsymbol{k})\|^2 \tag{2.32}$$

$$\Omega_k = \sum_{k-1<\|\boldsymbol{k}\|\le k} \|\widehat{\boldsymbol{\omega}}(\boldsymbol{k})\|^2 \tag{2.33}$$

where $k \in \{1, \dots, K\}$. The quantities E_k and Ω_k are then plotted as a function of k in order to study the energy and enstrophy distribution across the domain (see Section 3.2).

Related to those two latter variables is the mean energy radius and mean enstrophy radius, which refer to a weighted average of the radius where most of the energy (and enstrophy, respectively) is contained. They are defined as:

$$\mathcal{R}_E = \frac{\sum_{k=1}^{K} k^2 E_k}{\sum_{k=1}^{K} E_k} \tag{2.34}$$

$$\mathcal{R}_\Omega = \frac{\sum_{k=1}^{K} k^2 \Omega_k}{\sum_{k=1}^{K} \Omega_k} \tag{2.35}$$

Note that since the domain is square, the sums $\sum_{k=1}^{K} E_k$ and $\sum_{k=1}^{K} \Omega_k$ are slightly less than the total energy E and enstrophy Ω respectively, since they account for the contributions from the different annuli in the domain until their radii reach the radius of the incircle of the square.

Regarding the point-vortex model, the system is integrable and the following function H

$$H = \sum_{i=1}^{N} H_i, \qquad H_i = -\frac{1}{4\pi} \sum_{j\neq i} \Gamma_j \log \|\boldsymbol{z}_i - \boldsymbol{z}_j\|^2 \tag{2.36}$$

is a first integral of the system. Moreover the system is Hamiltonian, that is, it can written as:

$$\dot{x}_i = \frac{\partial H}{\partial y_i} \tag{2.37}$$

$$\dot{y}_i = -\frac{\partial H}{\partial x_i} \tag{2.38}$$

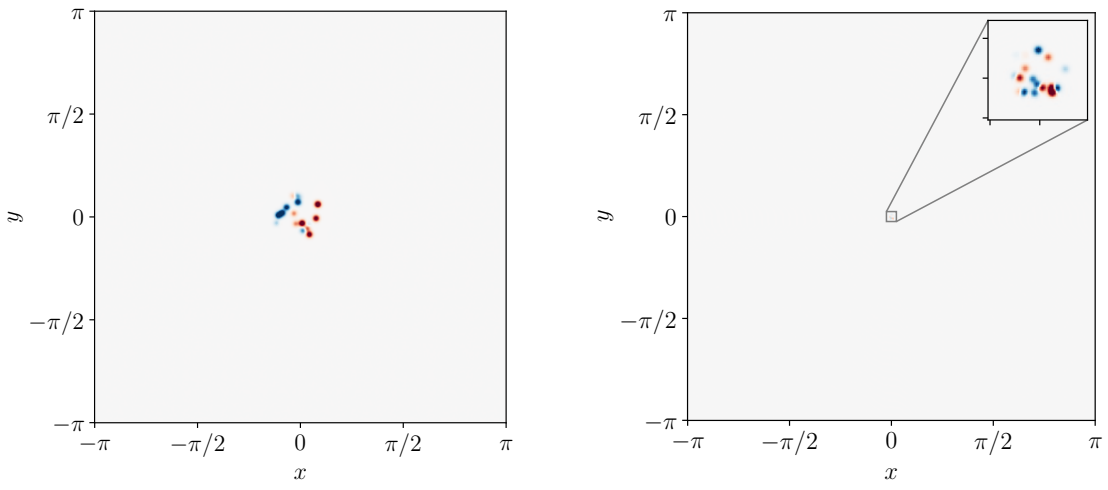
At first sight, the reader may notice that the function H is singular when two vortices get too close to each other. Although one could try to create an auxiliary function $f(H)$ that is regular in the whole domain and has a shape according the physical intuition while keeping the properties of a first integral, an alternative approach was chosen in this work to mimic the energy and enstrophy profiles for the Navier-Stokes equation.

The idea explored here relies on counting the number of point vortices in annuli and compare the density distribution with the respective functions E_k and Ω_k, once transformed back to the physical space.

3 Simulation

3.1. Numerical setup

The 2D incompressible Navier Stokes equations are forced in a periodic domain of size $2\pi \times 2\pi$ with a forcing term that is located in a disk of radius π/k_r centered at the origin. The range of values for the parameter k_r is taken to be $\{8, 16, 32, 64\}$ and in all the cases the size of the vortices, which is controlled by k_ℓ, is set to $k_\ell = 4k_r$. The parameter k_r being one of those values in the previous set represents how smaller is the perturbation region (in diameter) compared to the domain size (2π). The other parameter k_ℓ accounts for the size of the vortices, as $1/k_\ell$ represents a typical length scale of the vortices. Fig. 1 shows a graphical representation of the forcing term for two different values of k_r.



(a) $k_r = 8$ **(b)** $k_r = 64$

Figure 1: Vorticity forcing term for different values of k_r. Red colors and blue colors mean different direction of rotation for each vortex. The reader may notice that indeed the diameter of the forcing region is about 8 times smaller than the total size of the domain. In the second plot, this property is less noticeable, but it is still true, this case 64 times smaller.

The Reynolds number is the other parameter that plays an important role in the whole simulation. This project has simulated fluid flows for Reynolds numbers within the set $\{0.25, 0.5, 1, 2, 4, 8, 16, 32, 64, 128\}$, each of those requiring different resolution as we increase the Reynolds number in order to capture the smallest scales where energy gets dissipated by viscosity.

A pseudo-spectral method is used to solve the Navier Stokes equations, based on the Fourier basis and then using an improved 2th-order low-storage Runge-Kutta method to integrate the resulting ordinary differential equation. As explained in Brachet et al. 2008, this method differs from the conventional Runge-Kutta methods by reducing the amount of storage needed for each iteration at the expense of roughly doubling the time needed for evaluating the temporal derivatives at the same order as the usual

Runge-Kutta methods. Specifically, if $\boldsymbol{\psi}_n$ is the vector containing all the coordinates at the n-th step of the integration, the scheme follows the subsequent steps:

1. Copy $\widehat{\boldsymbol{\psi}}_n$ into $\widehat{\boldsymbol{\psi}}_*$.
2. For $i = s, \ldots, 1$, s being the number of stages of the Runge-Kutta method, update $\widehat{\boldsymbol{\psi}}_*$ as follows:

$$\widehat{\boldsymbol{\psi}}_* \leftarrow \widehat{\boldsymbol{\psi}}_n + \Delta t \frac{\boldsymbol{F}(\widehat{\boldsymbol{\psi}}_*)}{i} \tag{3.1}$$

   where $\boldsymbol{F}$ represents the field that defines the differential equation for $\widehat{\psi}$.
3. Set $\widehat{\boldsymbol{\psi}}_{n+1} := \widehat{\boldsymbol{\psi}}_*$.

In the second step, the evaluation of $\boldsymbol{F}$ is done in an explicit-exact manner. This means that the nonlinear term is treated explicitly in time, while the linear terms are solved exactly using their exponential solution. For more information about the scheme, the reader is encouraged to read the article from Brachet et al. 2008 or check the source codes in the link provided below. In this work, $s = 4$ is used for all simulations. While this produces a formal order of accuracy of 2, the errors are generally smaller compared to those of a standard second-order Runge-Kutta method.

The codes are run in two supercomputer centers, IDRIS[1] and MESOPSL[2], using 40 to 80 cores, depending on the simulation. Two different kinds of simulations are performed: fully parallel simulations and embarrassingly parallel simulations. In the parallel simulations, the Fourier domain is divided among all the processors, allowing them to work simultaneously on different parts of the problem. In the embarrassingly parallel simulations, each simulation runs independently on a single core. Multiple simulations are executed concurrently, one on each available core, and the results are averaged afterwards to obtain more accurate conclusions. This project uses MPI compilers to do the parallelism. Details about the parallelization of the code will not be delved into, but the main idea will be explained.

They key piece of the parallelization of any pseudo-spectral method is the efficient computation of the multidimensional Fourier transform. As a starting point, the physical domain, of size $N \times N$, is split in one dimensions creating several subdomains of sizes $\tilde{N} \times N$, where $\tilde{N} \simeq N/N_{\text{cores}}$ and N_{cores} is the number of cores used. Each core is responsible for computing $\tilde{N}$ 1D Fourier transforms using the standard Fast Fourier Transform (FFT) algorithm which reduces the operations from $\mathcal{O}(N^2)$ (using the naive approach) to $\mathcal{O}(N \log N)$. Since the initial data is real-valued, the complex-valued transformed data is then stored in an array $\tilde{N} \times (N/2+1)$, which is enough to store all the necessary information. Next, MPI communication is carried out in order to gather all the data, transpose it, and then split it again to produce slices of size $\bar{N} \times N$, where $\bar{N} \simeq (N/2+1)/N_{\text{cores}}$. Each core is, as before, responsible for computing $\bar{N}$ 1D Fourier transforms. Finally, all the data is gathered again to produce the desired FFT resulting in a memory block of size $(N/2+1) \times N$ consisting of complex-valued numbers. If the reader is interested in the details, the article from Mininni et al. 2011 is highly recommended.

For the parallel code, a variable time step is used throughout the whole simulations in order to take into account the advection stability condition. For the embarrassingly parallel code, a fixed time step is used, for the purpose of better comparing the results between the different runs from the same simulation. The time step is chosen by eye after studying the evolution of the time steps during the variable-time-step fully parallel simulations. Table 1 shows the different simulations performed during the project as well as the resolution in physical space used in each case.

The reader may observe that the resolution increases as both the Reynolds number and the forcing parameter k_r increase. For the former, the resolution is increased to resolve the smaller scales that appear in the system, which play an essential role in dissipating energy through viscosity. Thus, as Re increases, ν decreases, and the predicted Kolmogorov wave number, where dissipation occurs, becomes larger. For the latter, the resolution is increased as we raise the wave numbers of the forcing region, thereby shifting the energy injection to higher frequencies. It is worth-noting that the resolution in Fourier space is not the same as the one in physical space. Specifically, as mentioned before, the Fourier resolution is one third of the physical resolution in each dimension.

[1] For more information about the resources they provide, check their website: http://www.idris.fr/ (accessed on June 30, 2024).

[2] For more information about the resources they provide, check their website: https://wwwmesopsl-new.obspm.fr/ (accessed on June 30, 2024).

k_r \ Re	0.25	0.5	1	2	4	8	16	32	64	128
8	✓$_{512}$	✓$_{512}$	✓$_{512}$	✓✓$_{512}$	✓✓$_{1024}$	✓✓$_{1024}$	✓✓$_{1024}$	✓✓$_{2048}$	✓$_{2048}$	✓$_{4096}$
16				✓$_{1024}$	✓$_{2048}$	✓✓$_{2048}$	✓✓$_{2048}$	✓✓$_{2048}$	✓$_{4096}$	✓$_{4096}$
32				✓$_{2048}$	✓$_{4096}$	✓✓$_{4096}$	✓✓$_{4096}$	✓✓$_{4096}$	✓$_{8192}$	✓$_{8192}$
64						✓$_{8192}$	✓$_{8192}$	✓$_{8192}$		

Table 1: Simulations carried out during the study varying the Reynolds number and the forcing parameter k_r. In all cases $k_\ell = 4k_r$. The green check mark symbols indicate the simulations done in parallel, splitting the domain between different cores. The blue check mark symbols indicate the simulations done in embarrassingly parallel, where each simulation is done in a single but simultaneously in many cores in order to produce statistical results. In each cell, the number indicates the resolution in each dimension employed, which have been proved (a posteriori) to be enough to well-resolve the system.

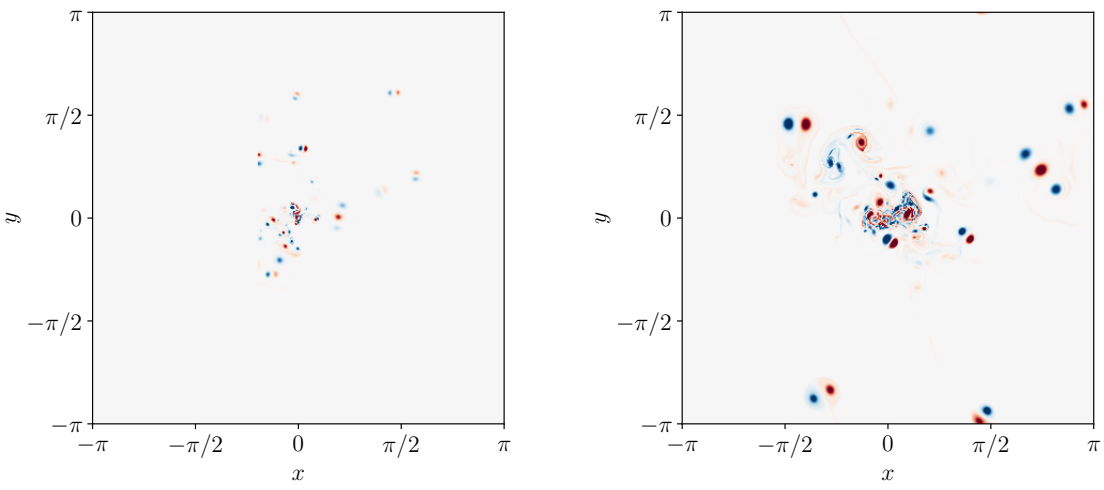
The integration of the system of differential equations (see Eqs. (2.28) and (2.29)) is done using a Runge-Kutta (7)8 method for adaptative time-step by means of the Fehlberg error estimate. These adaptative Runge-Kutta methods are based on the idea of using two different approximations of the solution at each step, in this case, one of order 7 and another of order 8. Then, the difference between both approximations is used to estimate the error between one of the approximations and the real solution. If the error is below a certain threshold, the time step is increased, and if it is above, the time step is decreased. The simulations for the point vortex model are done, as opposed to the integration of the Navier-Stokes equations, in a personal computer and in a single core. In this latter simulation there is only one parameter to control, which is the radius of the perturbation region, k_r. As the simulation is less computationally expensive, the range of values for k_r is increased to $\{8, 16, 32, 64, 128, 256\}$ compared to the Navier-Stokes simulations.

All the codes and data used for the simulations as well as animations of the dynamics of both problems are available in the following repository: https://github.com/victorballester7/final-master-thesis (accessed on June 30, 2024).

3.2. Results

The results of the simulations are presented in this section. The first part of the section is dedicated to the results of the Navier-Stokes simulations, while the second part is dedicated to the results of the point vortex simulations.

We start showing how the vortices spread across the domain in a visual manner.



(a) $k_r = 64$, Re $= 32$ **(b)** $k_r = 16$, Re $= 128$

Figure 2: Vorticity plots for different values of k_r and Re. The left-hand side figure is integrated with a resolution of 8192×8192 and the right figure is integrated with a resolution of 4096×4096.

Fig. 2 shows the vorticity evolution at two different time slices for the driven Navier-Stokes equation. The most notable feature is that the vortices appear more intense in the right plot compared to the left plot. This can be attributed to two main reasons. Firstly, the Reynolds number is smaller in the left plot, causing dissipation by viscosity to have a more significant impact, leading to a quicker decay of the intensity of the vortices. Secondly, the initial size of the vortices in the left plot is four times smaller than in the right plot, which restricts their growth in size as time progresses.

4 Conclusions

Acknowledgements

I would not like to finish this project without thanking the people who have helped me along the way. First and foremost, I would like to thank my two supervisors, Prof. Alexandros Alexakis and Prof. Emmanuel Dormy, for their guidance and constant support throughout the project. This project was, in the theoretical perspective, a bit far from my previous works, but their continuous help and advice made it possible. I would also like to thank the whole group led by Professor Stéphan Fauve in the Laboratoire de Physique de l'École Normale Supérieure, for giving me a desk and a place to work during my internship. Finally, I appreciate the huge amount of time and resources that both supercomputer centers, IDRIS and MESOPSL, have provided me during the development of this project. Without their technological resources, this project would have been impossible to carry out.

References

Alexakis, A. and L. Biferale (2018). "Cascades and transitions in turbulent flows." In: *Physics Reports* 767-769. Cascades and transitions in turbulent flows, pp. 1–101. ISSN: 0370-1573. DOI: 10.1016/j.physrep.2018.08.001.

Alexakis, A. (2023). "How far does turbulence spread?" In: *Journal of Fluid Mechanics* 977, R1. DOI: 10.1017/jfm.2023.951.

Aref, H. (2007). "Point vortex dynamics: A classical mathematics playground." In: *Journal of Mathematical Physics* 48.6, p. 065401. ISSN: 0022-2488. DOI: 10.1063/1.2425103.

Batchelor, G. K. (2000). *An Introduction to Fluid Dynamics*. Cambridge Mathematical Library. Cambridge University Press.

Boffetta, G. and R. E. Ecke (2012). "Two-Dimensional Turbulence." In: *Annual Review of Fluid Mechanics* 44.Volume 44, 2012, pp. 427–451. ISSN: 1545-4479. DOI: 10.1146/annurev-fluid-120710-101240.

Brachet, M. E. et al. (2008). *High-order low-storage explicit Runge-Kutta schemes for equations with quadratic nonlinearities*. arXiv: 0808.1883.

Ceci, S. and C. Seis (2022). "On the dynamics of point vortices for the two-dimensional Euler equation with L^p vorticity." In: *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 380.2226, p. 20210046. DOI: 10.1098/rsta.2021.0046.

Frisch, U. (1995). *Turbulence: The Legacy of A. N. Kolmogorov*. Cambridge University Press.

Griffiths, D. J. (2023). *Introduction to Electrodynamics*. 5th ed. Cambridge University Press.

Mininni, P. D. et al. (2011). "A hybrid MPI-OpenMP scheme for scalable parallel pseudospectral computations for fluid turbulence." In: *Parallel Computing* 37.6, pp. 316–326. ISSN: 0167-8191. DOI: 10.1016/j.parco.2011.05.004.

Saffman, P. G. (1993). *Vortex Dynamics*. Cambridge Monographs on Mechanics. Cambridge University Press.